An Integrated Semantic-Based Framework for Intelligent Similarity Measurement and Clustering of Microblogging Posts

Noufa Abdulaziz Alnajran

A thesis submitted in partial fulfilment of the requirements of the Manchester Metropolitan University for the degree of Doctor of Philosophy

School of Computing, Maths and Digital Technology
Faculty of Science and Engineering

March 2019

Abstract

Twitter, the most popular microblogging platform, is gaining rapid prominence as a source of information sharing and social awareness due to its popularity and massive user generated content. These include applications such as tailoring advertisement campaigns, event detection, trends analysis, and prediction of micro-populations. The aforementioned applications are generally conducted through cluster analysis of tweets to generate a more concise and organized representation of the massive raw tweets. However, current approaches perform traditional cluster analysis using conventional proximity measures, such as Euclidean distance. However, the sheer volume, noise, and dynamism of Twitter, impose challenges that hinder the efficacy of traditional clustering algorithms in detecting meaningful clusters within microblogging posts. The research presented in this thesis sets out to design and develop a novel short text semantic similarity (STSS) measure, named TREASURE, which captures the semantic and structural features of microblogging posts for intelligently predicting the similarities. TREASURE is utilised in the development of an innovative semantic-based cluster analysis algorithm (SBCA) that contributes in generating more accurate and meaningful granularities within microblogging posts. The integrated semantic-based framework incorporating TREASURE and the SBCA algorithm tackles both the problem of microblogging cluster analysis and contributes to the success of a variety of natural language processing (NLP) and computational intelligence research.

TREASURE utilises word embedding neural network (NN) models to capture the semantic relationships between words based on their co-occurrences in a corpus. Moreover, TREASURE analyses the morphological and lexical structure of tweets to predict the syntactic similarities. An intrinsic evaluation of TREASURE was performed with reference to a reliable similarity benchmark generated through an experiment to gather human ratings on a Twitter political dataset. A further evaluation was performed with reference to the SemEval-2014 similarity benchmark in order to validate the generalizability of TREASURE. The intrinsic evaluation and statistical analysis demonstrated a strong positive linear correlation between TREASURE and human ratings for both benchmarks. Furthermore, TREASURE achieved a significantly higher correlation coefficient compared to existing state-of-the-art STSS measures.

The SBCA algorithm incorporates TREASURE as the proximity measure. Unlike conventional partition-based clustering algorithms, the SBCA algorithm is fully unsupervised and dynamically determine the number of clusters beforehand. Subjective evaluation criteria were employed to evaluate the SBCA algorithm with reference to the SemEval-2014 similarity benchmark. Furthermore, an experiment was conducted to produce a reliable multi-class benchmark on the European Referendum political domain, which was also utilised to evaluate the SBCA algorithm. The evaluation results provide evidence that the SBCA algorithm undertakes highly accurate combining and separation decisions and can generate pure clusters from microblogging posts.

The contributions of this thesis to knowledge are mainly demonstrated as: 1) Development of a novel STSS measure for microblogging posts (TREASURE). 2) Development of a new SBCA algorithm that incorporates TREASURE to detect semantic themes in microblogs. 3) Generating a word embedding pre-trained model learned from a large corpus of political tweets. 4) Production of a reliable similarity-annotated benchmark and a reliable multi-class benchmark in the domain of politics.

Table of Contents

List of Equations

List of Figures

List of Tables

Acknowledgements

الحَمْدُ لله الذِّي هدَانا لِهذَا ومَا كُنَّا لِنَهْتدِيَ لَوْلَا أنْ هَدَانا الله

All praise are for Allah for the Guidance throughout this achievement, which otherwise wouldn't be possible.

All praises and thanks be to Allah for helping me overcome the challenges in the completion of this thesis. Indeed, I am eternally grateful for all the blessings Allah has given me.

My sincere gratitude is expressed to my esteemed and kind supervisor, **Dr. Keeley Crockett,** Reader in Computational Intelligence. She has given liberally of her time and freely of her expert knowledge, to the great benefit of this research. Both her patience and critical interest at every stage have been the source of great encouragement. Without her support, fruitful discussions and guidance, this research would not have been possible. Truly, I have been blessed to be supervised by the very best.

Particular thanks are appropriate to my co-supervisors, **Dr. David McLean** and **Dr. Annabel Latham** for their support, encouragements, and insightful suggestions. Their invaluable guidance and advice have been of immense benefit to this research.

My greatest thanks and appreciation go to my parents who taught me the value of hard work and dedication. Their belief in me never wavers, and they always lift my spirit with their constant love and support. Thank you for making this pursuit worthwhile.

Last but not least, I should not neglect to thank my sponsor, the government of Saudi Arabia, represented by the Ministry of Higher Education and the Saudi Cultural Bureau in London, for their continuing help and support whilst undertaking this research in the United Kingdom.

List of Author Publications

Conference

Alnajran, N., Crockett, K., Mclean, D. and Latham, A. (2016) *Cluster Analysis of Twitter Data: A Review of Algorithms.* Vol. 2: Science and Technology Publications (SCITEPRESS)/Springer Books, Portugal.

Alnajran, N., Crockett, K., Mclean, D. and Latham, A. (2018) *A Word Embedding Model Learned From Political Tweets*. 13th IEEE International Conference on Computer Engineering & Systems, Egypt.

Alnajran, N., Crockett, K., Mclean, D. and Latham, A. (2018) *An Empirical Performance Evaluation of Semantic-Based Similarity Measures in Microblogging Social Media*. In Proceedings of the Fifth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT 2018). Zurich: IEEE/ACM.

Alnajran, N., Crockett, K., McLean, D. and Latham, A. (2018) *A Heuristic Based Pre-processing Methodology for Short Text Similarity Measures in Microblogs*. IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Exeter.

Journals

Alnajran, N., Crockett, K., McLean, D. and Latham, A. (2018) *A Comparison of Unsupervised Learning Algorithms and Challenges in Microblogging Data Analysis*. Social Network Analysis and Mining (SNAM), (Accepted, in press).

Alnajran, N., Crockett, K., McLean, D. and Latham, A. (2018) *TREASURE: A Tweet Similarity Measure Based on Semantic and Syntactic Computation*. ACM Transactions on Knowledge Discovery from Data (TKDD), (under review).

Alnajran, N., Crockett, K., McLean, D. and Latham, A. (2018) *A Semantic-Based Cluster Analysis Algorithm for Detecting Semantic Themes in Microblogging Posts*. IEEE Transactions on Knowledge and Data Engineering (TKDE), (under review).

List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
ASW	Average Silhouette Width
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
FCM	Fuzzy *C*-means
FP	False Positive
FN	False Negative
GloVe	Global Vectors for Word Representation
HAL	Hyperspace Analogue to Language
IR	Information Retrieval
IRR	Inter Rater Reliability
JSON	JavaScript Object Notation
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
ML	Machine Learning
MAE	Mean Absolute Error
MSE	Mean Squared Error
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
OSN	Online Social Network
SBCA	Semantic-Based Cluster Analysis
SDLC	Software Development Life Cycle
SPSS	Statistical Package for the Social Sciences
STA	Semantic Textual Analysis
STS	Semantic Textual Similarity
STSS	Short Text Semantic Similarity
TN	True Negative
TP	True Positive
TREASURE	Tweet similaRity mEASURE
VSM	Vector Space Model

Chapter 1 – Introduction

1.1 Overview

This thesis presents the work undertaken in this research project and details the novel contributions to knowledge in the field of text mining and machine learning. In this chapter, the research is outlined and the contributions to knowledge are identified. The aim of this research is concerned with tackling the problem of predicting semantic similarities and discovering semantic themes (i.e. clusters) within microblogging posts. Towards achieving this aim, the work involved development of an innovative framework of integrated components for measuring the semantic similarity between short texts (sentence length) that can capture the challenging textual features in microblogging online social networks (OSN), primarily Twitter. The new similarity measure contributes in the development of a novel semantic-based algorithm for the problem of cluster analysis, which is intended to find semantically similar themes in the unstructured data. The aim is to develop a general and scalable approach that can jointly solve such interrelated problems and can be utilised in different contexts of pattern recognition such as leveraging marketing value through OSN analysis, event detection and summarization (De Boom et al., 2015b), political argumentation mining (Lippi and Torroni, 2016), and topic modelling (Fang et al., 2016).

1.2 Background and Problem Statement

The rapid evolution of Web 2.0 technologies such as OSNs, has led to a continuous generation of enormous volume of digital heterogeneous data being published at an unprecedented rate. Twitter (microblogging OSN) has quickly become a goldmine providing potential opportunities to extract actionable patterns that can be beneficial for businesses, users, and consumers (Gundecha and Liu, 2012). These opportunities include applications such as predicting presidential elections (Heredia et al., 2018), tailoring advertisements for groups with similar interests (Friedemann, 2015), event detection (De Boom et al., 2015b), trending issues extraction (Purwitasari et al., 2015), and prediction of micro-populations (Sinnott and Wang, 2017). Tremendous value lies in reasoning about such data in order to derive meaningful insights from it (Gundecha and Liu, 2012, Mondal and Deshpande, 2014). However, the sheer volume, noise, and dynamism of microblogging nature, imposes several challenges such as in the training

of machine learning (ML) algorithms to accurately segment this unstructured data into relevant clusters in order to achieve different higher-level natural language processing (NLP) applications (Adedoyin-Olowe et al., 2013).

ML applications provide a range of techniques to detect useful knowledge from massive datasets (Adedoyin-Olowe et al., 2013). Classification is a supervised machine learning technique where a labelled training dataset is provided for the classifier to be able to classify a testing dataset, whereas clustering segments data instances based on similarities between their features, with no prior understanding of the groups structure (Aggarwal and Zhai, 2012). The application of these techniques on microblogging posts could provide means of managing huge volumes of unstructured content and knowledge extraction. This has the potential to contribute to a paradigm shift of big data mining in the field of OSN. However, the application of traditional ML techniques on the massive human generated content yields degradation in their performance. This is often due to the natural language characteristics of OSN data, such as sparseness, large-scale, non-standardization, and ambiguities (Xu et al., 2013). Previous studies have proposed various models such as Bayes and Support Vector Machine to classify short text (i.e. microblogs) into predefined partitions using only syntactic text features (Lee et al., 2011, Go et al., 2009b). However, studies have shown that techniques utilizing only syntactic or static keyword lists, such as bag-of-words (BOW) are inadequate for providing rich mining results, as they do not analyse meanings behind the text (Cordobés et al., 2014, Sriram et al., 2010).

Semantic Textual Analysis (STA) considers inner structure semantic levels and the correlation of texts through utilizing lexical resources and knowledge bases such as the WordNet ontology (Miller et al., 1990), in order to convey meanings. Multiple studies of graph-based (Sriram et al., 2010) and vector-based (Xu et al., 2013, Li et al., 2006) approaches to short-text semantic analysis have been conducted, which exploited both semantic nets and corpus statistics. Previous studies often base their semantic computations on computing path lengths between synsets in a lexical taxonomy (Sultan, 2016), which encompasses relational specification of a conceptualization in graph-based hierarchy for the classical English concepts. Knowledge-based STA has demonstrated great success in NLP applications such as semantic similarity computation of different length text. However, solutions where often implemented for a more formal and structured English text, which do not work

for the language used in an informal sense, such as Twitter, due to the high presence of out-of-vocabulary (OOV) words.

Towards a generalizable integrated semantic-based framework for clustering microblogging text into semantic-driven themes, the research presented in this thesis integrates neural network based semantic technologies in the development of a novel short text semantic similarity (STSS) measure for microblogging posts, named TREASURE (**T**weet simila**R**ity m**EASURE**). TREASURE is incorporated into the development of a new semantic-based cluster analysis (SBCA) algorithm to detect semantic themes in the domain of politics (active OSN domain and rich source of controversial views). Therefore, this research contributes a novel framework that integrates new semantic approaches to intelligently discover similar themes, despite the high level of noise present in unstructured microblogging text. Unlike most existing studies that use formal knowledge bases in NLP applications of OSN text, this research utilizes large volumes of tweets to generate a neural embedding model that automatically (with no supervision) learns semantic relationships (co-occurrences) between words and the patterns in which words and common user conventions (e.g. hashtags) are employed in tweets. Thus, this approach not only captures the meaning of dictionary-based words, but also derives representations of the informal human generated words used in social media. In addition to the semantic features extraction, the novel approach takes into consideration the morphological structure of a tweet in assessing the underlying similarity. Altogether, the syntactic and semantics features derived from a tweet, jointly form the corresponding feature vector. Therefore, the work encompasses the development of a novel semantic similarity hybrid approach for extracting syntactic and semantic features from tweets based on training a word embedding model. Furthermore, a new semantic-based cluster analysis (SBCA) algorithm is developed using the new STSS method as the proximity measure. Thus, this research provides a generalizable semantic-based framework for automatically detecting potential themes in high volume social data, which indeed extends the field of NLP applications for the context of OSN textual analysis.

1.3 Research Area

This research spans several overlapping disciplines, including NLP, Machine Learning (ML), and Semantic Textual Analysis (STA), which are combined and built

upon to develop a novel integrated framework to induce semantic representations for the noisy and unstructured microblogging posts. This new framework can be generalized to solve multiple higher-level NLP tasks, such as credibility detection, arguments categorisation, knowledge extraction, and informal conversational agents. Therefore, it delivers a structured mechanism for intelligently processing and extracting different types of knowledge from the huge volume of user generated content in microblogs.

1.4 Research Scope

The research presented in this thesis integrates semantic technologies into similarity computation and cluster analysis of microblogging posts, particularly Twitter, applied in the domain of politics (active OSN domain and rich source of controversy views). The aim is to develop a novel integrated framework of semantic-based components to intelligently predict similarities and detect semantically similar themes within microblogging posts. Towards achieving this goal, a hybrid STSS measure (TREASURE) that consists of both semantic and syntactic components is developed to ultimately derive the structure and meaning of tweets in vectors and calculates the overall similarity accordingly. Existing solutions are often based on static keyword lists, lexical knowledgebase hierarchies such as WordNet, or classical word representations (further elaborated in Chapter 2). However, the new approach behind the development of the novel algorithm in this research utilizes a neural network architecture to train a word embedding model in order to construct a lexical resource from which semantic computations are computed. The trained word embedding model learns from a large corpus of tweets with no supervision to generate word vector representations that capture co-occurrence relationships between words. In addition to extracting semantic features from the text, the morphological and lexical structure of a tweet is analysed through deriving syntactic features such as part-of-speech (POS) tags and common Twitter user conventions such as hashtags. The hybrid feature set jointly form a tweet vector consisting of the semantic and syntactic attributes that represent the entities extracted from tweets. Ultimately, this research fills the gap of meaning-less keyword based similarity computation and cluster analysis in microblogs, and moves it towards semantic-based reasoning that can intelligently compute similarities and automatically detect latent themes. The research integrates NLP techniques involved in the data collection and pre-processing stages, and ML

algorithms in training the word embedding model utilised in the development of the novel STSS measure and development of the semantic-based cluster analysis (SBCA) algorithm for microblogs.

1.5 Research Aim

The overall aim of this research is to develop a novel semantic-based integrated framework that clusters OSN microblogging text, particularly Twitter, into different observed themes, according to the tweet's meaning. This will involve developing a short-text semantic similarity measure for the informal English language used in Twitter. Semantics will be combined with other features extracted from the tweet, to develop a distance measure in a new clustering algorithm. A new method of subjective evaluation will be dessigned to validate the new similarity computation method and clustering approach. An intrinsic evaluation will be performed with reference to existing benchmark datasets as well as using a benchmark dataset produced by human judges for the political domain.

1.6 Research Questions

Two general research questions are addressed in this work:

1. Is it possible to intelligently measure the degree of semantic equivalence between OSN microblogging posts using an automated semantic computation method?
2. Is it possible to automatically discover semantic themes in OSN microblogging posts based on an automated semantic computation method?

The answers pose the topic for this research –the need for adding intelligent semantic processing to enhance and improve ML applications on the unstructured OSN text.

1.7 Research Objectives

The objectives of the research presented in this thesis are:

1. Research current NLP and undertake a review of state-of-art STSS measures and empirically assess viability for incorporation in a cluster analysis algorithm for social media textual data.
2. Research current unsupervised learning technologies in the context of microblogging OSN, particularly Twitter, and its associated challenges. The aim is to deliver new insights into clustering microblogging posts by reviewing

and empirically assessing the viability of STSS measures and approaches for adding semantic meaning to clusters.

3. Stream social textual data (tweets) for a pre-defined domain (Politics) and design a pre-processing methodology to transform the raw data to a semantic-rich dataset.
4. Using the collected and pre-processed dataset, generate a word embedding model through unsupervised training of a neural embedding model.
5. Based on the pre-trained word embedding model, design and implement a novel short text semantic similarity (STSS) measure that is capable of capturing the underlying meaning and structure in microblogging posts.
6. Design an experimental methodology to produce a similarity-labelled benchmark by human judges on the political dataset and perform statistical tests to assess the level of inter-judge agreement
7. Design an experimental methodology for intrinsic evaluation of the developed similarity computation method with reference to benchmark datasets and compare the achieved correlation to state-of-art methods.
8. Design and implement a new semantic-based cluster analysis (SBCA) algorithm based on the implemented STSS, to find latent semantic themes (clusters) in the tweets dataset.
9. Design an experimental methodology to produce a multi-class benchmark by human judges on the political dataset and perform statistical tests to assess the level of inter-judge agreement.
10. Design an experimental methodology to validate the SBCA algorithm through computing external evaluation criteria to evaluate how well the clustering matches the benchmark classes.

Figure 1.1 outlines the research objectives, and where in this thesis it is addressed and situated.

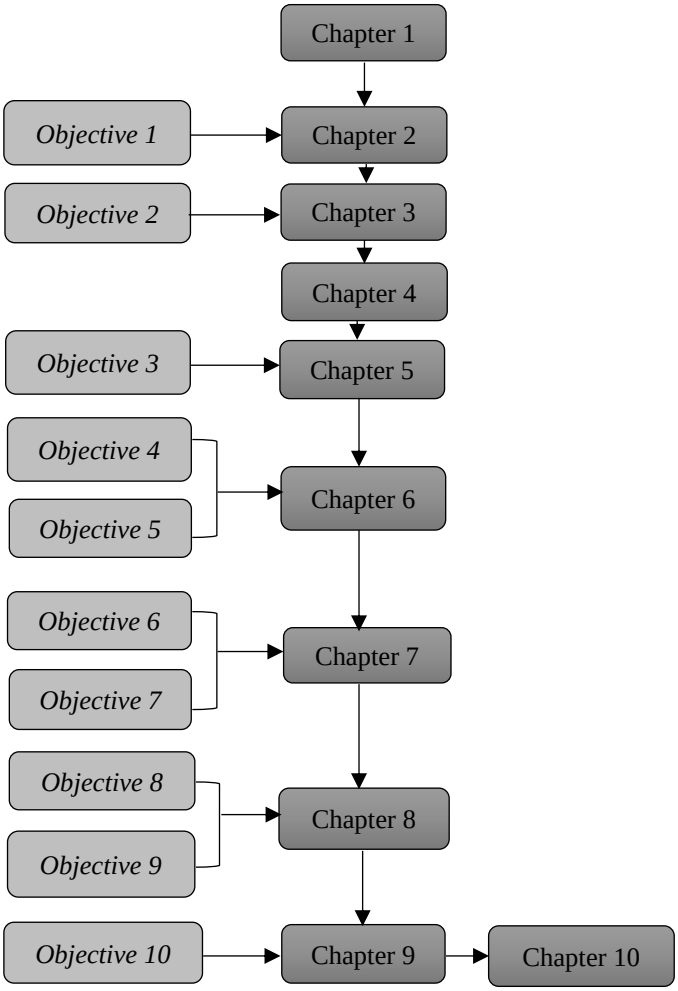


Figure 1.1 Mapping research objectives to their related chapters

1.8 Research Contributions

The novel contributions of the research presented in this thesis are:

1. A novel framework for streaming microblogging posts, pre-processing, and extracting semantic and syntactic hybrid feature set that reduces the challenges associated with the high volume of unstructured text (Chapter 5).
2. A generalizable methodology for building and training a neural network based language model on a microblogging text corpus to generate distributed word representations that capture semantic relationships between words (Chapter 6).
3. A novel architectural design for creating a semantic similarity computation measure for microblogging textual data and a generic development methodology (Chapter 6).

4. A methodology for producing benchmark datasets of human judgements demonstrating a good level of inter-judge agreement on classes and similarities of tweet pairs (Chapter 7).
5. A new methodology to evaluate and validate a semantic similarity computation measure for microblogging posts from the intrinsic and extrinsic perspective (Chapter 7).
6. Evidence that a novel generalized similarity computation measure based on extracting hybrid semantic and syntactic features can approximate humans' typical cognitive perceptions of similarities (Chapter 7).
7. A novel semantic-based cluster analysis (SBCA) algorithm that uses the new similarity computation approach as the proximity measure to detect semantic themes within microblogging posts (Chapter 8).
8. A new methodology to evaluate and validate a microblogging cluster analysis algorithm from the subjective perspective using external evaluation criteria (Chapter 9).
9. Evidence that a new semantic-based cluster analysis algorithm based on an intelligent proximity measure can detect semantic themes within microblogging datasets (Chapter 9).

1.9 Thesis Structure

The research conducted in this thesis is presented over ten chapters. Chapter two details the background review of existing literature and the current state of research related to the following:

- Role of microblogging OSN, particularly Twitter, in different aspects of information sharing and knowledge discovery.
- Importance of STSS measures for a wide range of microblogging applications.
- Main approaches to the development of STSS measures, which are knowledge-based, statistical-based, and hybrid-based STSS.
- Critical review of existing Twitter-based STSS applications and discuss their weaknesses in predicting the semantic similarities between tweets.
- Discussion on literature observations in terms of textual challenges for STSS measures in microblogs, which demonstrate the lack of an intelligent STSS measure.

Chapter three details the background review of existing literature and the current state

of research related to the following:

- Generalized comparison criterion, upon which a systematic review of unsupervised learning approaches and generalized conclusions are derived.
- Review of various clustering algorithms that are implemented for different features of microblogging textual datasets.
- Investigation of the reviewed algorithms to the comparative breadth of unsupervised learning approaches and success criteria that are used for measuring and evaluating the accuracy of clustering algorithms.
- Comparison of relevant studies in terms of clustering approaches, algorithms, number of clusters, dataset(s) size, distance measure, clustering features, evaluation methods, and results.
- Discussion on the main challenges faced by unsupervised analytical algorithms in social textual data.
- Highlighting potential weaknesses of current clustering algorithms in mining microblogging posts.

Chapter four details the research philosophy and methodology and the theoretical basis of the research derived from the background and literature chapters. It details the main phases of developing and evaluating the TREASURE STSS measure and the SBCA algorithm. In addition, the software facilitation and the data collection and analysis related to the benchmark production experiments are outlined and presented in this chapter.

Chapter five presents the methodology undertaken to collect, store, and construct a dataset from the Twitter microblogging platform in the particular domain of politics, which is a rich source of controversial views. It provides a description of the dataset in terms of size and utilised feature set. The chapter describes and evaluates a new pre-processing heuristic developed for short STSS measures. The consecutive rules of this heuristic process raw microblogging posts through different NLP stages in order to reduce noise and generate a semantic-rich dataset.

Chapter six presents the development process adopted to implement an STSS measure for microblogging posts, named TREASURE. Towards the development of TREASURE, chapter 6 describes the stages carried out to train a word embedding model and generate word vector representations that captures the statistical semantic relationships between words based on their co-occurrences. The development process

of TREASURE was divided into two phases: the first phase was designing and implementing the semantic modules using the pre-trained word embedding model, and the second phase was designing and implementing the syntactic modules. A weighting schema is also described from which the overall similarity score is calculated.

Following the development of TREASURE, chapter seven presents the intrinsic evaluation methodology in order to validate the effectiveness of the TREASURE STSS measure. The first experiment was conducted with human participants to generate a benchmark of similarity-annotated tweet pairs on the political domain, utilising the political dataset (described in Chapter 5), which is a rich source of controversial views. The second experiment uses the generated political benchmark to evaluate the strength of linear or monotonic association between TREASURE measurements and the human judgements. The third experiment was conducted to assess the generalizability of TREASURE to a different domain, which is general news in twitter. Statistical analysis was performed on results of the three experiments in order to test three hypotheses related to the first main research question outlined in Section 1.6.

Chapter eight in this thesis presents the development process adopted to implement a semantic-based cluster analysis (SBCA) algorithm for detecting semantic themes within microblogging posts. Chapter eight discusses the development process in terms of the clustering objective function, proximity measure (TREASURE), data structures utilised to reduce the algorithm's computational demand, and the clustroids. An illustration of the SBCA algorithm through a pseudocode and a flowchart is also presented in this chapter. Furthermore, the time and space complexities of the SBCA algorithm in relation to other clustering algorithms are discussed in order to provide means of the algorithms scalability to handle high volume microblogging posts.

Following the development of the SBCA algorithm, chapter nine presents the design of an evaluation methodology for the SBCA algorithm in order to answer the second main research question outlined in Section 1.6. In Chapter nine, the evaluation methodology was carried out through undertaking three experiments designed to evaluate the SBCA algorithm. The first experiment was conducted utilising a similarity labelled benchmark dataset (described in Chapter 5), which consists of similarity ratings for tweet pairs. This experiment was performed in order to determine the optimal value of TREASURE similarity threshold, τ, which will determine if the

SBCA algorithm will assign a new instance to an existing cluster or to a new cluster. The second experiment was conducted with human participants to generate a benchmark of tweets classifications into semantic categories utilising the political dataset (data collection, pre-processing methodology, and features extraction are described in Chapter 5). The third experiment used the threshold determined by experiment (1) in order to detect semantic themes within the political dataset. The resulting clusters were evaluated using five external evaluation criteria with reference to the multi-class benchmark generated from experiment (2).

Chapter ten presents the conclusions drawn from the research findings and discussion. It also outlines the main contributions of the research and provides recommendations for future research.

Chapter 2 – Semantic Textual Similarity (STS)

2.1 Overview

This chapter provides a background review of existing literature and the current state of research of short text semantic similarity (STSS) measurement and its applicability in the context of microblogging short text messages (posts). These posts share special lexical and syntactical characteristics such that the semantic similarities between them cannot be captured by traditional STSS measures, which analyse proper English sentences. Therefore, this chapter sets out to critically review and empirically evaluate different approaches to STSS measures and compare their performance in the context of microblogs, particularly Twitter. The critical analysis conducted in this review provides an important resource for research aiming to adapt or develop new STSS measures that consider the different sorts of noise present in social media data.

The purpose of this chapter is to:

1. Provide a background on the role of microblogging online social networks (OSN), particularly Twitter, in different aspects of information sharing and knowledge discovery.
2. Highlight the importance of STSS measures for a wide range of microblogging applications.
3. Describe the three main approaches to the development of STSS measures, which are knowledge-based, statistical-based, and hybrid-based STSS.
4. Undertake a critical review of existing Twitter-based STSS applications and discuss their weaknesses in predicting the semantic similarities between tweets.
5. Discuss literature observations in terms of textual challenges for STSS measures in microblogs, which demonstrate the lack of an intelligent STSS measure.

2.2 Role of Microblogging in Social Consciousness and Knowledge Discovery

Microblogs are OSNs that allow users to create and share short messages. Twitter is one of the most popular microblogging platforms in wide areas around the globe (Mohammadi et al., 2018). Twitter is gaining rapid prominence as a source of information sharing and social awareness due to its popularity and massive user generated content. Furthermore, Twitter has become a goldmine of potential insights

and knowledge discovery serving different purposes. In academia, Twitter has been utilized to communicate and publish messages related to scientific events in real-time (Ross et al., 2011). Another important use case of Twitter is demonstrated in the business domain for marketing purposes. In this case researchers have been analyzing users posts and comments related to certain products and services in order to promote the competitiveness of a certain business strategy (Boffa et al., 2018). Twitter have also been utilized for healthcare and community awareness related research. In this context, Twitter provides the latest medical research as professional healthcare organizations largely have Twitter accounts, which are used to disseminate information regarding the latest research findings related to healthcare (Thompson et al., 2015). Moreover, Twitter have always been used to broadcast real-time risk awareness messages related to threatening events such as the hurricane Sandy (Lachlan et al., 2014). In the domain of politics, Twitter can be utilized to predict polls outcomes based on statistical analysis of *pro* and *against* political campaigns.

2.3 Importance of Similarity Computation for Microblogging Posts

Twitter applications have emphasized the importance of an effective approach to compute the semantic similarity between tweets. Examples of such applications are political engineering (Jungherr, 2016), trend analysis, truth discovery, and search ranking (Kim et al., 2018). These applications can be achieved through conducting cluster analysis of tweets to generate a more concise and organized representation of the massive raw tweets. An intelligent similarity measure, instead of conventional distance measures (e.g. Euclidean distance), incorporated within a clustering algorithm shall contribute in generating accurate and meaningful granularities for the target application. Measuring tweets similarities is useful for user-related applications as well. In detecting human behavior, tweets similarity can reveal hidden patterns on different human cognition and attitudes. In machine learning, tweet similarity is used to classify tweets into pre-determined categories (Lin et al., 2014). Moreover, the incorporation of tweet similarity is beneficial for applications such as bilingual tweet translation evaluation (Jehl et al., 2012), where the quality of the system translation output is assessed by measuring the degree of equivalence between a human translation and the machine output. These exemplar applications show that computing tweet similarity plays a significant role in computational linguistics and has become a

generic component for the research community involved in OSN-related knowledge analysis and representation.

2.4 Short Text Semantic Similarity (STSS) Measures

STSS measures are employed for measuring the degree to which short-texts are subjectively evaluated by humans as being semantically equivalent to each other (Agirre et al., 2016b). Short-texts refer to typical human utterances that are of sentence length ranging from 10 to 25 words (O'Shea et al., 2008b). O'Shea et al. (2008a) suggested that semantic similarities of these short-texts can be measured through the application of STSS measures. However, human generated sentences in microblogs, such as tweets are prone to forms of text that do not conform to typical grammatical and syntactical rules of a sentence. Therefore, it is imperative to adapt traditional STSS measures in order to cater for the special characteristics of the sentences propagated in microblogs.

STSS measurements are gaining prominence contributing to the success of various research in the field of natural language processing (NLP) and artificial intelligence (AI). The task of assessing the semantic similarity between short-texts has been a central problem in NLP, due to its importance in a variety of applications. Some of the earliest text similarity applications have been implemented for text classification and information retrieval (Rocchio, 1971), automatic word sense disambiguation (Lesk, 1986), and extractive text summarization (Salton and Buckley, 1988). Further applications of STSS include the incorporation of the measure in a conversational agent to reduce the time associated with the scripting process (O'Shea et al., 2010), measuring the similarity between documents (Lin et al., 2014), and in supervised learning and text classification (Albitar et al., 2014).

Measuring semantic similarity can be performed at various levels, ranging from words, phrases and sentences, to paragraphs and documents. Each of these categories employ different methods and techniques to gauge the underlying meaning at that particular level.

The subsequent sections review the three major categories of semantic text similarity computation approaches: Knowledge-based methods, corpus based methods, and hybrid-based methods.

2.4.1 Knowledge-Based STSS

The semantic similarity between short-texts can be gauged through defining a topological similarity, which is based on using knowledge bases such as ontologies. The distance between terms and concepts are determined by means of these resources. Calculating the topological similarity between ontological concepts can be done either by using the edges and their types (edge-based) or the nodes and their properties (node-based) as data sources. Liu and Wang (2014) presented a topological measure for computing the semantic similarity between short texts based on the structural and semantic relationships in a predefined hierarchical concept tree (HCT), without requiring any additional corpus information. A major drawback of this approach is that it does not take into account the word's sequence in which it appears in the sentence. For instance, the sentences *the cat chased the dog* and *the dog chased the cat* would be considered identical.

Another drawback is related to the scalability and performance of the current state-of-the-art semantic measures libraries. The authors in (Lastra-Díaz et al., 2017) argue that these drawbacks are due to using naïve graph representation models, which fail to capture the intrinsic structure of the represented taxonomies. Consequently, topological algorithms that are based on naïve models suffer from degraded performance due to demanding high computational cost. This complexity problem is derived from the caching strategy adopted by current semantic measures libraries. This strategy stores all nodes' ancestors and descendants within the taxonomy, which significantly increases memory usage leading to scalability problems concerning the taxonomy size. Moreover, the dynamic resizing of the caching data structures, further memory allocation, or the integration with external relational databases will raise performance issues.

Three path length based methods were used to calculate the lexical similarity between words in WordNet, LCH (Leacock and Chodorow, 1998), JCN (Jiang and Conrath, 1997), and LESK (Lesk, 1986). LCH finds the shortest path between concepts in WordNet. This path length is then scaled by the maximum length observed in the "is-a" hierarchy, in which the two concepts occur. JCN, on the other hand, includes the information of the least common subsumer in addition to the shortest path length. Finally, LESK incorporates information from WordNet glosses, where it finds overlaps between the glosses of the two concepts under consideration, in addition to the

concepts that directly link to them.

Current state-of-the-art in knowledge-based STSS is a representation model for taxonomies, along with a new software library, which is based on that model (Lastra-Díaz et al., 2017). The model is claimed to properly encode the intrinsic structures and bridges the aforementioned gaps of scalability and performance in the field of semantic textual analysis. It is an adaptation of the half edge representation in the field of computational geometry (Botsch et al., 2002) in order to represent and interrogate large taxonomies in an efficient manner.

While the reviewed approaches show relatively high correlations with human judgments when applied to annotated English sentence pairs, they are expected to fall short when used to compute the similarity between tweets. This is due to the common Twitter-based features that contribute to the overall tweet similarity (e.g. hashtags, mentions, emoticons, etc.), which are not taken into consideration.

2.4.2 Statistical-Based STSS

Statistical approaches (sometimes referred to as corpus-based approaches) determine the semantic similarity between short texts through calculating words co-occurrence frequencies and weightings based on a large corpus of text. Term weighting assigns a value to unigrams according to their information content in a text corpus (Li et al., 2006) The most common corpus weighting approach is 'term frequency-inverse document frequency' (TF-IDF) (Salton and Buckley, 1987), which assumes that documents have common words (Allan et al., 2003, Akkaya et al., 2009). This method is generally used in IR systems, in which each word is normalized by the frequency of its occurrence over all documents. It aims to favor documents' discriminatory traits over nondiscriminatory ones (e.g. *Trump* vs. *on*). That is, words that frequently occur in a document or a corpus such as prepositions are considered less informative than words occurring less frequent. It is claimed by Atoum et al. (2016) that this method is not suitable for short-text of sentence length such as tweets because these may have null common words. The researcher argues that also words in a tweet are likely to occur only once as tweets are length-constrained, which creates an upper limit on the TF, reducing the importance of that portion of the weighting scheme. However, IDF should still give smaller weights for commonly occurring words in the corpus of all dataset tweets and higher weights for less occurring ones.

2.4.2.1 Count-Based Approaches

Latent Semantic Analysis (LSA) is the traditional statistical-based semantic similarity measure, which is provided as a method for information retrieval (Deerwester et al., 1990). LSA, which is sometimes referred to as Latent Semantic Indexing (LSI), is based on the distributional hypotheses that words similar in meaning will occur in similar contexts (Harris, 1968). Therefore, calculating words similarities can be derived from a statistical analysis of a large text corpus. The set of unique terms and documents (short-texts in this context) in the corpus are used to generate a high dimensional matrix of terms occurrences. This term-document matrix is commonly decomposed by the application of a matrix factorization algorithm such as Singular Value Decomposition (SVD). The incorporation of SVD into LSA reduces the dimensionality of the single frequency matrix through approximating it into three sub matrices, term-concept matrix, singular value matrix, and concept-document matrix. The SVD process in LSA preserves the important semantic information while reducing noise presented in the original space. It has been found that SVD has improved the effectiveness of word similarity measures (Landauer and Dumais, 1997). Hyperspace Analogue to Language (HAL) (Agirre et al.) is a variation of LSA where a word by context-word matrix is implemented instead of the word by document matrix (Burgess et al., 1998). HAL maintains a moving window of a predefined fixed size that sifts through the entire corpus, recording word/term co-occurrences in preceding and subsequent contexts. Vectors are formed from the co-occurrence matrices, from which the semantic similarity may be measured. Terms from which the co-occurrence matrix is derived are often valued by the TF-IDF weighting scheme (Jurafsky, 2000). HAL performs as well as LSA but without requiring the mathematical complexity steps of SVD.

Latent Dirichlet Allocation (LDA) is a semantic topic extraction model that is based on probabilities (Blei et al., 2003). LDA is a significant extension of LSA, where terms are grouped into topics, in which most of these terms exist in more than one topic (Crossno et al., 2011). Despite the commonalities between LDA and LSA, each of the algorithms generate distinct models. While LSA uses SVD in which the maximum variance across the data is determined for a reduced number of dimensions, LDA employs a Bayesian model. This model considers each document as a mixture of underlying topics and every topic is modelled as a mixture of term probabilities from

a vocabulary. Moreover, even though LDA and LSA outputs may be used in similar scenarios, the values of their outputs represent completely different quantities, with different ranges and meanings. LSA generates term by concept and document by concept correlation matrices, with values ranging between -1 and 1 with negative values denoting inverse correlations. On the other hand, LDA generates term by topic and document by topic probability matrices, in which probabilities range from 0 to 1. LDA has an advantage over LSA, which is its ability to tackle the problem of disambiguation and therefore has higher accuracy. This is achieved by comparing a document to two topics and determining which of them is closer to the document, across all combinations of topics that seem broadly relevant. This direct interpretation of similarities and differences between the most effective statistical semantic measures is important for the challenging process of understanding which measure may be most appropriate for a given text analysis task.

2.4.2.2 Prediction-Based Approaches

Based on the idea of corpus-based statistics, prediction based distributed representation of words learned by neural networks emerged, generating dense and continuous valued vectors called *embedding* (Collobert and Weston, 2008, Mikolov et al., 2013b). These embedding of words have become one of the strongest trends in machine learning and NLP to represent sparse and high dimensional data in a vectorial space of semantic features (Beam et al., 2018). Prediction based word embedding models, such as *word2vec* (Mikolov et al., 2013b, Mikolov et al., 2013a) and *GloVe* (Pennington et al., 2014) is gaining more attention over classical frequency-based vector representation models such as LSA, LDA, and HAL. Word embedding provides a more expressive and efficient representation of words by preserving their contextual similarity and constructing low dimensional vectors (Naili et al., 2017). In word embedding, an unsupervised learning approach is performed on a huge corpus to learn word representations using a neural network. Naili et al. (2017) reported that prediction-based word embedding models outperform the classical counter-based word vector representation in LSA. Furthermore, it has been reported that Word2Vec outperform GloVe for both English and Arabic languages (Naili et al., 2017).

In recent years, there has been an increase in approaches proposing to compose word vectors by using neural language models, which have a core of trained neural networks

(Christoph, 2016). Given a sequence of initial words, early neural models were designed to predict the next word in the sentence (Mnih and Hinton, 2009) (e.g. text input auto-completion). While these models can be trained with a variety of techniques to achieve different tasks, they share a common feature of having at their core a dense vector representation of words that can be exploited for computing similarity. This representation is commonly referred to as "neural word embedding", in which their effectiveness varies with regard to the chosen technique and corpus for similarity computation.

2.4.3 Hybrid-Based STSS

Some of the topological methods of estimating the semantic similarity may incorporate a statistical function of term frequency in a corpus in order to determine the value of a concept (Aggarwal et al., 2012, Li et al., 2006, Das and Smith, 2009, Kashyap et al., 2016, Bär et al., 2012). However, their fundamental component of determining the degree of semantic equivalence remains based on a predefined ontology. The similarity computation might also be composed of a combination of statistical and topological methods.

STASIS (Li et al., 2006) is an effective measure that estimates the semantic similarity between short sentences based on topological information derived from WordNet ontology and statistical information obtained through the use of the Brown corpus (Francis and Kucera, 1964). This measure calculates the overall semantic score of similarity between two sentences based on a function of multiple factors. These factors include the path between two synsets in the ontology, depth of the subsumer in the hierarchical semantic nets, and information content derived from the Brown corpus. STASIS forms a word order vector composed of unique words contained in both sentences. The combination of syntactic word order and semantic information determines the overall similarity. Although STASIS does not consider word sense disambiguation for polysemous words as this would scale up the measure's complexity, it still performs well as per the experimental results.

During the last few years, many state-of-the-art STSS approaches have used linear combinations of measures. For example, six topology-based and two statistical-based measures were tested in (Mihalcea et al., 2006), for the related task of paraphrase identification. In this work, the efficacy of applying topological-based word similarity measures was explored in comparison to texts. They reported that the two approaches

are comparable to corpus-based measures such as LSA. Islam and Inkpen (2008) proposed a method that uses a combination of mandatory (string and semantic word) and optional (common word order) similarities. Evaluated on a dataset of 30 sentence pairs, this method outperformed the correlation obtained by Li et al. (2006). Moreover, a hybrid approach was proposed in (Aggarwal et al., 2012) where the authors combined a statistical-based semantic relatedness measure over the complete sentence in addition to a topology-based semantic similarity scores that were computed for the words that share similar syntactical role labels in both sentences. These calculated scores act as the features that were fed to machine learning models to predict a single similarity score given two sentences. Results of this method showed a significant improvement of a hybrid measure compared to corpus-based measures taken alone. UKP (Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures) (Bär et al., 2012), is a similarity detection system that showed reasonable correlation results. It implemented a string similarity, a semantic similarity, text expansion mechanisms and measures related to structure and style. These multiple text similarity measures were combined with a simple regression model based on training data.

The reviewed work on hybrid measures demonstrate a number of successful studies in the field of STSS. However, these contribution may not produce good results for the task of measuring the semantic similarities between microblogging posts, particularly tweets. This is based on the consideration that, although the studies implement a hybrid approach, they derive the semantic relationships between words from a knowledgebase such as WordNet. The statistical analysis is used to obtain knowledge on the information content of the words from which a sentence is composed. Tweet contain many rapidly generated out of vocabulary (OOV) words that are not present in a formal English knowledgebase. Therefore, the application of the aforementioned approaches on these microblogging posts is anticipated to generate less accurate similarity measures.

2.5 Use of STSS in Twitter Applications

The variations in natural language expressions impose challenges in determining the degree of semantic equivalence between sentences. In natural languages, a single meaning of a sentence can be expressed in many ways, and therefore the task of

measuring the semantic similarity of natural language sentences is very complex. This problem is more prevalent in microblogging posts due to the informal nature and the high degree of lexical variations used. Areas of work within related fields, such as classification and clustering of tweets face similar issues when identifying similarities in natural language text presented in Twitter (Alnajran et al., 2017).

To illustrate some challenges present in Twitter, consider the following tweet (Farzindar and Inkpen, 2017):

> *#qcpoli enjoyed a hearty laugh today with #plq debate audience for @jflisee #notrehome tune was that the intended reaction?*

The presence of symbols, spelling mistakes, letter repetitions, and abbreviations complicate the process of tokenization and Part-of-Speech (POS) tagging required by text analysis tasks (Gómez-Adorno et al., 2016).

Little research has been conducted in the area of semantic analysis of Twitter data especially in relation to semantically measuring the degree of equivalence between tweets. This may be attributed to the characteristics of such data that make the task significantly more difficult than analysing general short-text. However, several studies highlighted the potential and significance of developing semantic similarity measures (Guo and Diab, 2012) and paraphrase identification techniques (Xu et al., 2013, Zanzotto et al., 2011) specifically for tweets. In the context of Twitter, semantic similarity measures are particularly useful in reducing the challenge of high redundancy and the sparsity inherent in its data. One of the possible approaches to reduce the complexity of dealing with massive data is through incorporating these measures in applications of ML such as topic detection (Rosa et al., 2011, Kim et al., 2012) and sentiment analysis (Ahuja and Dubey, 2017).

In general, there is considerable literature on measuring the similarity between sentences or short texts (Li et al., 2006, Soğancıoğlu et al., 2017, Pawar and Mago, 2018), but there are very few published research relating to the measurement of similarity between tweets. The subsequent sections review some related work in order to explore the strengths and limitations of previous methods, and to identify the particular difficulties in computing tweet similarity.

2.5.1 Keyword-Based Approach

The keyword-based methods are often known as the bag-of-words (BOW) representation, which is commonly used in NLP and Information Retrieval (IR)

applications (Barry et al., 2007). This model represents text as an unordered list of the words from which the text is composed. It does not consider grammatical structure or word order. In case of IR systems, a query is considered as a document, and the relevant documents to be retrieved are the ones that share similar keywords vector with the query vector. This method relies on the assumption that the similarity between documents increases as the common words between them increase. If this technique was applied to tweet similarity, it would have three obvious limitations:

1. Each tweet is represented by a feature vector of a precompiled Twitter-based word list with *n* words, in which *n* is generally in the millions in order to include all unique keywords (i.e. features) in the dataset under consideration. Hence, the resulting vectors are very sparse, as they would have many null components.
2. Most of the works in Twitter use a BOW model that ignores the discourse particles and stop words such as *but*, *as*, *since*, *of*, etc. However, these words cannot be ignored in tweet similarity computation as they carry structural information, which contributes to the interpretation of tweet semantics (Li et al., 2006). The inclusion of such words will increase the vector dimensionality even greater.
3. Tweets that are similar in meaning do not necessarily share common words and sharing many words does not imply similarity. Thus, the precompiled static list of words does not reflect the correct semantic information in the context of compared tweets.

An enhancement of the keyword-based approach is the use of semantic information to augment the keywords vector with semantic features to compute the similarity of word pair taken from the two candidate tweets. Similarity values of all word pairs are then aggregated to compute the overall tweet similarity (Okazaki et al., 2003). Subsequent sections provide a discussion on the work done in semantic similarity computation of tweets.

2.5.2 Knowledge-Based STSS in Twitter

Studies on detecting short-text similarity have centered on the traditional approach of analyzing potential types of relations in ontologies such as WordNet (Miller et al., 1990). These approaches consider hierarchical (e.g. *is-a*), associative (e.g. *cause-effect*), and equivalence (synonymy) relations of concepts. Such methods are usually

effective when dealing with text of proper English in which most of the terms used are present in the lexical hierarchy (Pawar and Mago, 2018). However, in Twitter, most of the text used is not likely to be present in semantic nets. This is mainly due to the 140-character limit, which imposes lots of shortened lingo of abbreviations and acronyms. Although Twitter has recently doubled the limit to 280-characters, it is still considered a short limit, which makes such microblog prone to informal jargons that pose serious computational challenges.

Rudrapal et al. (2015) proposed a method for measuring the semantic similarity between Bengali tweets using the Bengali WordNet developed by Das and Bandyopadhyay (2010). The Bengali model computes the semantic similarity score of a pair of tweets with a lexical based method. It is built based on analyzing common words similarity among tweets. The overall tweet similarity is obtained by dividing the sum of synonym words by the sum of *n* (length of tweet 1) and *m* (length of tweet 2). This method is similar to BOW as it presents a naïve approach to semantic similarity. This is due to the lack of consideration to the hierarchical relations such as path length or depth for words that are not in the same synset. Rather, it assigns a distance of one between them (i.e. 0 similarity). Authors claim that Bengali tweets are less noisy in nature compared to English, and therefore requires less comprehensive pre-processing. This is because people tend to use fewer abbreviated words (e.g. "great" instead of "gr8"), character repetition (e.g. "heeeey" for "hey"), etc. in Bengali tweets. Nevertheless, despite this claim, the authors proposed method is still weak in capturing the underlying similarities in tweets.

Another approach to applying knowledge-based STSS is provided in (Chen et al., 2012). The authors utilized WordNet to estimate the semantic score between microblogs and recommended the top similar microblog records to the user. In their approach, the authors computed the similarity between sentences based on the similarity of the pairs of words contained in the corresponding sentences. Furthermore, the semantic similarity between two word senses is captured through path length, in which the taxonomy is treated as an undirected graph and the distance is calculated between them based on WordNet. The performance of this approach was compared to a statistical based approach, and findings suggested that this knowledge-based approach performed better than the statistical-based one in terms of precision.

Based on a critical review, it has been observed that knowledge-based approaches often fall short when applied in Twitter similarity applications due to three main reasons:

1. Due to its informal nature, Twitter contains many improper words (i.e. misspellings, jargons, acronyms, slangs, etc.) that people come up with rapidly. These words are usually not present in semantic nets as they are generally human crafted dictionaries that do not capture all possible words. Therefore, much of the similarity between tweets will be missing because of the lack of word presence in the semantic hierarchy.
2. The most widely used knowledge base, WordNet, is limited in the number of verbs and adverbs synsets compared to the available nouns synsets. Hence, referring to the first reason, WordNet is considered a limited resource to be used for tweet similarity.
3. Semantic nets model polysemy and synonymy relations between concepts (unigrams). Therefore, relations between bigrams such as 'computer science' (or trigrams) are not represented.

A well-established and active field of research that contributes to semantic similarity computation is related to methods based on corpus statistical information of words. Corpus-based methods are generally categorized into: 1) word weighting methods (sometimes referred to as information content) and, 2) word co-occurrence methods.

2.5.3 Statistical-Based STSS in Twitter

As discussed in Section 2.4.2.1, LSA, LDA, and HAL are amongst the early word co-occurrence statistical models contributing to text similarity computation based on estimating continuous representation of words in a huge corpus. Steiger et al. (2015) used LDA to assess the semantic similarity among tweets. A corpus of 20.4 million processed tweets was created as the lexical resource for which LDA performed its semantic probabilistic model. The application of LDA reduced the semantic dimensions through clustering co-occurring words into topics. Each topic is referred to by labelling it with the highest probability associated words (>0.03). In their adopted approach of LDA, the authors assumed each tweet α contains a random number of topics, and each topic is characterized by a word distribution β (see Figure 2.1).

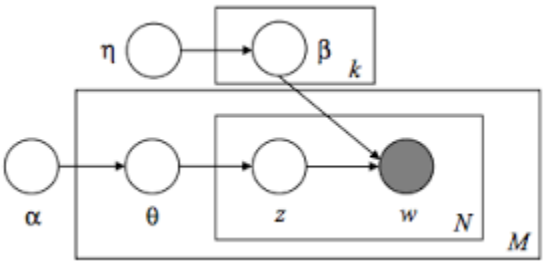


Figure 2.1 LDA graphical model (Blei et al., 2003)

For an individual word *w* within each tweet, *z* is the corresponding associated topic. The topic distribution for the overall number of tweets, *M* is denoted by *θ*, each being of length *N*. The main challenges encountered, were the estimation of the posterior parameter and the computation of variables such as the number of topics *k*. However, this study has several limitations that need to be further addressed. Some pitfalls within the bag-of-words (BOW) assumption of LDA caused words to be assigned to different topics while they should be associated with the same topic. Moreover, taking into consideration the syntactical structure (e.g. n-grams) would allow for word orders to be associated to several topics, and therefore better handle semantic complexities. Further, this study did not include the author-topic model (Zhao et al., 2011) (i.e. all tweets of the same user are treated as a single document) due to missing benchmarking process.

Another study that used LDA to gauge the semantic similarity in the context of Twitter data, includes the work presented by Chen et al. (2012), in which a corpus of 548 tweets is used. In this approach, each microblog post (tweet) is represented as a topic vector, and consequently, the similarity calculation between tweets is equal to the dot product of the two corresponding topic vectors. This statistical method of assessing the semantic similarity was evaluated and compared to the performance of the knowledge-based approach explained earlier in Section 2.4.2. The results showed that the knowledge-based approach performed better than the topic-based one in terms of precision.

However, when LSA is used to calculate tweet similarity, a vector for each tweet is constructed in the reduced dimension space; similarity is then measured by calculating the similarity between these two vectors (Foltz et al., 1998). LSA may fall short for tweet similarity computation due to two reasons:

1. The computational limitation of SVD imposes that the dimensionality of the reconstructed word-to-document matrix is limited in size. Therefore, the

reduced dimension space of LSA may not include important words in tweets from an unconstrained domain (and thus not represented in the corpus of training documents).

2. The vector representation of a tweet is likely to be very sparse as the dimension in LSA is fixed and vectors are therefore fixed.
3. LSA does not take into consideration any syntactic information from the two tweets being compared.

Therefore, LSA is considered to be more appropriate for text segments that are larger than the short text dealt with in this work (Dennis et al., 2003). Similarly, LDA falls short when applied to tweet similarity because, the idea behind LDA is that it assigns relevant topics for each document based on the context in each document, and as tweets lack context due to shortness, it will yield poor representations. Unlike LSA and LDA, HAL is memory-intensive as it does not perform any dimensionality reduction technique and therefore can be problematic when used in applications processing big datasets such as tweets.

In conclusion, as LSA, topic models (LDA), and HAL have been powerful in discovering latent semantic structures and traditional tasks for long document similarity computation, they fail in modeling tweets due to the severe sparseness and noise present in them (Mehrotra et al., 2013, Hong and Davison, 2010)

2.5.4 Prediction-Based Word Co-occurrence Approaches in Twitter

There is not much research conducted in OSN analysis using word embedding, particularly for tweet similarity computation. De Boom et al. (2015a) trained a Word2Vec model on a dataset of 10 million Wikipedia couples (i.e. pairs) to learn semantic similarities for short text fragments. Their proposed method combines knowledge from TF-IDF and word embedding to measure the semantic similarity between two fixed length pairs. The degree to which two pairs are semantically similar depends on the degree of similarity between their corresponding vector representations according to some distance measure. Their results show that the Word2Vec vectorial representation of words, combined with TF-IDF weightings might lead to a better model for semantic content within very short text fragments. Nevertheless, this conclusion needs further investigation for application in the context of Twitter. This is because Wikipedia contains structured information and is completely different textual

platform than a social medium such as Twitter, in which the content is mostly slang, abbreviated and erroneous (De Boom et al., 2015a). Moreover, the results are derived for short text of fixed length and have not analysed text of arbitrary length such as tweets. Dey et al. (2017) proposed a word embedding training model for single and multiple hashtags recommendation towards tweets. They developed one model for learning the embedding of each word in the corpus vocabulary and another model for learning the embedding of each word in the scope of an accompanying hashtag. Using word embedding, their system demonstrate a lift of 7.48 and 6.53 times for recommending a single hashtag and multiple hashtags to a given tweet respectively.

The observed literature around word embedding in the context of Twitter-based semantic textual analysis indicates and reveals potential capabilities of such techniques for OSN analysis. However, word embedding has not been used in semantic representation of tweets in the scope of semantic similarity computation. In addition, while syntactic information contributes to the overall meaning in a text fragment (Li et al., 2006), most of the aforementioned methods consider only semantic information when computing the similarity. As discussed in Section 2.5, microblogging posts can be challenging for knowledge-based methods, as most of the terms used in Twitter are not present in a structured and formal language ontology. Furthermore, tweets are challenging for classical vector representations and topic modelling methods due to the inadequate information and lack of context for manipulation by a computational method (Alnajran et al., 2018a).

2.5.5 Contribution of Hybrid STSS Approaches in Twitter

Das and Smith (2009) proposed an approach for measuring the semantic similarity between pairs of tweets through identifying whether the two hold a paraphrase relationship. The probabilistic model incorporates syntax and lexical semantics to compute the similarity between two sentences by using a logistic regression model, with eighteen features based on n-grams. The system builds a binary classification model for identifying paraphrase through using precision, recall, and F1-score of n-gram tokens from sentence pairs. The model is capable of determining whether there exists a semantic relationship between a pair of tweets. However, it may be improved by principled combination with more standard lexical approaches.

SemSim is a hybrid based semantic textual similarity system, composed of several

modules designed to handle the automatic computation of the degree of equivalence between pieces of multilingual short-text (Kashyap et al., 2016). The system was developed to handle general short texts segments, however as well as from other datasets, it has been tested on a Twitter news dataset. The system is composed of two main modules, one for calculating the semantic similarity of words and the other for pairs of short-text which includes submodules for text in English and Spanish. The former is the core of the system that computes the semantic similarity based on a combination of HAL and knowledge obtained from WordNet. The semantic textual similarity module manages the multilingual text input and uses the semantic word similarity model to calculate the similarity between pairs of short-text. Two text sequences are represented as two sets of relevant keywords. Keywords similarities are calculated through the word similarity module after aligning multiple terms in one sentence to a single term in the other sentence. The words are then paired and the overall similarity score is computed through the semantic textual similarity (STS) module. Within the HAL algorithm, SVD was applied to the word by context-word matrix and the 300 largest singular values were selected and the 29K word vectors were reduced to 300 dimensions. The HAL similarity between a pair of words is defined as the cosine similarity of their corresponding word vectors after computing the SVD transformation. The word co-occurrence models were based on a predefined English of nouns and noun phrases. Proper nouns were manually excluded and WordNet was used to assign POS tags to the vocabulary words as statistical POS parsers may produce incorrect POS tags to words. Generally, SemSim demonstrated good performance in terms of correlation against human assessment, however, it performed poorly when dealing with informal language such as the case in Twitter. This is attributed to the absence of some words in the dictionary, and the top definitions of other words are not always reliable as they may be less prominent.

Further research aimed at comparing the performance of several models for determining topic coherence in relation to a Twitter dataset with human assessments has been conducted by Fang et al. (2016). Among the utilized models, the approach employed an individual thesaurus and corpus based measures to determine the semantic similarity between terms within extracted topics from the Twitter dataset. The topics were identified through LDA and each topic was represented by the top ten words ranked according to their probabilities in the term distribution. Any two words

from these top ten form word pairs of a topic and the topic coherence is measured by averaging the semantic similarity of all word pairs in that topic. In this approach, the semantic similarity was computed by using individual measures on WordNet and statistical measures on Wikipedia and a Twitter corpus containing 30 million processed tweets.

2.6 Literature Observations on STSS Challenges for Microblogs

One of the most difficult aspects of NLP is to establish the understanding and reasoning of the underlying meaning of the text. The challenge of measuring the semantic similarity increases when there is a reduced quantity and quality of text. In terms of social media data, particularly Twitter, the task becomes much harder due to many inaccuracies that may be present in the short pieces of text. These inaccuracies include:

1. Poor grammatical and syntactical structure due to the character limit which encourage the frequent use of abbreviations and irregular expressions (Alnajran et al., 2017).
2. Misspellings, OOV words, and acronyms.
3. Lots of redundant information as people tend to repost some original messages.
4. Conventions such as hashtags and other metadata that may interrupt the potential meaning in a text.

Due to these inaccuracies, computers face difficulties in understanding the intended meaning or associating the semantic similarity between pairs of tweets. This is especially true in a tweet which expresses sarcasm, such as "I enjoy waiting forever for my appointment", which is common in social media. Therefore, the automation of this process through computation is a challenging task as there are general conventions (hashtags, mentions, URLs, and etc.) and improper English, such as spelling mistakes (e.g. *bcuz* instead of because), shared on this communication platform. Many approaches to STSS measures have been based upon adaptation of existing document similarity methods of general English, with no comprehensive consideration of the language used in Twitter. As such, existing STSS measures are less applicable to the problem domain of Twitter analysis.

Several key points with regard to the challenges of the STSS approach in social media datasets, particularly Twitter, have been observed within this research:

1. Topological-based approaches use ontologies to capture the semantic similarity between concepts. These approaches often demonstrate scalable and acceptable performance, however, when applied in the context of social media, their performance degrades. This is due to the informal terms used in these sites that are absent from these English dictionaries. To minimize this problem, some approaches suggest using external informal dictionaries for dealing with OOV tokens (Liu and Kirchhoff, 2018). However, the research presented in this thesis argues that, this approach may be adequate for less rapidly generated OOV such as named entities, but may be less efficient for the slang words that are often associated with trending topics. This is because the later will require frequent maintenance to the external OOV dictionary in order to keep it up to date.
2. Count-based statistical methodologies are not effective for measuring the semantic similarity for short and sparse text as they are for long and rich text. However, they tend to perform better when the utilized corpus consists of the same domain than the case of general corpus, such as the Brown corpus (Francis and Kucera, 1964). This is because these corpora contain information from traditional media and therefore may fail to capture specific terms and trends dynamically propagated through social media networks.
3. The observed literature around word embedding in the context of Twitter-based semantic textual analysis indicates and reveals potential capabilities of prediction-based statistical approaches for OSN analysis in terms of scalability and computational complexity. However, to the best of the researcher's knowledge, there is not much research conducted in integrating neural embedding models within STSS measures in the context of microblogs, and therefore it is worth further exploration.
4. Although not many hybrid-based systems were developed for the intended approach, it can be observed that these approaches outperform single measures of determining the semantic similarity between short segments of texts. However, they tend to consume high computational resources.

Moreover, it has been observed that a robust pre-processing and feature extractor function that is able to normalize and extract Twitter specific text features may significantly improve the performance of STSS measures in the context of social media data (Duong et al., 2016, Demirsoz and Ozcan, 2016, Gómez-Adorno et al.,

2016).

2.7 Chapter Summary

The critical review of the literature conducted in this chapter demonstrates that traditional STSS approaches fall short when applied to measure the semantic similarities for microblogging posts. This is mainly due to the significant difference between the structural and contextual features of formal English sentences and social media posts such as tweet. Furthermore, state-of-the-art contributions towards measuring similarities in the context of microblogs feature at least one of the following weaknesses:

- Neglecting the contribution of syntactical features, such as common user conventions, hashtags, and special symbols to the overall similarity score.
- Neglecting the contribution of contextual features, such as words and phrases and relying on single features to compute the overall similarity. For example, deriving conclusions on the similarity between candidate tweets based on the common hashtags they share.
- Similarity computations are based on keyword matching of shared words in the candidate posts rather than analyzing the semantic meaning beyond the text.
- Basing their semantic computations on statistical methods that are more suitable for context-rich text segments, such as LSA.
- Basing their semantic computations on lexical resources that are more applicable for short text composed of formal English words (e.g. dictionary definitions), such as WordNet.

Therefore, this research aims to develop a semantic similarity measure for tweets, TREASURE that can be extended to different microblogging posts. TREASURE, which is further described in Chapter 6 and evaluated in Chapter 7, fills the gap and overcomes the weaknesses of STSS measures in the context of microblogging social media.

Chapter 3 – Unsupervised Machine Learning

3.1 Overview

This chapter reviews previous research that has applied various unsupervised algorithms, particularly cluster analysis, to analyse microblogging streams and identify hidden patterns where text is highly unstructured. It provides a comparative analysis on approaches of unsupervised learning in order to determine whether empirical findings support the enhancement of machine learning (ML) applications in the context of online social networks (OSN). The different challenges that hamper the performance of traditional unsupervised algorithms on such data and potential weaknesses of current approaches are discussed.

The main purpose of this chapter is to:

1. Establish a generalized comparison criterion, upon which a systematic review and generalized conclusions are derived.
2. Review various clustering algorithms that are implemented on different features of microblogging textual datasets and investigate their application in the context of microblogging OSN.
3. Compare the reviewed approaches in terms of clustering methods, algorithms, number of clusters, dataset(s) size, distance measure, clustering features, evaluation methods, and results.
4. Discusses the main challenges faced by unsupervised analytical algorithms in social textual data.
5. Highlight potential weaknesses of current clustering algorithms in mining microblogging data.

3.2 The Problem of Cluster Analysis

Cluster analysis is the unsupervised process of grouping data instances into relatively similar categories, without prior understanding of the groups' structure or class labels (Han et al., 2011). It is a prominent component of exploratory data analysis. A subfield of clustering includes text mining, where large volumes of text are analysed to find patterns between documents (Godfrey et al., 2014). The growth of these unstructured data collections, advances in technology and computer power, and enhanced software capabilities, has made text mining an independent academic field.

The problem of clustering has been widely studied owing to the huge amounts of data collected in databases. Several approaches have been proposed to address clustering in the context of various data mining, statistics, and machine learning applications (Jain and Dubes, 1988). For example, in the field of text mining, Hotho et al. (2002) introduced a new approach using *k*-means for ontology-based text clustering in order to improve documents' clustering results. The principle idea of their approach involves generating a set of clustering results automatically for a given input of documents, in which the user may decide to prefer one to the other. This approach has the advantage of producing diverging views of clustering onto the same input, through applying background knowledge. However, their method in text clustering is intended for documents rather than short text. The method narrows the feature space of a document by mapping terms to concepts in an ontology in order to find structure. This may restrict its applicability to documents (e.g. webpages) rather than short text (e.g. tweets) which lack contextual clues and is more challenging due to the sparsity and noise.

Huang and Mitchell (2006) supported the suggestion of user preferred clustering by proposing a novel approach to mixed-initiative clustering that handles several natural types of user feedback. They incorporated user input into automated clustering algorithms to allow the user and computer jointly produce coherent clusters that capture the categories of interest to the user. It is true that the mappings of terms to concepts can provide larger margin of similarity between documents than term-term approaches, however they do not consider semantic distances and relations between these concepts. In addition, as this approach incorporates computers with human beings, it might provide much accurate results compared to autonomous clustering. However, this cooperation comes at a major drawback. The need of manual input is costly (especially when clustering large and unstructured datasets such as social data) and leaves the system handicapped, which does not allow it to make fully automated decisions.

Seifzadeh et al. (2015) applied statistical semantics for short text document clustering and considered the correlation between terms. In this study, the authors applied random sampling and low rank approximation of a term-term correlation matrix to reduce the run time while maintaining the semantic performance. The experiments showed that this application has outperformed *k*-means and spherical *k*-means baseline

methods. However, the effectiveness of their results depends on the selected terms, which may not be representative as they are being selected randomly. The experiments have also shown that using a larger number of terms (rank-10k compared to rank-5k) increases the normalized mutual information (NMI), but this yields a consequent increase in the computation time as well.

Unlike supervised learning which uses labelled training tuples to model each group, clustering analyses data objects where each of their class labels are unknown. Hence, it is considered an unsupervised learning process, which plays a significant role in data mining applications. Clustering becomes desirable when the process of assigning a class label for each tuple in the dataset is costly and infeasible as in large databases. Clustering is defined by Han et al. (2011) as the process of grouping physical or abstract objects into classes, so that objects within a cluster have high similarity in comparison to one another but are very dissimilar to objects in other clusters. Measuring the similarity or distance between two data points is the core body of the clustering process (Boriah et al., 2008). Distance measures are often used for this purpose (e.g. *Euclidean distance*) to assess the similarities between objects and their attributes. Clustering has the advantage of observing useful features that distinguish different groups (Han et al., 2011). For this reason, it is considered a technique of learning by observation rather than by examples, as is the case with classification. Different clustering algorithms exist and each varies in strengths and weaknesses according to the type and complexity of information to be considered. It might not be trivial or handy to identify independent categorization of the available clustering methods as they may overlap. One algorithm may incorporate features from various categories. Nevertheless, the main clustering algorithms can be used with categorical features such as text (Aggarwal and Zhai, 2012).

3.3 Cluster-Based Mining of Microblogs

The emergence of microblogging social networks has yielded new frontiers for academic research, where researchers in the broad area of NLP consider text analysis one of the most important research areas. Recent studies in various disciplines have shown increasing interest in micro-blogging services, particularly Twitter (Sheela, 2016). The applications of text mining tools for studying features of content and semantics in tweets propagating through the network has been widely studied (Kumar et al., 2014). Several studies have aimed at analysing social data from Twitter through

performing data mining techniques such as classification (Castillo et al., 2011). However, these techniques could be considered to have limited capabilities due to the unpredictable nature of the dataset. Cluster analysis of tweets has been reported to be particularly suitable for this kind of data for two reasons (Go et al., 2009a):

1. The amount of data for training is too vast for manual labelling.
2. The nature of the data implies the existence of unforeseen groups that may carry important nuggets of information, which can only be revealed by unsupervised learning.

Among the research conducted around clustering tweets' short-text and other text mining applications on Twitter, researchers aim to find relevant information such as inferring users' interests and identifying emergent topics.

Many clustering methods exist in the literature, and it is difficult to provide a crisp categorization of these methods as they may overlap and share features. Nevertheless, the major clustering methods (Han et al., 2011) and their applications in OSN analysis are reviewed in this chapter. Clustering has been widely studied in the context of Twitter mining. It has been applied to analyse social behaviours in a variety of domains to achieve different tasks, such as tailoring advertisements for groups with similar interests (Friedemann, 2015), event detection (De Boom et al., 2015b), and trending issues extraction (Purwitasari et al., 2015). The subsequent sections focus on the major clustering methods: partition, hierarchical, density, graph, and hybrid, which have been used in to mine microblogging textual data.

3.3.1 Review Comparison Criteria

In this review, a comparison criterion has been established to provide a systematic analysis of the unsupervised learning approaches. This criterion identifies general factors in a cluster analysis problem. Each criterion has impact on others and contributes to the overall performance of the resulting clusters.

Table 3.1 presents a general criterion for a systematic comparison of cluster analysis applications.

Table 3.1 A general comparison criterion for unsupervised learning problems

ID	Criterion	Definition
C1	Problem Domain	The task that the clustering method is required to address. A proper understanding of the problem domain is key to the accurate decision on which unsupervised learning approach to use.
C2	Dataset Size (d*ependent on C1*)	Defines the total number of objects (i.e. data points) to be clustered. No rule-of-thumb exist about the exact dataset size for cluster analysis. Decision on the sample size is a trade-off between efficiency and effectiveness as small datasets lead to uncritical applications while large datasets raise scalability issues.
C3	Feature Set (*dependent on C1*)	An unordered list of unique variables that represent the raw data and used to build a predictive model.
C4	Distance Measure (*dependent on C1, C3*)	A method for quantifying the dissimilarity between points, which determines their cluster belongingness. Hence, *d* is a distance measure if it is a function from pairs of points to reals.
C5	Algorithm (d*ependent on C1-C4*)	An automatic method of assigning data objects into homogeneous groups (i.e. clusters) and ensuring that objects in different groups are dissimilar (Aggarwal and Reddy, 2013). Clustering algorithms are generally distinguished into partition-based, hierarchical-based, density-based, graph-based, and hybrid-based.
C6	Number of Clusters (*dependent on C1, C2, and C5*)	Determines the number of clusters that will be generated. While partition-based algorithms require the number of clusters to be pre-specified, hierarchical approaches allow for selecting the number of clusters after the clustering results has been obtained. Density based clustering does not require either but require specifying the minimum number of points in a neighbourhood. Clustering based on graph theory only requires a predefined distance threshold, which will determine the resulting number of clusters.
C7	Evaluation Method (*dependent on C1*)	An objective or subjective function that validates the extent to which a clustering algorithm achieves the optimal goal of attaining high intra-cluster similarity and low inter-cluster similarity.

Figure 3.1 shows a dependency graph of the cluster analysis comparison criteria defined in Table 3.1. The nodes in this graph represent criteria and the arrows represent dependencies.

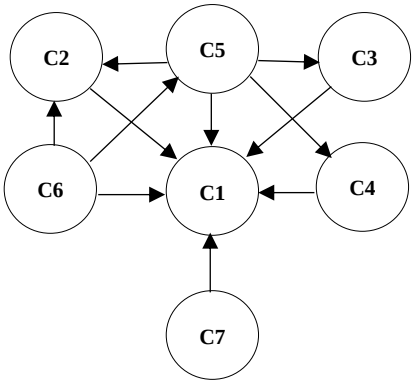


Figure 3.1 Dependency graph of the cluster analysis comparison criteria

The next section provides a background and a critical literature review on the cluster analysis approaches and applications in the context of microblogs.

3.4 Partition-Based Clustering

Partitioning algorithms attempt to organize the data objects into *k* partitions ($k \leq n$); each representing a cluster, where *n* is the number of objects in a dataset. Based on a distance function, clusters are formed such that objects within the cluster are similar (intra-similarity), whereas dissimilar objects lie in different clusters (inter-similarity). Partitioning algorithms can be further divided into hard and fuzzy (soft) clustering. In this section, six articles are summarized in which partitioning-based clustering algorithms has been applied in the exploratory analysis of Twitter.

3.4.1 Hard Clustering

Methods of hard partitioning of data assign a discrete value label (0, 1), in order to describe the belonging relationship of objects to clusters. These conventional clustering methods provide crisp membership assignments of the data to clusters. *K*-means and *k*-medoids are the most popular hard clustering algorithms (Arora and Varshney, 2016).

K-means is a centroid-based iterative technique which takes the number of representative instances, around which the clusters are built. Data instances are assigned to these clusters based on a dissimilarity function (i.e. distance measure). In each iteration, the mean of the assigned points to the cluster is calculated and used to replace the centroid of the last iteration until some criteria of convergence is met. The square-error criterion can be used, which is defined as (Han et al., 2011),

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$$

Equation 3.1 *K*-means square error

Which means that for each data point *p* in each cluster space, the distances from the data points to their centroids are squared and summed. This criterion aims to provide the most compact and separate *k* clusters as possible. *K*-means has been adapted in numerous ways to suit different datasets including numerical, binary, and categorical features.

In the context of microblogging unsupervised applications, the *k*-means approach for clustering customers of a company using social media data from Twitter was proposed (Friedemann, 2015). The technique constructs features from a massive Twitter dataset and clusters them using a similarity measure to produce groupings of users. The study

performed *k*-means clustering and produced satisfactory experimental results. It is considered to be relatively computational efficient.

Soni and Mathai (2015) proposed a 'cluster-then-predict' model to improve the accuracy of predicting Twitter sentiment through a composition of both supervised and unsupervised learning. After building the dataset, *k*-means was performed such that tweets with similar words are clustered together. This unsupervised phase was performed after a feature extraction process. After the clustering phase, classification was done on the same data. The data was divided into training and testing sets, with 70% and 30% of the data respectively. Finally, the Random Forest learning algorithm was used for building the learning model, which was applied to each of the training datasets individually (Breiman, 2001). This algorithm has been chosen as it provides satisfactory trade-off between accuracy, interpretability, and execution time. Empirical evaluation shows that combining both supervised and unsupervised learning (*k*-means then Random Forest) performed better than various stand-alone learning algorithms.

K-medoids is an object-based representative technique that deals with discrete data. It is an improvement to *k*-means in relation to its sensitivity to outliers. Instead of referring to the mean value of cluster objects, *k*-medoids picks the nearest point to the centre of data points as the representative of the corresponding cluster. Thus, minimizing the sum of distances between each object, *o*, and its corresponding centre point. That is, the sum of the error for all objects in each cluster is calculated as (Han et al., 2011),

$$E = \sum_{j=1}^{k} \sum_{p \in o_j} |p - o_j|$$

Equation 3.2 *k*-medoids error

Where k is the number of clusters, p is an object in the cluster C_j, while o_j is the representative object of C_j. The lower the value of E, the higher clustering quality.

A recent study focused on the usage of *k*-medoids algorithm for tweets clustering due to its simplicity and low computational time (Purwitasari et al., 2015). In this study, the author applied this algorithm to extract issues related to news that is posted on Twitter in Indonesia, such as "flight passengers asking for refund". Their proposed methodology for Twitter trending issues extraction consists of clustering tweets with *k*-medoids, in which they divided the tweets dataset into groups and used a

representative tweet as the cluster centre. Terms that are related to topic issues are then selected from the clusters result and assigned higher weight values. The terms that weigh over a certain threshold are extracted as trending issues. Weight score is calculated as the frequency of word occurrences in the dataset. Average Silhouette Width (Rousseeuw, 1987), a method for validating clusters' consistency, was used to measure and evaluate the clustering performance (Ramaswamy, no date). In the work, the experiments demonstrated good results of using *k*-medoids for this purpose; however, re-tweets (i.e. duplicates) had influenced the clustering results. Another study used k-means and *k*-medoids respectively to cluster a single Twitter dataset and compare the results of each algorithm (Zhao, 2012). Initially, *k*-means was applied, which took the values in the term-document matrix as numeric, and set the number of clusters, *k*, to eight. After that, the term-document matrix was transformed to a document-term matrix and the clustering was performed. Then, the frequent words in each cluster and the cluster centres were computed in order to discover the meaning of the cluster centroid. The first experiment showed that the clusters were of different topics. The second experiment was conducted using *k*-medoids, which used representative objects instead of means to represent the cluster centre. However, the resulting clusters tend to be overlapping and not well separated.

Comparing *k*-means to *k*-medoids, the latter has the advantage of robustness over *k*-means as noise and outliers has less influence on it. However, this comes at the cost of efficiency. This is due to the high processing time that is required by *k*-medoids compared to *k*-means. Both methods require the number of clusters, *k*, to be fixed. In terms of clustering sparse data such as tweets, *k*-medoids may not be the best choice as these do not have many words in common and the similarities between them are small and noisy (Aggarwal and Zhai, 2012). Thus, a representative sentence does not often contain the required concepts to effectively build a cluster around it.

3.4.2 Fuzzy Clustering

This partition-based method is particularly suitable in the case of no clear groupings in the data set. Unlike hard clustering, fuzzy algorithms assign a continuous value [0, 1] to provide reasonable clustering. Multiple fuzzy clustering algorithms exist in the literature, however fuzzy *c*-means (FCM) (Bezdek et al., 1984) is the most prominent. FCM provides a criteria on grouping data points into different clusters to varying

degrees that are specified by a membership grade. It incorporates a membership function that represents the fuzziness of its behaviour. The data are bound to each cluster by means of this function.

In the context of Twitter analysis, a recent study presented a simple approach using fuzzy clustering for pre-processing and analysis of hashtags (Zadeh et al., 2015). The resulting fuzzy clusters are used to gain insights related to patterns of hashtags popularity and temporal trends. To analyse hashtags' dynamics, the authors identified groups of hashtags that have similar temporal patterns and looked at their linguistic characteristics. They recognized the most and least representative hashtags of these groups. The adopted methodology is fuzzy clustering based and multiple conclusions were drawn on the resulting clusters about variations of hashtags throughout a period. Their clustering was based on the fact that categorization of hashtags is not crisp, rather, most data points belong to several clusters according to certain degrees of membership.

Another study compared the performance of supervised learning against unsupervised learning in discriminating the gender of a Twitter user (Vicente et al., 2015). Given only the unstructured information available for each tweet in the user's profile, the aim is to predict the gender of the user. The unsupervised learning involved the usage of fuzzy in conjunction with hard clustering algorithms, which are *k*-means and FCM. Both *k*-means and FCM were applied on a 242K Twitter user profiles. The unsupervised approach based on FCM proved to be highly suitable for detecting the user's gender, achieving a performance of about 96%. It also has the privilege of not requiring a labelled training set and the possibility of scaling up to large datasets with improved accuracy.

Comparing fuzzy to hard clustering, experiments have shown that the former is more complex than clustering with crisp boundaries. This is because fuzzy clustering requires more computation time for the involved kernel (Bora et al., 2014). Fuzzy methods provide relatively high clustering accuracy and more realistic probability of belongingness. Therefore, they can be considered an effective method that excludes the need of a labelled dataset. This is particularly useful for large volumes of tweets, where human annotations can be highly expensive. However, these methods generally have low scalability and results can be sensitive to the initial parameter values. In terms of optimization, fuzzy clustering methods can be easily drawn into local optimal

(Khan et al., 2012).

Mukherjee and Bala (2017) approach the problem of sarcasm in microblogs using fuzzy clustering algorithms. The authors worked with a small dataset of 2000 tweets from which they extracted features such as function words, content words, part of speech (POS) tags, POS *n*-grams, and their combinations in an attempt to interpret the linguistic styles of authors in order to detect sarcasm. In their work, the authors hypothesize that sarcasm is based on the author writing style as well as the content of the tweets. They applied fuzzy *c*-means clustering and Naïve Bayes classification and reported that the former is less effectiveness in detecting sarcasm. Another recent unsupervised fuzzy approach in the domain of public health surveillance was proposed by Dai et al. (2017). The authors collected 2,270 tweets through Twitter APIs and manually labelled them to create a benchmark for testing. The proposed word embedding based algorithm assigns a tweet to different clusters of similar words according to the semantic relationships between their vectors. They found that the number of clusters varies per tweet and each tweet typically belong to 3-5 fuzzy clusters. Their results support the view that word embedding is a promising direction for processing microblogging posts.

3.5 Hierarchical-Based Clustering

In hierarchical clustering algorithms, data objects are grouped into a tree-like hierarchy (i.e. dendrogram) of clusters. These algorithms can be further classified depending on whether their composition is formed in a top-down (divisive) or bottom-up (agglomerative) manner. This section reviews three studies that performed hierarchical-based clustering algorithms in applications of Twitter mining.

Ifrim et al. (2014) used hierarchical clustering for topic detection in Twitter streams, based on aggressive tweets/terms filtering. The clustering process was performed in two phases, first the tweets and second the resulting headlines from the first clustering step. Their methodology is composed of initially computing tweets pair-wise distances using the cosine metric. Next, a hierarchical clustering is computed such that tweets belonging to the same topic shall cluster together, and thus each cluster is considered as a detected topic. The tightness of clusters is controlled by "cutting" the resulting dendrogram at 0.5 distance threshold. In this way, they will not have to provide the number of required clusters a-priori as in k-Means. The threshold was set to 0.5 as a

midway between tight and loose clusters. Each resulting cluster is then assigned the score of the term with highest weight in the cluster and ranked according to that score. The top 20 clusters are then assigned "headlines", which are the first tweet in each of them (with respect to publication time). The final step involved re-clustering the headlines to avoid topic fragmentation (also using hierarchical clustering). The resulting headlines are then ranked by the one with the highest score inside a cluster. The headlines with the earliest publication time are selected and their tweet text is presented as a final topic headline.

Another study implemented a hierarchical approach for the purpose of helping users parse tweets results better by grouping them into clusters (Ramaswamy, no date). The aim was for fewer clusters that are tightly packed, rather than too many large clusters. The work involved using a dataset of tweets to see how the choice of the distance function affects the behaviour of hierarchical clustering algorithms. Ramaswamy (no date) conducted a survey of two clustering algorithms that are both hierarchical in nature but differ in the implementation of their distance functions. A total of 925 tweets comprising of various topics with common keyword have been used in the experiments. In the first algorithm, the author considered each of the given objects to be in different clusters. Then determining if the object *o* is close enough to cluster *c*, and if so, add *o* to *c*. This process continues until the maximum size of the desired clusters is reached or no more new clusters can be formed. In this first algorithm, the notion of the distance between an object and a cluster has been defined using concepts from association rule problems –support and confidence. The second algorithm maintained the average distance of an object from each element in the cluster as the similarity measure. If the average is small enough, the object is added to the cluster. Both clustering algorithms involve reading the tweets, tokenizing them, clustering them and returning the clustered output. Although the overall behaviour was found to be similar for both algorithms, the second one seemed to fare better for each of the confidence and support level value.

An integrated hierarchical approach of agglomerative and divisive clustering was proposed to dynamically create broad categories of similar tweets based on the appearance of nouns (Kaur, 2015). In this study, only nouns have been utilized as features as the authors claim they are the most meaningful entities among other part of speech tags, such as verbs, adjectives, and adverbs. Therefore, their approach tends

to discard all sentence tokens but nouns. The adopted bottom-up technique merges similar clusters together to reduce their redundancy, in which a recursive and incremental process of dividing and combining clusters has been applied in order to produce more meaningful sorted clusters. The divisive stage works by dividing clusters down the hierarchy to arrange most similar tweets in different clusters. Afterwards, the bottom-up procedure is applied to remove or merge redundant information, if any. This proposed combinatorial approach showed increase in clustering effectiveness and quality compared to standard hierarchical algorithms. However, due to the problem of tweets' sparsity discussed in Section 3.8, some tweets might lack the presence of nouns to form a rich nouns foundation in the clustering dataset. Therefore, it might be useful to consider other textual features in addition to nouns to enhance the system's performance.

In this context, empirical evaluations provided that hierarchical methods performed slower than hard partition-based clustering, particularly *k*-means (Kaur and Kaur, 2013). Therefore, for massive social media datasets, hard partitioning methods are considered relatively computationally efficient as well as producing acceptable experimental results.

3.6 Density-Based Clustering

This method groups data located in the region with high density of the data space to belong to the same cluster. Therefore, it is capable of discovering clusters with arbitrary shape. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is the prominent density-based algorithm. It grows regions with sufficiently high density into clusters (Ester et al., 1996). In this section, three articles are summarized in which density-based algorithms have been applied in the exploratory analysis of Twitter.

A density-based clustering has been adopted in the context of Twitter textual data analysis to discover cohesively the information posted by users about an event as well as the user's perception about it (Baralis et al., 2013). The provided framework adopts a multiple-level clustering strategy, which focuses on disjoint dataset portions iteratively and identifies clusters locally. DBSCAN has been exploited for the cluster analysis as it allows discovering arbitrarily shaped clusters, and increases cluster homogeneity by filtering out noise and outliers. Additionally, it does not require prior

specification of the number of expected clusters in the data. In this approach, DBSCAN has been applied iteratively on separate dataset portions and identifying clusters locally. All the original dataset is clustered at the first level, and then tweets labelled as outliers in the previous level are re-clustered at each subsequent level. To discover representative clusters for their Twitter dataset, they attempt to avoid clusters containing few tweets. They also attempt to limit the number of tweets labelled as outliers and thus un-clustered, in order to consider all different posted information. Through addressing these issues, DBSCAN parameters were properly set at each level.

A recent study employed DBSCAN as part of its novel method for creating an event detection ground truth through utilizing tweets hashtags (De Boom et al., 2015b). The authors clustered co-occurring hashtags using DBSCAN. The method required setting two thresholds: the minimum number of hashtags per cluster and a minimum similarity measure between two hashtags, above which the two hashtags belong to the same neighbourhood. A collection of clusters of sufficiently co-occurring hashtags on the same day was obtained by running DBSCAN for every day in the dataset.

A recent study has introduced the application of DBSCAN for representing meaningful segments of tweets in batch mode (Anumol Babu, 2016). The segmentation was done based on calculations of the stickiness score. This score considers the probability of a segment being a phrase within the batch of tweets (i.e. local context) and the probability of it being a phrase in English (i.e. global context) (Li et al., 2015). Sentimental variations in tweets were then analysed based on these segments. Each word in the text is assigned a sentiment score according to a predetermined sentiment lexicon. The sentiment of a tweet is then denoted as the summation of the most positive score and the most negative score among individual words in the tweet. In this approach, the core of the clustering consisted of integrating DBSCAN with Jaccard Coefficient similarity function. Empirical evaluations indicated an enhancement of the existing system because of using DBSCAN for clustering.

It can be observed from the literature surrounding Density-based algorithms in Twitter mining, that they are highly efficient and can be particularly suitable for clustering unstructured data, such as tweets, as it allows the identification of clusters with arbitrary shape. Moreover, it is less prone to outliers and noise, and does not require

initial identification of the required number of clusters. However, clustering high data volumes requires a large amount of memory.

3.7 Graph-Based Clustering

These clustering methods are effective in providing results similar to human intuition (Jaromczyk and Toussaint, 1992). Graph-based clustering construct a graph from the set of data and then use the built graph during the clustering process. In these methods, objects are considered as graph vertices and edges are treated in different ways depending on the implemented algorithm (Vathy-Fogarassy and Abonyi, 2013). The graph is a complete graph in its simplest case, and the edges are labelled with the degree of similarity between the objects, which in this case is considered a weighted complete graph. Two articles are reviewed in this section, in which graph-based clustering was utilized in the context of Twitter mining applications.

An approach to graph-based clustering for multi tweet summarization was proposed by Liu et al. (2012), where Twitter-specific features were incorporated to make up for the information shortage in a tweet. In their approach, the number of input varies from hundreds to tens of million tweets. Trending topics were searched and retrieved and a maximum of one thousand English tweet was collected in relation to each trending topic. A set of representative tweets were manually selected from the "gold standard" summarization dataset. This is the optimal data set with human annotations in which the system's output will be evaluated against. It was used for evaluating the proposed graph-based system which showed improvements compared to the LexRank (Erkan and Radev, 2004) baseline. However, these results may not be considered reliable as the manual annotation methodology of the gold standard might be biased.

Dutta et al. (2015), developed a methodology for summarizing tweets based on the graph approach, in which a tweet dataset is taken as input, and a subset of the tweets are derived as the summary of the entire set. This methodology incorporated WordNet (Fellbaum, 1998) to account for the semantic similarities among tweets which may not use common terms to express the same information. Community detection techniques, which detects the existence of non-trivial network organizations (Yang et al., 2016), are then applied to the constructed graph of tweet similarity in order to cluster similar tweets, and the summary includes a representative tweet from each cluster. In their research, the authors collected 2921 tweets related to the flood in Uttaranchal region of India in 2013, through Twitter API. A set of human generated summaries were

obtained for performing evaluations, which were assessed through application of precision (P), recall (R), and F-measure (F).

The main issue in using graph-based algorithms for clustering large Twitter datasets, is that computation of the complete weighted graph consumes lots of resources in terms of time and storage. This complexity can be reduced with several methods. This may be through working only with sparse matrices rather than utilizing the complete graph. These matrices contain information about the small subset of the edges corresponding to higher degrees of similarity. Graphs based on these sparse matrices visualize these similarities in a graphical way. The complexity may also be reduced through the application of Vector Quantization technique, such as *k*-means and Neural Gas (Martinetz and Schulten, 1991), to represent the entire set of objects by a set of representative instances that has a lower cardinality than the one of the original dataset.

3.8 Hybrid-Based Clustering

Hybrid approaches involve integrating two or more of the previously discussed algorithms to perform clustering. The robustness of hierarchical clustering algorithms is relatively high, as they tend to compare all pairs of data. However, this makes them not very efficient due to their high computational demands. On the other hand, partitioning algorithms may not be the optimal choice despite being more efficient than hierarchical algorithms. This is because the former may not be very effective, as they tend to rely on small number of initial cluster representatives. This trade-off has led researchers to propose several clustering algorithms that combined the features of hierarchical and partitioning methods in order to improve their performance and efficiency. These hybrid algorithms include any aggregations between clustering algorithms. In general, they initially partition the input dataset into sub clusters and then construct a new hierarchical cluster based on these sub clusters.

There is not much research conducted using a hybrid clustering approach in the area of Twitter mining. Nevertheless, one approach implemented clustering of keywords that are presented in the tweets using agglomerative hierarchical clustering and crisp *c*-means (Miyamoto et al., 2012). The clustering features were based on a series of tweets as one long sequence of keywords. The approach involved building two datasets, each composed of 50 tweets in different timeframes. Several observations of agglomerative clusters obtained by cutting the dendrogram and *c*-means clusters, with

and without pair-wise constraints were analysed. Better clustering results are provided using pair-wise constraints; however, the size of datasets is relatively small for a generalization.

3.9 Challenges of Clustering Microblogging Posts

Most of the research conducted in clustering tweets, aims to interpret these short-texts through text mining applications to discover relevant and meaningful information that support reasoning on potential conclusions, such as inferring users' interests and identifying emergent topics. However, several natural challenges of such data prevent standard clustering algorithms being applied with their full potentials. These text challenges present in Twitter datasets necessitate intelligent techniques and comprehensive pre-processing stages that depend on the application domain. The incorporation of statistical or ontological semantic techniques should provide dynamic algorithms that can process and analyse such complex datasets and convey meanings and correlations (Alnajran et al., 2017).

3.9.1 Sparseness

Unlike traditional methods of clustering documents, which are performed on rich context, Twitter imposes a textual length restriction of 140 characters. Therefore, users tend to produce short pieces of texts that may be rich in meaning, which implies the usage of abbreviations and other syntactic conventions in order to fit the specified limit.

3.9.2 Out-of-Vocabulary (OOV) Words

The English lexicon is witnessing a high deviation from the formal written version. This is due to the language used in social media, which is mostly driven by new words and spellings that are constantly polluting traditional English. In Twitter, users have invented many ways to expand the semantics that are carried out by the short text. This includes the usage of slang, misspelled, and connected words, besides self-defined hashtags to identify topics or events. These out-of-vocabulary (OOV) words form the primary entities of such language. Examples of word lengthening OOVs include "noooo, pleaseeee, okk, and damnnn", expression OOVs include "haha, uhh, ughh, ahah, and grr", and word shortening OOVs include "lol, omg, yolo, rofl, oomf".

3.9.3 Volume

The rapid generation of user content in Twitter has led to massive volumes of unstructured data, most of which is text. The analysis of these huge streams of data for different applications require high scalability techniques, such as parallel processing, that scale well with the number of data instances. In Twitter, even using the live public streaming API, the maximum sample retrieved is approximately 1% of all tweets that are currently being published by users. Therefore, it is imperative to develop algorithms that work with the data in a scalable fashion.

3.9.4 Credibility

Twitter allows users to instantly report events, news, and incidents acting as social sensors. Therefore, this platform provides first-hand data, however, distinguishing truthful information from rumours and misinformation is one critical problem (Abbasi and Liu, 2013, Derczynski et al., 2017). In most cases, Twitter data is user generated and thus can be subjective, biased, and misleading. In consequence, information propagated in Twitter is not necessarily trustworthy, and therefore means of credibility assessment should be applied prior to decision making.

3.10 Literature Observations

Several approaches of unsupervised learning applications for mining unstructured social media data have been reviewed, following the criterion defined in Section 3.3.1 to conduct a systematic comparison of the unsupervised learning applications in Twitter. The featured surveys are discussed in terms of research approach, clustering method, algorithm, number of clusters, dataset size, distance measure, clustering features, evaluation methods, and results. The seventeen reviewed studies spanning from 2011 to the present in which the clustering of Twitter data was performed in various settings and domains to achieve different business goals or satisfy certain application requirements. The subsequent sections provide a discussion on the studies performing cluster analysis on Twitter in relation to the general comparison criteria defined in Section 3.3.1. The impact of each criterion on the clustering performance is further analysed.

3.10.1 Problem Domain

The clustering approaches in Twitter range from pure clustering perspectives, such as determining the impact of a distance function choice on a clustering behaviour, to a more general pattern recognition application, such as targeting advertisements and event detection. It has been observed that the majority of Twitter-based unsupervised learning applications perform clustering in order to detect news, topics, events, and facts and to predict sentiments. Moreover, there are several different unsupervised ML algorithms that can be used to identify patterns. Therefore, understanding the problem domain is key to deriving the right decision on which clustering algorithm is the most appropriate and will ultimately yield valuable analysis.

3.10.2 Dataset Size

Generally, there is no rule-of-thumb about the optimal sample size for cluster analysis. However, the sample size is expected to be correlated with the number of features (i.e. attributes) and critically evaluated before the cluster analysis is computed. In 2002, a study that explored unsupervised learning segmentation has reported that the smallest sample size detected contains only ten elements while the biggest one contains 20,000 (Dolnicar, 2002). In less than ten years, the massive user generated content in OSN, has led to a dramatic increase in the dataset sizes as observed in the reviewed Twitter-based unsupervised approaches. Among these explored studies, which span the period from 2011-present, the average dataset size detected contains 757,255 tweets, ranging from 50 tweets to 10 million tweets. Moreover, the average Twitter user accounts was found to be 126,329, ranging from 10,000 to 242,658 distinct user accounts. Consequently, this massive increase in datasets raises scalability issues in the performance of unsupervised learning in applications of Twitter predictive analysis. However, the majority of the dataset sizes observed in the surveys are considered relatively small with regard to the high volume challenge of Twitter data. Therefore, scalability issues have not been taken into consideration. Effective unsupervised algorithms are expected to scale well to the massive amounts of Twitter data. In this matter, the scalability (in terms of clustering performance) of most of the algorithms implemented in the surveys is questionable, as these algorithms have not been tested on considerably large datasets.

In relation to dataset sizes and feature set for unsupervised learning, it has been

recommended that the dimensionality is not too high compared to the number of observations to be grouped by the clustering algorithm. Formann (1984) suggests the minimal dataset size should be no less than 2^k objects (k = number of features), preferably $5*2^k$.

3.10.3 Feature Set

The set of variables are extracted from the raw data to form feature vectors that represent the dataset points. The process of feature selection is critical to the performance of the resulting clusters. Depending on the problem domain, these variables can be numerical, categorical, or a combination of both. In Twitter-based unsupervised applications, textual clustering using the common BOW method raises a problem of high dimensionality feature space and inherent data sparsity. This problem will cause scalability issues and the performance of the clustering algorithm will consequently decline dramatically (Aggarwal and Yu, 2000).

Based on the review of existing approaches, it has been observed that different feature sets were used depending on the problem domain. These features include some or all of the following:

- *Hashtags* –31% of the reviewed surveys included hashtags in the features set and considered their impact, 23% treated hashtags as normal words in the text, and 31% removed hash-tags before analysis (excluding the 15% studies that are clustering upon user accounts).
- *Account metadata* – the username, date, status, latitude, longitude, followers, and account followings.
- *Tweet metadata* – the tweet id, published date, and language.
- Maintaining a bag-of-words (BOW) of the unique words contained in each textual data of a tweet and their frequencies as the feature vector. Some included hashtags in the BOW while others ignored them.

Whilst "retweets" and "mentions" conventions in Twitter are claimed to have an impact in boosting tweet popularity (Pramanik et al., 2017), none of the surveys studied the impact of these conventions in assessing the granularities of the unsupervised algorithms in applications of Twitter analysis. Rather, some datasets did not remove the retweeted tweets, which affected the resulting clustering credibility. Because tweets commonly get large number of retweets, keeping them in the dataset

will produce large clusters containing redundant tweets rather than tweets with similar features. This will consequently reinforce false patterns and increase run time. Therefore, it is imperative that the raw data undergo a complete set of pre-processing to ensure that it is ready for the unsupervised learning process with minimal noise possible.

3.10.4 Distance Measure

In clustering algorithms, the results are strongly influenced by the choice of distance measures. It has been observed from the literature that the choice of the selected distance measure is not often justified for Twitter-based unsupervised applications. Euclidean distance is the default for partitioning algorithms, whereas hierarchical algorithms commonly implemented the cosine similarity measure.

However, it is recommended that the distance measure is chosen based upon a thorough understanding of the problem domain and a critical analysis of the feature set. In general, if the magnitude of the feature vector does not matter, cosine is used because it measures the angle between two vectors rather than their distance in the feature space. Thus, it is a measure of orientation and not magnitude. For example, consider a text with the word "sea" appearing eight times and another text with the word "sea" appearing three times, the Euclidean distance between their feature vectors will be higher but the angle will still be small. This is due to the two vectors pointing to the same direction, which is what matters when performing unsupervised learning in the context of Twitter (e.g. clustering tweets). Therefore, it is ultimately important to choose the right distance function for the unsupervised problem under consideration.

3.10.5 Clustering Algorithms

It has been observed from the literature surrounding unsupervised Twitter analysis that partition-based algorithms are used when the problem domain implies knowledge on the granularities present in the dataset. That is, the number of required clusters to be generated is known a priori. Hierarchical algorithms are generally used for topic detection applications where there is lack of knowledge on the themes in the dataset. Density-based methods are used in event detection applications where hashtag features are utilized to identify dense areas in the feature space, which are considered as events (i.e. clusters of arbitrary shapes). Furthermore, it has been observed that graph-based

clustering is used for tweets summarization, in which the algorithm only requires pre-specifying the threshold of similarity between pairs in the dataset.

3.10.6 Number of Clusters

As partitioning algorithms require the number of clusters, *c*, to be pre-specified, *c* has been included in this study to provide a generalized indication on the number of clusters that might be appropriate for similar tasks. From the featured surveys, the average number of clusters maintained is seven, with two as the minimum clusters and ten as the maximum. Generally, the number of clusters, *c*, depends on the target application as large *c* indicates, optimally, fine-grained granularities (i.e. more similarity between data points); whereas small *c* indicates coarse grained granularities, (i.e. more towards topic modelling than pairs semantic similarity).

However, when the number of clusters is unknown, a common practice is to perform an iterative method in order to find the most pure segmentation that provides the minimum intra-cluster variance and maximum inter-cluster variance.

3.10.7 Evaluation Method

Evaluation methods vary from objective measures, such as average silhouette width (ASW) to manual observations, such as manually comparing an algorithm's detected topics with Google news headlines. It can be observed that objective evaluation of clusters quality such as ASW has been utilized by most of the studies in Twitter to measure the clustering performance. Some of the evaluation methods are derived from other data mining techniques such as association rules and classification. These methods calculate precision, recall and the F-measure from a contingency matrix.

In unsupervised text clustering applications, it is generally recommended to perform a subjective evaluation of clusters, as these will reveal the semantic relations between the centroids and the data points in the same clusters and their degree of belongingness. Theoretically, subjective evaluation methods may involve a researcher to acquire an intuition for the results evaluation. However, in practice, the massive amounts of social data and the specific details and variety of vocabulary used in these textual data representations make the intuitive judgment difficult for application over the whole dataset. The existence of a benchmark dataset, which is ideally produced by human judges with a good level of inter-judge agreement, can be used as a surrogate for user judgments. However, this is not always available and can be expensive to generate.

3.11 Chapter Summary

This chapter provides a comprehensive literature review of the problem of cluster analysis and the associated challenges in the context of microblogging textual data.

1. It presents a detailed explanation on the different forms of textual challenges presented in the unstructured data of Twitter. In addition, for each of these challenges, provides different implemented approaches in the literature for alleviating them and discusses their effectiveness. This is extremely important for research, not only in unsupervised learning, but also for other data mining and NLP research that require textual data pre-processing in the context of Twitter analysis.
2. The review established a general comparison criterion for unsupervised leaning in Twitter, which defines each criterion in a cluster analysis problem and associated dependencies. This criteria has been used to conduct a systematic comparative analysis on applications that utilized and tuned unsupervised approaches to the characteristics of Twitter unstructured data.
3. It concentrated on algorithms of the general unsupervised methods: (1) partition-based, (2) hierarchical-based, (3) hybrid-based, (4) density-based, and (5) graph-based, in Twitter mining, and discuss them in the context of Twitter analysis.
4. It provides a comprehensive comparative information and discussion across the dataset size, approach, clustering methods, algorithm, number of clusters, distance measure, clustering feature, evaluation methods, and results.

Seventeen articles were reviewed in this chapter, and the results indicates that there is a sufficient improvement in the exploratory analysis of social media data. However, many of the existing methodologies have limited capabilities in their performance and thus limited potential abilities in recognizing patterns in the data:

- Most of the dataset sizes are relatively small which is not indicative of the patterns in social behaviours and therefore generalized conclusions cannot be drawn. Because of the sparsity of Twitter textual data, it is difficult to discover representative information in small datasets. Therefore, future studies should aim to increase the size of the dataset.
- Some of the algorithms implemented may have provided effective results in terms of efficiency and accuracy. However, this may be attributed to the small size of dataset as the scalability has not been evaluated.

- Some of the reviewed datasets included redundant tweets (i.e. retweets) which yields inaccurate clustering. Therefore, future studies should perform a comprehensive pre-processing phase in which retweets and other noise, such as URLs, are removed from the dataset prior to clustering.
- Most of the studies implemented keyword-based techniques, such as term frequencies and BOW, which ignores the respective order of appearance of the words and does not account for co-occurrence correlations between text segments. Therefore, future research should incorporate and measure the underlying semantic similarities in the dataset.
- In terms of clustering evaluation, objective techniques that measure the granularity compactness, such as ASW, have been applied. However, it is imperative to incorporate subjective procedures to the evaluation process to validate the semantic belongingness and similarities among clusters' data points.

With reference to the comparison criteria discussed in section 3.3.1, general conclusions and recommendations can be made on the state-of-the art unsupervised learning in Twitter:

- *(C1)* –the massive user generated content in microblogs (e.g. Twitter) provide potential value for different applications. The use of unsupervised algorithms for Twitter can reveal hidden patterns due to several reasons as discussed in section 1.
- *(C2)* –the dataset sizes has dramatically increased since 2002 due to huge data volume in Twitter. Hence, for an unsupervised learning algorithm to provide high performance predictions, it requires large datasets. However, this raises scalability issues.
- *(C3)* –depends on the problem domain. Dimensionality reduction methods can be applied carefully when the feature space it too big in order to enhance the performance of the unsupervised learning algorithm.
- *(C4)* –depends on the target application and the representation of features. Empirical experiments can be performed to find the best performing measure for the problem under consideration.
- *(C5)* –the choice of the algorithm is influenced by the dataset size as some algorithms are more efficient in dealing with the massive Twitter data.

- *(C6)* –the experimentation of different clusters to find the best segmentation of the dataset is recommended. However, this does not always translate into good effectiveness in an application and therefore an efficient evaluation criteria is required.
- *(C7)* – Objective evaluation is generally used to evaluate microblogging clusters. However, subjective evaluation criteria using a benchmark dataset is an ultimate evaluation for textual clustering problems. However, if these benchmark are not available, generating a reliable benchmark for the purpose of evaluating clusters can be a labour intensive and expensive task (Schütze et al., 2008).

In conclusion, it can be clearly established that unsupervised learning is an important element of exploratory text analysis in microblogs, particularly Twitter. The unstructured data generated in this microblogging social networking platforms is an important source of information for applications of pattern recognition, knowledge discovery, and identification of user potentials and interests. However, current unsupervised approaches feature several weaknesses in detecting latent semantic themes in microblogging posts. Therefore, the research presented in this thesis aims to fill the gap in the current state of NLP for microblogging posts similarity measurement and semantic-based segmentation. Towards achieving this aim, this research develops a novel similarity measure for tweets, namely TREASURE (Chapters 6 and 7), which is incorporated in a semantic-based cluster analysis (SBCA) algorithm (Chapters 8 and 9) to create an integrated semantic-based framework for detecting meaningful clusters (i.e. themes) in Twitter microblogging posts.

Chapter 4 – Research Methodology

4.1 Overview

Chapters 2 and 3 provided a review of related works in four key areas associated to the research presented in this thesis, including:

- Identification of textual challenges in microblogging online social networks (OSN) compared to the formal English language present in traditional documents.
- Short text semantic similarity (STSS) measures and their applications and adaptation for microblogging posts analysis,
- Statistical-based semantic computations and the potentials of artificial neural embedding models in learning the nature of language used in microblogging platforms.
- Weaknesses of traditional unsupervised learning algorithms to detect semantic themes in large-scale microblogging posts.

This review of literature provided guidance and paves the way towards the development of a novel integrated framework for measuring the semantic similarities between microblogging posts, particularly tweets. The framework will encompass a new STSS measure, known as TREASURE (**T**weet simila**R**ity m**EASURE**), which is described in Chapter 6 and incorporated in a semantic-based cluster analysis (SBCA) algorithm to detect semantic themes within microblogging posts (described in Chapter 8).

This chapter details the research approach undertaken to develop and evaluate TREASURE STSS measure as well as the SBCA algorithm. It describes the research methodology in terms of philosophy, strategy, design, and data collection and analysis. In this chapter, Section 4.2 describes the underlying philosophy upon which the research questions emerged. Section 4.3 discusses the general research strategy and the methodologies adopted at each phase. Section 4.4 describes the methods used in the development and evaluation. Section 4.5 illustrates the data collection and the analysis methods used. Section 4.6 describes the software used in facilitating the various aspects of the research manipulation and visualisation, and Section 4.7 summarises the chapter.

4.2 Research Philosophy

A research philosophy is a belief about the way in which data considering a phenomenon should be collected, analysed and used (Blaxter et al., 2006). The term epistemology (what is *knowledge*) as opposed to doxology (what is *belief*) encompasses the different philosophies of research approaches (HOLSTEIN, 1994). The purpose of conducting a scientific research, then, is the process of transforming *believes* (doxa) into *knowledge* (episteme). Two major research philosophies have been recognized in the Western tradition of science, namely *positivist* and *interpretivist* (Creswell and Creswell, 2017).

Positivist researchers assume that reality is stable, directly measurable, and observable and that there is just one truth, one external reality (Levin, 1988). Positivism adheres to the view that only "factual" knowledge gained through observation, including measurement without bias using standardized instruments, is trustworthy. This group argue that phenomena should be isolated and that observations should be repeatable. This often involves manipulation of reality with variations in only a single independent variable in order to derive relationships between, some of the constituent elements of the social world.

In contrast, interpretivist researchers accept that there is a reality but argue that it cannot be measured directly, only perceived by people, each of whom views differently, based on prior experience, knowledge, and expectations. Interpretivists claim that there may be many interpretations of reality, and that these interpretations are in themselves a part of the scientific knowledge they are pursuing (Blaxter et al., 2006).

4.2.1 Rational for Choice of Research Approach

The researcher's concern is that the undertaken research methodology should be both appropriate to the research questions, as defined in Chapter 1, and rigorous in its operationalisation. Ultimately, the researcher believes that a positivist philosophy is required for this purpose, i.e. implementing close-end questionnaires to gather and quantify humans' subjective perceptions on similarities and classification of natural language text. This research depends on quantifiable observations that lead to statistical analyses to test the informed guesses (i.e. *hypotheses*) about what the findings will be. Thus, it commences with a deductive approach in which a hypothesis is developed upon reasoning with a theory and then a research strategy is designed to

test the hypothesis. This hypothesis is tested by confronting it with observations that either lead to an acceptance or a rejection of the hypothesis. The various elements of the research approach are further elaborated in the subsequent sections: Research Strategy, Research Design, and Data Collection and Analysis.

4.3 Research Strategy

This research is exploratory in nature; it explores the subject areas to induce the development of knowledge. In this section, the researcher identifies and justifies the choice of methodologies and explains how they operate and interoperate in each stage.

4.3.1 Build Methodology

The research commences with a "build" methodology to develop a software artefact. This artefact is a novel framework of a semantic-based cluster analysis for microblogging posts integrating a new similarity measure. This methodology involves an overall design from the abstract level of architecture components down to the low level of code modules. A plan is also designed for testing and evaluating the built algorithms in order to answer the research questions. Furthermore, investigations of various programming languages that share similar functionalities, such as MATLAB[1], were undertaken and the choice of Python (Sanner, 1999) as an adequate programming language was made upon several considerations:

- Unlike MATLAB, Python is open source, which makes it freely usable and distributable and therefore, the code can run everywhere.
- Compared to MATLAB, Python has broader set of libraries that facilitate text manipulation.
- Expressive in nature, which makes Python easily readable and understandable.
- Interpreted programming language that executes the code line-by-line.
- Cross platform compatibility that can run on different platforms such as Windows and Linux.
- Python is an object-oriented language.

MATLAB These factors are important for the development of the algorithms intended to answer the main research questions.

[1] https://uk.mathworks.com

4.3.2 Model Methodology

This research involves a "model" methodology (Elio et al., 2011) in different stages of its design and development. This methodology defines an abstract model for a more complex system, and therefore allows the researcher to use the model to perform experiments that could not be performed in the system itself because of cost or accessibility. The development of the semantic similarity measure, TREASURE (described in Chapter 6) involved modelling words co-occurrences in a corpus using an artificial neural network. The model is empirically tested and used to derive semantic relationships between words. Furthermore, a triangle geometry model is used in designing the cluster analysis algorithm. This model is used to map all the cases in a local optimal solution implemented to compute clustroids.

4.3.3 Experiment Methodology

An "experiment" methodology is used to evaluate the novel built approach in two phases: 1) an exploratory phase where the researcher takes measurements to identify the questions that should be asked with regard to the algorithm under evaluation, and 2) an evaluation phase that attempts to answer the research questions.

According to the research objectives, the researcher intend to develop a new similarity measurement, used to detect semantic themes in microblogging posts. Towards determining both how the measure performs in relation to human typical cognitive perceptions of similarities, and, later on, how well this measure contributes in detecting meaningful clusters, the researcher needs an instrument that enables quantifying the evaluation results.

A questionnaire is a key data collection device. The use of questionnaires to formulate a subjective control was made as they allow a researcher to study different variables at one time than is typically possible in other methods. A key drawback is that it is difficult to recruit relevant participants to undertake the experiment. Moreover, bias may be introduced by possibly self-selecting the nature of participants, the point in time when the questionnaire is conducted, and in the researcher him/herself through the design of the questionnaire itself.

In this research, the researcher attempts to avoid bias as much as possible through:

- Identifying a sampling criterion that identifies a group of participants sharing similar characteristics.

- Designing a methodology for selecting the data (i.e. questions) in which participants are asked to classify and judge for similarity.
- Designing a well-established set of instructions to ensure a thorough and uniform understanding of the task.
- Distributing the questionnaires over close timeframes and having participants conduct the questionnaires without supervision.
- Undertaking statistical reliability tests over the acquired responds to ensure a good level of inter-judge agreement is attained.

In order to answer the research questions, the researcher designed close-end questionnaires to gather human judgments on similarities and classifications of tweets. These questionnaires enabled the researcher to obtain the required data upon which quantitative analytical techniques are used to draw inferences from this data regarding correlations and accuracies. The statistical results of the experiment methodology shall provide the validity of the research in its answer to the research question.

4.4 Research Design

The research presented in this thesis has multiple objectives for the NLP research community:

1. Research current STSS measures based on lexical taxonomies and STSS measures based on statistical probabilities from textual corpora in order to develop a novel similarity measure for microblogging posts that is unique and addresses the research challenges in the field.
2. Undertake a review of unsupervised learning algorithms and gaps in current applications of conventional cluster analysis algorithm to analyse microblogging posts.
3. For a chosen domain (Politics), create a corpus of streamed and pre-processed posts, and train an artificial neural network model to learn distributed word representations from that corpus.
4. Design and implement an architecture for a semantic similarity measure for tweets (STSS), which can be extended to other microblogging social media platforms.
5. Design an experimental methodology to conduct intrinsic evaluation of the developed STSS with reference to human judgement and to assess its validity for capturing the semantic similarities in microblogging posts.

6. Design and implement a new clustering algorithm (SBCA) using the new STSS measure to detect semantic themes within microblogging posts.
7. Design a subjective experimental methodology to evaluate the generated clusters through conducting an experiment to produce a reliable multi-class benchmark dataset of tweets belongingness to clusters for the evaluation of the SBCA algorithm.

4.4.1 Development of TREASURE STSS

TREASURE (development described in Chapter 6) is a tweet semantic similarity measure, which is composed of semantic and syntactic components. It captures the semantic relationships between posts published in Twitter, the most popular microblogging OSN. Based on the research conducted into the development of Twitter-based STSS and the challenges and NLP complexities of the informal language used in social media and lack of benchmark resources, there are not much research conducted to measure the semantic similarities between tweets. Most existing studies tend to extract abstract features from microblogging posts and ignore the contribution of structural and syntactical features. In addition, studies that implement semantic similarities for microblogs often follow the topological semantic approach used in measuring similarities for traditional text documents and formal English sentences. This approach falls down when attempting to measure short texts found in OSN due to the high rate of OOV words that do not exist in hierarchical taxonomies. Consequently, an artificial neural network was trained to generate word vectors that learn distributed representations of words based on their co-occurrences in a large corpus of microblogging posts. The produced pre-trained model demonstrates a core component in the semantic module of TREASURE, from which the words similarities are derived. In terms of OSN research, Twitter has been focused on mainly as it is considered the most popular microblogging platform in the meantime. Furthermore, despite the international spread and popularity of Twitter with tweeters from all over the world, this research focuses on the English language among other western and eastern languages. This is due to two reasons:

1. The high volume of English lexical resources and development packages such as WordNet, NLP libraries such as NLTK, and textual corpora such as the Brown corpus.

2. The mature level of research achieved in the English literature in different areas of research related to this thesis interest.

A preliminary experiment was conducted to evaluate, assess, and compare the viability of different existing STSS approaches in the context of Twitter microblog. The review of literature and preliminary experiment revealed the prediction-based statistical semantic approach (discussed in Chapter 2) potentials for microblogging posts as it caters for the informal language used in OSNs. Furthermore, the hybrid architecture of semantic and syntactic similarity computation is considered as a promising approach with NLP because it combines different textual features and weighs them according to their contributions to the overall similarity. Therefore, in this research, a hybrid approach of semantic and syntactic components was used to design and develop TREASURE, which implements a statistical semantic module to compute the semantic relationships between words.

TREASURE STSS measure was developed through incremental stages with the following main features:

- A new heuristic-based pre-processing methodology to transform raw microblogging posts into semantic-rich, less noisy text, while maintaining their structural features and identity for similarity analysis. For example, Twitter common conventions such as hashtags and mentions are retained.
- A novel similarity measure, which is composed of semantic and syntactic components in order to capture representative set of features to compute the overall similarity score.
- Ability to extend to other microblogging platforms and generalize to different domains.

Details of the TREASURE design and development methodology are present in Chapter 6.

4.4.2 Evaluation of TREASURE STSS

Following its development, TREASURE was evaluated through two phases (evaluation described in Chapter 7). The first is an intrinsic evaluation that was performed by assessing its correlation with reference to similarity benchmarks and inferential statistics to test the subsequent hypotheses and their questions, which address the first main research question outlined in Chapter 1.

Hypothesis A: A statistically significant correlation exists between TREASURE and human similarity judgments:

QuestionA.1: Can TREASURE provide similarity measures that approximate human cognitive interpretation of similarity for microblogging posts?

Hypothesis B: TREASURE can be generalized to different microblogging domains:

QuestionB.1: Does TREASURE demonstrate a performance degradation when applied to a different domain?

Hypothesis C: TREASURE achieves the highest correlation to human judgments among existing measures:

QuestionC.1: Does TREASURE demonstrate a statistically significant correlation to human judgments with regard to existing STSS methods in the context of microblogs?

Human raters whose first language is English and were educated to a graduate level or above (further justified in Chapter 7 Section 7.3.3) were targeted for providing similarity judgments on pairs of tweets to produce a ground truth benchmark. In order to evaluate the validity of TREASURE against typical human cognitive approximation of similarity and make reasonable conclusions, it is important to have reliable benchmark annotations. The level of inter-judge agreement was assessed through undertaking a statistical reliability test.

A further extrinsic evaluation that was performed through monitoring the performance of TREASURE in an end application, which is the SBCA algorithm. TREASURE represent a core component of the SBCA algorithm, which is the proximity measure. The subjective evaluation of the generated clusters, and whether they share meaningful relations not only assesses the SBCA algorithm's performance, but also validates TREASURE as the proximity measure. Intrinsic and extrinsic evaluation methodologies were used to evaluate TREASURE. Details of the evaluation results as well as reliability statistical test analysis are provided in Chapter 7 and Chapter 9.

4.4.3 Development of the SBCA Algorithm

SBCA is a novel Semantic Based Cluster Analysis algorithm that aims to detect semantic themes in microblogging posts (development described in Chapter 8). SBCA is a linear clustering algorithm that uses TREASURE to compute the pairwise distance between dataset instances. It traverses the dataset and assigns instances to clusters

based on a distance threshold derived upon empirical experiments. Existing approaches to cluster microblogging posts often apply traditional clustering heuristics and algorithms such as *k*-means, which fall short for the challenges and nature of the textual data generated in OSNs. Furthermore, most clustering applications that exist in the literature perform unsupervised learning based on specific features extracted from the text. For example, clustering tweets based on the hashtags they contain, community detection by clustering users based on the trending hashtags they often use, and clustering tweets based on their polarity (i.e. sentiment analysis). The problem of detecting semantic clusters (i.e. themes) in microblogging posts through analysing the underlying meanings is an NLP and ML interrelated problem. This research develops a novel framework that integrates intelligent technologies to detect semantic themes in Microblogs, which may have significant impact to the research community. The SBCA algorithm was developed with the following main features:

- A novel proximity measure, which is TREASURE STSS measure to compute the semantic pairwise distances between Twitter posts, and can be extended to other microblogging platforms.
- A semantic based algorithm, which implements linear clustering with complexity O(*n*) in order to scale further for larger datasets.
- Fully unsupervised, which does not require determining the number of clusters beforehand, rather instances are assigned to clusters is performed based on a distance threshold that was derived upon empirical experiments.
- SBCA can be adapted to different applications by increasing or decreasing the distance threshold to generate loosely or tightly coupled clusters.

Details of the SBCA algorithm design and development are present in Chapter 8.

4.4.4 Evaluation of the SBCA Algorithm

Following its development, SBCA was evaluated through subjective evaluation criteria with reference to a multi-class benchmark dataset in order to answer the questions associated with the second main research question outlined in Chapter 1.

Question 1: Can the SBCA algorithm generate pure clusters?

Question 2: Can the SBCA algorithm generate accurate clusters by undertaking correct separation and combining decisions with reference to a benchmark?

Towards addressing these questions, an external evaluation criteria (Schütze et al.,

2008) was undertaken with reference to a multi-class benchmark using the following metrics:

- *Purity* –a measure that tests the extent to which a cluster contains a uniform class.
- *Rand Index* –accuracy measure that computes how similar the generated clusters are with regard to the benchmark classifications.
- *Precision (P)* –the fraction of detected class members that were correct (combined documents that are similar).
- *Recall (R)* –the fraction of actual class members that were detected (similar documents that are combined).
- *F-Measure* –a harmonic mean of precision and recall used to balance the contribution of false negatives by assigning more weight to recall.

The ground truths in the multi-class benchmark were obtained by participants whose first language is English and educated to a graduate level or above. The participants were asked to classify a set of microblogging posts into their relevant classes. A statistical test was performed on the participants' judgments to assess the reliability of the produced benchmark. Details on the external evaluation criterion for SBCA and the corresponding reliability statistical test analysis are available in Chapter 9.

4.5 Data Collection and Analysis Method

This research study employed quantitative methods in order to answer the main research questions, defined in Chapter 1. Using quantitative methods implies systematic empirical investigations to provide evidence supported via statistical, mathematical, and computational techniques. The quantitative methodology includes data from TREASURE (estimated) similarity results (Chapter 7), data from SBCA generated clusters (Chapter 9), and the questionnaires that were conducted to gather human judgments (actual) on similarities and classifications (Chapters 7 and 9).

4.6 Research Facilitation Software

Various software packages were used in undertaking different stages in this research. They feature a long developmental history and runs on the Windows platform that is standard to the operating environment with which the researcher is familiar. The researcher's choice of software has been affected by a number of considerations.

1. **Data collection and storing**: the data collection was performed in a remote Linux machine server, which run a data collection script using Twitter Streaming API. The streamed microblogging posts were stored in MongoDB –a NoSQL non-relational database. Details on data collection are further elaborated in Chapter 5.
2. **Programming language and development software**: Python shell was used for implementation due to the reasons outlined in Section 4.3.1.
3. **Evaluation and interpretation**: Statistical Package for the Social Sciences (SPSS) is, arguably, the most widely used software for statistical analysis. The required quantitative analysis was done with the aid of both SPSS and Microsoft Excel to get the results which were analysed.

4.7 Chapter Summary

This chapter presents the methods used to develop a novel semantic-based framework for microblogging cluster analysis (SBCA) which integrates a new similarity measure (TREASURE). It describes the research methodology, in terms of the research philosophy, strategy, design, data collection and analysis, and the instruments and software that were followed in conducting this research.

The research undertakes a positivist philosophy towards testing the hypothesis and addressing the main research questions. The methods to enable development of the research objectives were made through a two-step process. The first is to design and develop a semantic similarity measure for microblogging posts (TREASURE). The second process involved developing a cluster analysis algorithm (SBCA), which integrates TREASURE to detect semantic themes in microblogging posts.

To evaluate the components of the developed framework, a quantitative method of data collection and analysis was used. The data gathered from questionnaires were compared to the system's output and statistically analysed using SPSS to derive evidence and draw conclusions.

Details of the development and evaluation of TREASURE TSS measure is described in Chapter 6 and Chapter 7 respectively. The development and evaluation of SBCA algorithm are presented in Chapter 8 and Chapter 9 respectively.

Chapter 5 – Data Collection and Pre-Processing

5.1 Overview

This chapter presents the methodology undertaken to collect, store, and construct a dataset from the Twitter microblogging platform in the particular domain of politics. It provides a description of the dataset in terms of size and utilised feature set. Throughout this thesis, this dataset will be referred to as the EU Referendum dataset. This chapter describes and evaluates a new pre-processing heuristic developed for short text semantic similarity (STSS) measures. This heuristic processes raw microblogging posts through different natural language processing (NLP) stages before being transferred to the different component in the novel semantic-based framework.

Furthermore, this chapter describes the SemEval-2014 STS.tweet_news (Guo et al., 2013) Twitter-based dataset as to demonstrate the generalizability of the developed framework and its subsequent components. This general news tweets domain is used to illustrate and evaluate the pre-processing methodology.

In this chapter, Section 5.2 provides a brief introduction to the Twitter streaming Application Programming Interface (API) (Boicea et al., 2012) that was utilised in this research. Section 5.3 describes the non-relational database used to store the unstructured data. Section 5.4 demonstrates the data collection process in a particular domain (politics), provides a description on size, and attributes for the datasets considered in this research. Human similarity judgements will be gathered for the political tweets dataset through an experiment that is covered in Chapter 7. Section 5.5 emphasizes the importance of pre-processing and the drawback of using a general pre-processing methodology. Section 5.6 describes the new pre-processing heuristic developed for STSS measures. Section 5.7 discusses an evaluation experiment conducted to demonstrate the effectiveness of the new methodology compared to a baseline, which is a standard set of pre-processing stages that are generally applied as a reuse component in NLP applications. Section 5.8 illustrates the semantic and syntactic features extracted from a tweet. These features are used to generate the representative semantic and syntactic vectors consequently (detailed in Chapter 6). Finally, Section 5.9 summarises the chapter.

5.2 Twitter Streaming API

The Twitter API provides a streaming mechanism for establishing a connection and continuously streaming real time tweets according to a certain set of search terms. Communicating with the Twitter platform was made possible via the open authentication (OAuth) mechanism. This mechanism requires an application registration on the Twitter platform beforehand. Kumar et al. (2014) provides a comprehensive overview of the authentication process required by the Twitter API. Twitter streamed instances are returned as JavaScript object notations (JSON) data structures, which are composed of multiple metadata per tweet. These JSON objects were stored in a NoSQL database called MongoDB (Banker, 2011).

5.3 MongoDB NoSQL

MongoDB is a fully scalable non-relational database, intended for storing unstructured data, such as text, as documents instead of tuples in tables. It has been trusted by several web 2.0 big data sites such as Foursquare, Disney Interactive Media Group, The Guardian, GitHub, and Forbes (Boicea et al., 2012). The entire 1.2TB text corpus of Wordnik (Davidson, 2013) is also stored in over five billion MongoDB records.
While structured data is usually maintained in relational databases and schemas, features of natural text data require special means of management and storage due to lack of structure. In the context of this research, these unstructured data are the tweets JSON objects that were returned by Twitter streaming API.

```
client = MongoClient('localhost', 27017)
db = client['twitter_db']
collection = db['twitter_collection']
tweet = json.loads(data)
collection.insert(tweet)
```

Figure 5.1 The script for streaming a JSON object and inserting in MongoDB

These objects are inserted into MongoDB using the script shown in Figure 5.1 for streaming JSON objects from the API and storing them in a MongoDB database.

5.4 Building the EU Referendum Dataset

In this research, the political domain of the EU Referendum is considered, as it has been an active trend in OSNs and a rich source of controversial views. The United Kingdom European Union Membership (known as EU Referendum) took place on the 23rd of June 2016 in the UK. Based on a voting criteria, the voters were exposed to

two opposing campaigns supporting remaining or leaving the EU. Three months prior to the day of the referendum, the data collection process has commenced using Twitter API, and lasted until one month past that day. To build the tweets corpus relevant to the aforementioned domain, the following search terms have been incorporated in the *keywords* attribute of the API to formulate the following query:

Keywords = ("*EU*" AND "*stay*") OR ("*EU*" AND "*leave*") OR ("*vote*" AND "*remain*") OR ("*vote*" AND "*leave*") OR ("*Britain*" AND "*remain*") OR ("*Britain*" AND "*leave*") OR "*Brexit*" OR "*EUReferendum*" OR "*StrongerIN*" OR "*strongerOut*", Languages = *English*.

Following the aforementioned data collection methodology, a dataset of 4 million tweets, referred to as the "EU_Referendum" dataset, has been constructed and stored in MongoDB. Each instance in the dataset is a tweet associated with multiple metadata. These metadata contain information relating to the tweet, users, and entities. Figure 5.2 shows an example of one tweet and all the associated metadata in a JSON object.

The restrictions on using Twitter public data in research is detailed in the "Developer Agreement and Policy" report (Twitter International Company, 2018). Each published tweet is associated with all the attributes shown in Figure 5.2 of descriptive information (features). After insertion of the JSON object into the database, any of these metadata (i.e. attributes) can be queried and processed. A list of these metadata and their descriptions can be found in Appendix A.

```
"favorited": false, "contributors": null, "truncated": false, "text":
"(via @FullFact) #Politics What is the single market? -Putting it
simply, the aim of EU rules is to make it as...https://t.co/IdjFN2d0FZ",
"possibly_sensitive": false, "is_quote_status": false,
"in_reply_to_status_id": null, "user": {"follow_request_sent": null,
"profile_use_background_image": true, "default_profile_image": false,
"id": 106715844, "verified": false, "profile_image_url_https":
"https://pbs.twimg.com/profile_images/706521440649142272/UTHdEFWe_normal
.jpg", "profile_sidebar_fill_color": "252429", "profile_text_color":
"666666", "followers_count": 1633, "profile_sidebar_border_color":
"181A1E", "id_str": "106715844", "profile_background_color": "1A1B1F",
"listed_count": 42, "profile_background_image_url_https":
"https://abs.twimg.com/images/themes/theme9/bg.gif", "utc_offset": 0,
"statuses_count":8258, "description": "Welcome to my twitter profile.
All views are my own and re-Tweets are not endorsements.",
"friends_count": 589, "location": "Notting Hill, London, UK",
"profile_link_color": "2FC2EF", "profile_image_url":
"http://pbs.twimg.com/profile_images/706521440649142272/UTHdEFWe_normal.
jpg", "following": null, "geo_enabled": true, "profile_banner_url":
"https://pbs.twimg.com/profile_banners/106715844/1431173637",
"profile_background_image_url":
"http://abs.twimg.com/images/themes/theme9/bg.gif", "name": "Marc
Edgeley", "lang": "en", "profile_background_tile": false,
"favourites_count": 109, "screen_name": "MarcEdgeley", "notifications":
null, "url": null, "created_at": "Wed Jan 20 13:38:55 +0000 2010",
"contributors_enabled": false, "time_zone": "London", "protected":
false, "default_profile": false, "is_translator": false},
"filter_level": "low", "geo": null, "id": 707178221192744960,
"favorite_count": 0, "lang": "en", "entities": {"user_mentions": [{"id":
80862758, "indices": [5, 14], "id_str": "80862758", "screen_name":
"FullFact", "name": "Full Fact"}], "symbols": [], "hashtags":
[{"indices": [16, 25], "text": "Politics"}], "urls": [{"url":
"https://t.co/IdjFN2d0FZ", "indices": [115, 138], "expanded_url":
"http://ht.ly/3cbyQI", "display_url": "ht.ly/3cbyQI"}]},
"in_reply_to_user_id_str": null, "retweeted": false, "coordinates":
null, "timestamp_ms": "1457439401325", "source": "<a
href=\"http://www.hootsuite.com\" rel=\"nofollow\">Hootsuite</a>",
in_reply_to_status_id_str": null, "in_reply_to_screen_name": null,
"id_str": "707178221192744960", "place": null,  retweet_count": 0,
"created_at": "Tue Mar 08 12:16:41 +0000 2016", "in_reply_to_user_id":
null
```


Figure 5.2 A sample JSON object tweet

The dataset of raw tweets has undergone several pre-processing stages following a new heuristic-based methodology developed for STSS, which is described in Section 5.6. This methodology aims to eliminate the unwanted noise such as redundant tweets (retweets) and tweets containing no text, while preserving its identity as a tweet, such as hashtags. The pre-processing has significantly reduced the dataset by x3, from four to one million instances. A sample of the collected data is provided in Appendix B (only the text field is shown to save space).

5.5 The Role of Pre-processing

Pre-processing techniques play a significant role in text mining algorithms. These techniques are required in various information systems in order to maintain data

quality. The unstructured text generated in microblogs is highly susceptible to noise, redundancy, and inconsistency as they are generated from heterogeneous sources. Therefore, a mechanism for removing noise and inconsistencies is imperative because performing analysis on low-quality data will inevitably produce low-quality results Ciszak (2008). The focus of this research is on analysing microblogging posts, particularly tweets, where the majority are erroneous (i.e. misspelt) and highly unstructured, due to the informal nature of the communication channel. Hence, in order to build better NLP and machine learning (ML) algorithms, it is necessary to work with clean data. Towards achieving this goal, these data need to undergo several pre-processing stages. The cleaning process aims at reducing confusion during the execution of an algorithm as much as possible. For example, an algorithm that maps a tweet's semantic features to a language model in which no hashtags are present will not be able to recognize these hashtags in order to map them to their actual words representations if no pre-processing was performed to remove the hash sign. Therefore, the pre-processing stages aim to produce feature sets with minimal irrelevant data in order to eliminate noise introduced to NLP and ML applications (such as STSS measures and cluster analysis algorithms).

5.5.1 Drawbacks of Reusing a General Pre-processing Methodology

Pre-processing is a primary factor contributing to the pureness of an extracted feature set, and thus accuracy of the produced results. A major problem has emerged as pre-processing becomes a reuse component that is not being adapted to the target application. Consequently, the analysis may fail to generate expected results because the data has not been properly processed in the previous stage (Angiani et al., 2016, Kannan and Gurusamy, 2014, Jianqiang and Xiaolin, 2017). For example, in the context of Twitter analysis, one may apply a pre-processing heuristic that works well for a sentiment analyser in a semantic similarity identification task. Intuitively, this will reduce the performance of the latter task due to the persistent noise from the perspective of the algorithm under consideration. This problem is particularly common in applications of STSS measures (Satyapanich et al., 2015, Zhang and Lan, 2014, Sultan, 2016) employing one or more of the following pre-processing pitfalls:

- Following common practices for data scrubbing such as tokenization, part-of-speech (POS) tagging, stemming, lemmatization, etc. and regardless of the required feature set and target application. As an example of application-based

pre-processing, retaining terms with repeated characters is of high value for sentiments analysis applications, but should be normalized to their standard forms for STSS applications in order to map to a vocabulary for interpretation.

- Preforming a crude and comprehensive pre-processing steps, which result in discarding important information and consequently, losing the identity of tweets. Stemming and removal of function words, abbreviations, punctuations, numbers, hashtags, mentions, URLs, and emoji altogether from very short text such as tweets will result in loss of information nuggets that may altogether contribute in the overall meaning of a tweet (Li et al., 2006).
- Performing inadequate pre-processing steps, which retain unwanted noise in the data. For example, failing to remove redundant data such as re-tweets when performing cluster analysis will result in false clusters (Alnajran et al., 2017).

Therefore, this research develops and evaluates a new heuristic-based methodology for the pre-processing of data for the novel STSS measure, known as TREASURE (described in Chapters 6), proposed in this research. The methodology can be adapted to other STSS measures in the context of microblogs. The steps undertaken in this methodology are described in the subsequent sections.

5.6 The STSS Pre-Processing Heuristic

A heuristic is a problem solving approach that employs a set of consecutive rules. In this research, a set of pre-processing rules are integrated to transform tweets from their raw noisy form to a semantic-rich form to be processed by the STSS measure. In TREASURE, a tweet is processed as a representative feature vector. These vectors are derived from raw tweets after undergoing pre-processing. Towards extracting effective feature sets for TREASURE, this research implements a novel heuristic-driven comprehensive list of pre-processing practices. This heuristic is composed of consequent rule-based processing steps that aims to generate condense and semantic-rich tweets for which representative feature vectors can be derived. The sequence of the steps implemented in this pre-processing methodology was identified using empirical experiments. The subsequent sections describe the steps undertaken for processing Twitter feeds before they are transferred to the feature extraction and then STSS measure for similarity computation.

5.6.1 Decoding

This form of processing consists of transforming the text into a simple machine readable format. Text may exist in diff erent formats such as *Latin*, *UTF-8*, etc. For an STSS measure to perform internal computations, it is necessary to format text consistently in a standard encoding format. It is generally recommended to use UTF-8 as it is widely accepted.

5.6.2 Retweets and URLs Removal

In Twitter, the "retweet" option allows users to share other user's tweets, which consequently generate redundant information. Retweets are therefore removed from the dataset for two reasons:

1. Retaining them in the dataset will result in an increased feature space.
2. Introducing bias when transforming the dataset into a corpus to compute information contents of terms. Distinctive terms that carry rich meaning will contribute less to the similarity score because they appear in retweets and thus weigh less, yielding misleading results.

Uniform resource locators (URLs) are common in Twitter where users refer to articles, videos or images. In STSS measures, the task involves measuring the similarity between the short texts. URLs introduce noise to the similarity and thus are removed from tweets, although URLs may be utilized for tasks related to word sense disambiguation, which will be further investigated in future work.

5.6.3 HTML Tags Conversion

Lots of html characters such as *< > &* are embedded in the original data retrieved from the web. This research employs regular expressions to convert these tags to their standard html formats. For instance, & is converted to "*and"*. Python provides some packages and modules such as *htmlparser* that facilitate this conversion.

5.6.4 Tokenization

The *n*-gram language model (Brown et al., 1992) is the basic building block in constructing a feature vector. For the TREASURE STSS measures, tweets are transformed into tokens of unigrams and bigrams (*n*-gram, *n*=1 and *n*=2).

5.6.4.1 Unigrams

The natural language toolkit (NLTK) tokenizer is used instead of the Stanford

tokenizer (Manning et al., 2014) because the former is familiar with Twitter conventions and emojis, and therefore will not split hashtags or emoticons. An example of the NLTK and the Stanford tokenizers for a tweet, *T*, is illustrated in Table 5.1.

Table 5.1: Different tokenization of a sample tweet, *T*

Sample tweet (*T*)	**NLTK tokenizer**	**Stanford tokenizer**
voting results *#Remain 44% #Leave 46%*	'voting' 'results' '#Remain' '44%' '#Leave' '46%'	'voting' 'results' '#' 'Remain' '44' '%' '#' 'Leave' '46' '%'

It can be observed that the NLTK tokenization scheme produces logical tokens in terms of twitter-based features and conventions.

5.6.4.2 Bigrams

The Chi-squared test is computed to capture two-word phrases (i.e. collocations) that are not likely occurring together by random chance:

$$Chisq_{x,y} = N * \emptyset^2$$

Equation 5.1 Chi-square statistic

Where ϕ is essentially a normalized sum of squared deviations between the expected and observed frequencies, *N* is the number of tokens in the corpus, *x* and *y* are two words that are being tested. The theoretical frequencies are derived from the base probabilities of every term appearing in the text. Whereas the observed values come from the frequencies of the corresponding bigrams. Nltk's module of bigram association measure has been used to compute this test. This method not only captures intuitive phrases like '*thank you*' and '*I am*', but also the multifaceted composition of Twitter which describe certain event of phenomena, such as "*#eureferendum*", "*#voteleave*", and "*#strongerin*".

5.6.5 POS Tagging

For STSS measures, POS tagging is necessary to identify the syntactical similarity based on the grammatical structure of a tweet. In this methodology, NLTK's simple statistical unigram tagging algorithm is used, which assigns the tag that is most likely for a given token. For example, it will assign the tag *JJ* to any occurrence of the word "beautiful", based on the concept that "beautiful" is used as an adjective (e.g. a *beautiful* city) more often than it is used as other parts of speech.

5.6.6 Trimming User Handles

A rule-based heuristic is implemented for stripping the user handles at the beginning of a retweets, such as *RT @ronnyhansen1*. If the tweet contains a ':' and the amount of text after this punctuation is larger than the text before it, then anything before is discarded. For example,

> *RT @ronnyhansen1: @CORCAS_AUTONOMY: yes, #Saharawi are sovereign in #WesternSahara, not Morocco. Why not hold agreed referendum to find out…*

Becomes,

> *yes, #Saharawi are sovereign in #WesternSahara, not Morocco. Why not hold agreed referendum to find out…*

Which demonstrates semantically richer and more condense content. The algorithm implemented for trimming a tweet is demonstrated in Table 5.2.

Table 5.2 Tweet trimming procedure pseudocode

Algorithm 1 Trimming user handles

```
1 function Trim(tweet):
  Input: tweet text
  Output: a tweet that does not contain only tweet –related
  text.
2   t ← tweet
3   if t contain ':' :
4     t_lst ← t.split(':')
5       if t_lst[0] ≤ t_lst[1]:
6         t ← t_lst[1]
7   return t
8 end function
```

5.6.7 Punctuations and Special Symbols

Unlike common approaches of removing all punctuations and special symbols, this research develops a heuristic-based approach for dealing with punctuations and special symbols to refine the tweet content. Common Twitter conventions and punctuations are most likely to be omitted in methods of semantic inferences in social data (Singh and Kumari, 2016). However, in this research, the author hypothesises that these symbolic structures are of no less importance than words in social contexts. That is, they carry information nuggets that cannot be discarded. This is particularly true in Twitter microblog as users do not often follow a grammatical structure in tweets due to the informal nature of the social network. For example, consider the two tweets,

T_1, '*going to Rome this weekend!*'

T_2, '*going to Rome this weekend?*'

Although both tweets are constructed from the same words, punctuating them

differently changes the complete function of the tweet. The exclamation mark in T_1 expresses the user's excitement, whereas T_2 is an interrogative sentence expressing the user's uncertainty. Another common use in informal contexts such as Twitter (albeit out of scope) is the sarcastic case. To further elaborate the role of expressive punctuations (i.e. interrogation and exclamation marks) in Twitter, the tweet '*Do I really need to mention this again!*' has a latent rhetorical interrogation mark that indicates intended sarcasm.

Furthermore, special symbols (e.g. $ and %) are prevalent in tweets and carry syntactic information that cannot be ignored. These syntactical feature are used in formulating the representative syntactical feature vector from which TREASURE computes the syntactic similarity (further elaborated in Chapter 6). Therefore, the aforementioned special characters are retained and the rest of the punctuations, such as commas and full stops are removed.

5.6.8 Stemming and Lemmatization

Stemming and lemmatization are special forms of normalization. They aim to reduce inflectional morphology of words through identifying a canonical representative as a common base form for a set of related word forms. The choice of employing either technique is a trade-off between effectiveness and efficiency. Stemming employs a crude heuristic operating on a single word without accounting for the context, and therefore does not take into consideration part of speech tags to discriminate between them. Although stemmers are faster and easier to implement, this research uses lemmatization as it operates based on a vocabulary and morphological analysis of a word form to link it back to its lemma. For example, the word "worst" has "bad" as its lemma. As this link requires a dictionary lookup, it is missed by stemming. WordNet (Miller et al., 1990) is used in this research for the lemmatization algorithm as a lookup for word roots in order to reduce the feature space by unifying multiple word forms.

5.6.9 Twitter Conventions

While highlighting the role of expressive characters in Section 5.6.7 and their importance in delivering the overall meaning of a tweet, common Twitter conventions (e.g. *#hashtags* and *@mentions*) are taken into account as well. Hash-tagging timely events and mentioning users over the network are frequently apparent in Twitter and

almost every tweet contains at least one of them. The lexical parser module in the pre-processing component breaks down the tokens in a tweet and produces a list of the hashtags and mentions. Hashtags are common conventions generated by users to create and follow a thread of discussion by prefixing a word with the '#' character (Wang et al., 2017). Many studies perform topic identification based on classifying hashtags as these greatly contribute to the meaning of a tweet (Antenucci et al., 2011). Therefore, these are important pieces of information that should be represented in the feature set for an STSS measure. However, hashtags are not usually intuitive to interpret by a computer program.

A major problem with hashtags is that they are often composed of joined words. While some hashtags are composed of joined words starting with capital letters, such as "*#JoyDivision*", most joined words are lowered cased. In the latter case, the challenge lies in determining where the boundaries are between the joined words. For example, given a hashtag such as *#talksofthemonth* return "talks of the month" and not "talk soft he month". Table 5.3 shows samples of joined hashtags and their possible interpretations. Due to this challenge, most approaches to STSS measures in Twitter either ignore hashtags (Satyapanich et al., 2015) or simply remove the hash character and treat the rest as a single word (Fócil-Arias et al.). Consequently, a portion of the similarity between the two texts will be missing.

Table 5.3: Examples of preferred and ambiguous hashtag tokenization

Hashtag	**Target tokenization**	**Ambiguous tokenization**
#longisland	long island	Long is land
#isreal	isreal	is real
#facebook	Facebook	face book
#healthexchange	health exchange	heal the x change

In this work, we propose a heuristic-based pre-processing methodology for handling the problem of hashtag compound segmentation. Let *h* be a hashtag of compound words, our algorithm works as follows:

1. If the regular expression based conditional statement *S < h is composed of upper and lower case characters>* is *true*, the boundaries upon which the words in *h* are split, are the change in character case.
2. If *S* if *false*, a dynamic programming is performed using the Viterbi algorithm (Forney, 1973). As this algorithm uses language model of words distributions

to calculate the most probable sequence, an English corpus[2] is used from which word frequencies are computed.

The hashtag segmentation component takes the compound hashtag and the words distribution model as input, and converts the hashtag to a vector of words composing them.

Another common Twitter convention, which is related to users more than the topic of a tweet, is a "*mention"*. Users use the @ sign to mention other users as a way of referring or having discussions with them in a public realm (e.g. *@RubyAS came yesterday*). While these common Twitter conventions may be useful in modelling user behaviour or community detection applications, they do not contribute to the meaning of the text. Therefore, a record of the existence of a mention in a tweet is identified as a flag in the syntactic feature vector however, these are removed from a tweet when deriving the semantic vector (semantic and syntactic feature vectors are detailed in Chapter 6).

5.6.10 Function Words and Contractions

It is a common practice to remove function words (also known as *stop words*) from a short text in applications of STSS as well as traditional information retrieval systems (Yoon et al., 2013, Shah, 2008, Satyapanich et al., 2015). However, while function words are not very useful in tasks computing documents similarity, function words carry structural information and therefore cannot be ignored in a very short text such as tweets (Li et al., 2006). Nevertheless, although function words are retained in the Twitter-based datasets used in this research, they are considered to carry less information content and therefore contribute less to the overall meaning compared to other infrequently occurring words.

Furthermore, converting contractions to their expanded format would reduce word sense ambiguities by means of structure. It involves converting words with apostrophes to its standard lexicon (e.g. *should've* becomes *should have*). This is particularly important to avoid confusion between contractions and possessiveness (e.g. *it's* versus *its*).

5.6.11 Digits

Unlike most pre-processing strategies followed by researchers that remove digits, as

[2] http://norvig.com/big.txt

with function words, this research keeps digits because they are considered to carry information in a very short text such as a tweet. Dealing with a digit as a string or as an integer is a technical aspect related to the implementation of an STSS measure. TREASURE considers digits and decimals as a syntactic feature that contribute to the syntactic similarity between a pair of tweets (further elaborated in Chapter 6).

Figure 5.3 shows a flowchart of the heuristic-driven pre-processing methodology followed in this study.

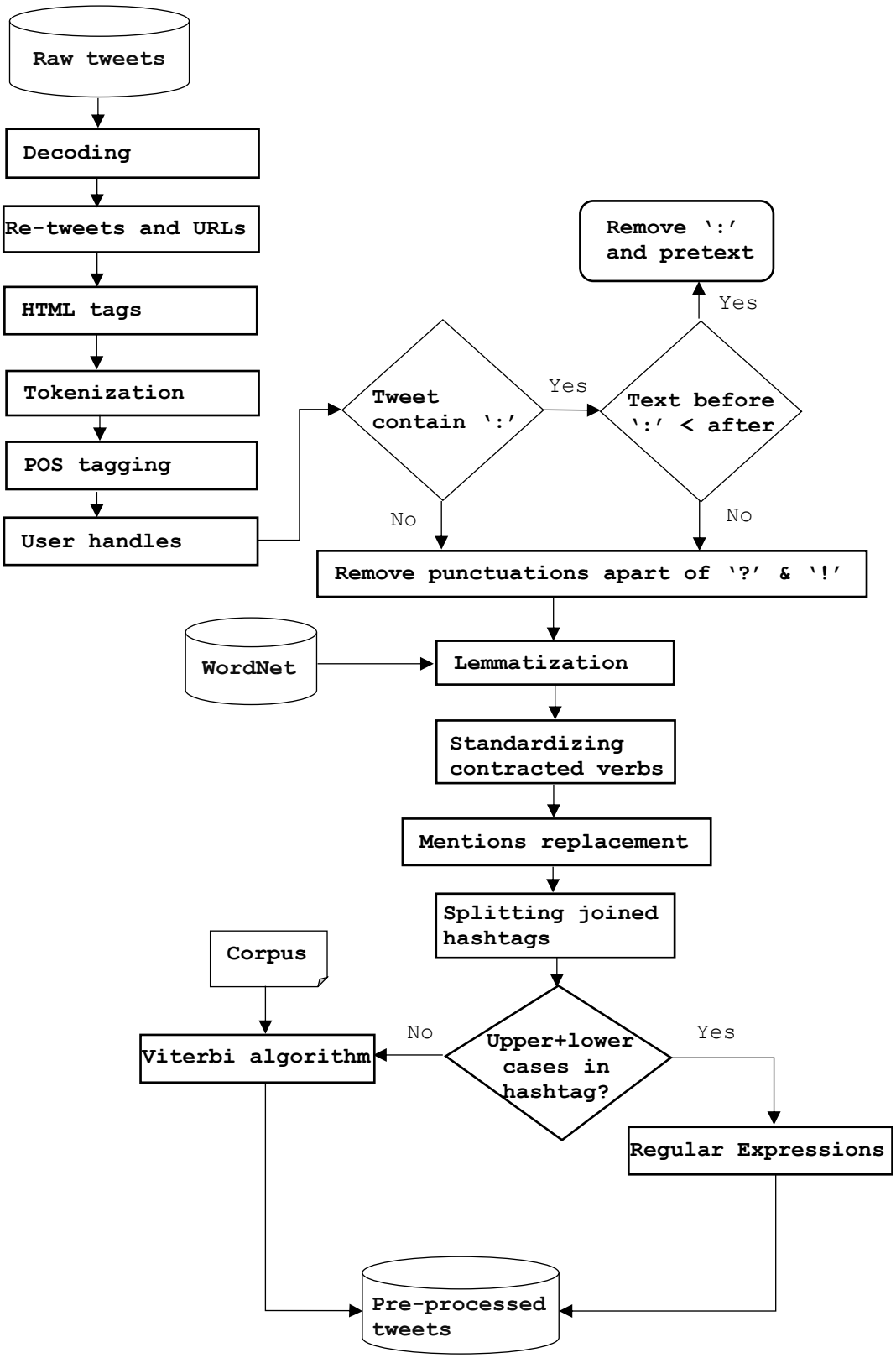


Figure 5.3 The heuristic-driven pre-processing flowchart

5.7 Experiment to Evaluate the Pre-Processing Methodology

This section describes the experiment conducted to evaluate the effectiveness of the pre-processing methodology on the performance of a textual similarity measure. This experiment aims to provide evidence that the new pre-processing heuristics described in Section 5.6 are more effective (in the context of STSS measurement) than the pre-processing baseline, which is a set of stages used in most NLP applications. This evidence is derived through examining the results of correlation analysis and error rates achieved by keyword-based cosine similarity STSS using two different pre-processing methodologies. These methodologies are the proposed heuristics versus the baseline pre-processing method (C-Method) (described in Section 5.7.3) with reference to the STS.tweet_news trial gold standard dataset which is further elaborated in the following section.

5.7.1 SemEval-2014 Similarity Benchmark

SemEval is a collection of online computational semantic analysis shared tasks intended to explore the natural meaning in different languages. Part of the SemEval-2014 shared task published a trial gold standard STS.tweet_news dataset of 750 annotated pairs of tweets and news headlines (Guo et al., 2013). This benchmark dataset adopted a 6-point Likert scale to measure the degree of similarity score between pairs. People undertaking the experiment were requested to assign each pair a similarity score as defined by Agirre et al. (2012):

(0) On different topics.

(1) Not equivalent, but are on the same topic.

(2) Not equivalent, but share some details.

(3) Roughly equivalent, but some important information differs/missing.

(4) Mostly equivalent, but some unimportant details differ.

(5) Completely equivalent, as they mean the same thing.

The similarity scores labels on the STS.tweet_news are the average of five scores assembled using Amazon Mechanical Turk (AMT) (Buhrmester et al., 2011) for each pair. The STS.tweet_news dataset is a subset of the Linking-Tweets-to-News dataset (Guo et al., 2013), which is composed of 34,888 tweets and 12,704 news articles headlines. A random sample pair and its assigned similarity label from the STS.tweet_news benchmark dataset is shown in Table 5.4.

Table 5.4 A sample pair from the STS.tweet_news benchmark

Pair		**Similarity label**
Tweet	**News headline**	
I need a 'stop day' in my life. #CNN	The importance of a 'stop day'	2.8

The tweets are the comments on the news articles and the news short text sentences are the titles of the news articles.

5.7.2 Similarity Measure for Evaluating the Pre-processing Heuristic

To assess the effect of the proposed pre-processing methodology on an STSS measure, keyword-based cosine similarity is computed on a TF-IDF weighted corpus to scale down the value of common occurring words and scale up the value of rare words. The Scikit-learn Python library was used to perform the vectorization and weighting.

Given two tweets, T_1 and T_2, a joint feature vector V is derived, which is composed of the unique unigrams in T_1 and T_2. T_1 and T_2 are then represented by v_1 and v_2 respectively, which are frequency vectors calculated based on V. The cosine similarity is then computed between v_1 and v_2.

5.7.3 Baseline and Evaluation Criteria

The baseline method for performing pre-processing is the classic method (C-Method) using *n*-grams, which has been used in most STSS approaches (Guo et al., 2013, Hajjem and Latiri, 2016). This method applies six classical pre-processing steps, including removing URLs, removing stop words, removing numbers, standardizing words, and removing punctuations. The evaluation metrics (discussed later in this section) are also computed for the raw data.

A good STSS measure is one with high correlations and low error rates (Alnajran et al., 2018a). Therefore, the Pearson correlation coefficient and error rates were selected to evaluate the overall performance of the STSS measure (described in Section 5.7.4) as follows:

- Correlations are used to detect whether a linear relationship can be modelled between the actual (human) and estimated (STSS measure) readings. The effect of the pre-processing techniques are assessed by a comparison of the correlations between the human judgments and the estimations recorded by the measure for the baseline and the proposed methodology.

Error rates are negatively oriented scores that are used in predictive modelling. In addition to correlations, the mean absolute error (MAE) and the mean squared error

(MSE) were calculated. MAE is considered robust to outliers as it does not make use of square, whereas MSE emphasizes the extremes. This means that the square of a very small number (smaller than 1) is even smaller, and the square of a big number is even bigger.

5.7.4 Experiment Results

In this section, the researcher reports the results obtained on raw tweets before and after the application of the new developed pre-processing heuristic and the baseline individually. The baseline (C-Method) is the method that applies the classical pre-processing steps as described in Section 5.7.3 and the proposed methodology applies the rules described in Section 5.6. Thus, the cosine similarity measure was computed on the raw data, the baseline, and the developed pre-processing heuristic using the STS.tweet_news similarity benchmark for evaluation. The impact is analysed and assessed through computing the evaluation criteria outlined in Section 5.7.3.

Table 5.5 demonstrates the performance of the cosine similarity measure depending on the pre-processing method applied. Regarding the pre-processing representations, the measure's behaviour is not uniform. It is apparent that the proposed methodology in this research achieves the highest correlation coefficient, significant at 0.01.

Table 5.5 Results of evaluating the pre-processing methodologies

Pre-processing Method	***r***	**MAE**	**MSE**
Raw Data	0.7017	1.1296	2.0281
C-Method	0.7264	1.1288	1.94
Research Method	**0.7585**	**1.0759**	**1.7425**

Figure 5.4 provides a graph visualisation of the evaluation results for the pre-processing methodologies. The evaluation results indicate that the proposed pre-processing methodology outperforms the baseline in terms of correlation and error rates. For the STS.tweet_news similarity-labelled dataset, the research methodology for pre-processing tweets achieves 3% enhancement over the C-Method and 6% over the raw dataset. The variance between the correlations is expected to increase for different twitter-based domains. This is attributed to the case that STS.tweet_news dataset is not considered as noisy as typical twitter data (e.g. EU Referendum dataset). With regards to error rates, the proposed methodology generates the least variance compared to the C-Method and the raw dataset. By observing the readings of MAE and MSE, it can be concluded that the dataset has many outliers. This is because MSE is 0.7 higher than MAE, which is more robust to outliers.

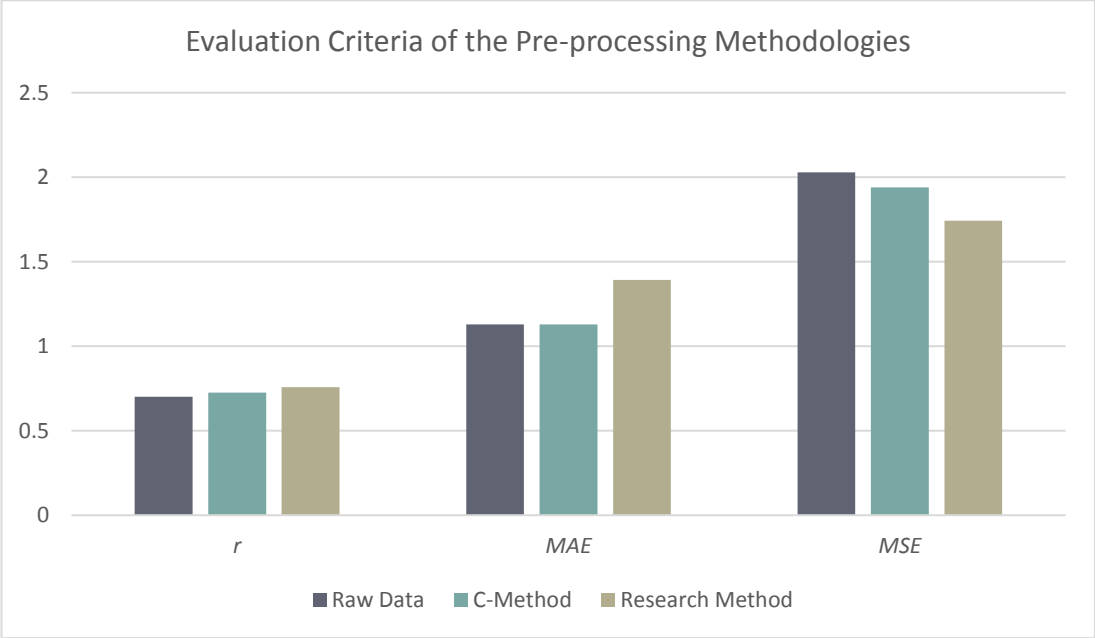


Figure 5.4 Results of the pre-processing methodologies in terms of correlation (r), MAE, and MSE

While the overall evaluation results may indicate low accuracy of the keyword-based cosine similarity measure, the purpose of this experiment is not to evaluate the performance of the similarity measure, rather the effect of the proposed pre-processing methodology in enhancing the results of the similarity measure compared to common practices of pre-processing.

5.8 Feature Extraction

The feature extraction process is carried out after a pre-processed dataset is derived from the raw dataset of tweets using the heuristic described in Section 5.6. This section describes the semantic and syntactic feature set extracted from the pre-processed tweets.

As discussed earlier, tweets are associated with multiple features that represent their syntactic and semantic status. Some of these features are straightforward while other features are derived from joint features or calculated from the corpus of the tweets. In this research, the utilized set of features that contribute to the core body of the proposed TREASURE STSS measure (described in Chapter 6 and evaluated in Chapter 7) to be used in the semantic-based cluster analysis (SBCA) algorithm (described in Chapter 8 and evaluated in Chapter 9) are categorized in the subsequent sections. The process of generating a tweet's corresponding semantic and syntactic feature vectors are further elaborated in Chapter 6.

5.8.1 Syntactic Feature Set

The syntactic features that are extracted and derived from tweets, which will be required for manipulation by the syntactic component of the novel STSS measure, namely TREASURE (detailed in the next chapter), are as follows:

- *POS tags* –refer to tokenizing text segments based on their morphological role in the corresponding tweet: function word, noun, verb, adjective, adverb, and digit (Section 5.6.5).
- *Twitter conventions* –refer to the common user conventions in a tweet such as hashtags and mentions (Section 5.6.9).
- *Punctuation marks* –refer to exclamation and interrogation marks (Section 5.6.7).
- *Special symbols* –refer to special symbols that are prevalent in microblogs such as currency and percentage characters, which may indicate the certain theme of a tweet (Section 5.6.7).

The general categories of syntactical features along with their corresponding subcategories are discussed in detail in Chapter 6.

5.8.2 Semantic Feature Set

The semantic features extracted from the pre-processed tweets, which will be utilised by the semantic component of TREASURE (detailed in the next chapter) are the *n*-grams from which a tweet post is composed (Section 5.6.4). The *n*-grams may be words, phrases, or hashtags that carry different weights according to their information content derived from a large corpus of collected tweets. The weighting scheme employed to determine the significance of a token is detailed in Chapter 6.

5.9 Chapter Summary

In this chapter, the EU Referendum and SemEval-2014 STS.tweet_news datasets utilised in this research are described. The EU Referendum dataset is constructed through streaming tweets on the political domain and the STS.tweet_news dataset consists of tweet-news pairs that are labelled with human similarity judgements. The consequent processes of data collection, storage, and a new heuristic-based pre-processing methodology for enhancing the performance of STSS measures are described. The pre-processing methodology is composed of several consecutive rules that were configured from empirical experiments based on the trial and error problem

solving method (Starch, 1910). An experiment was conducted using the cosine coefficient as the similarity measurement for verifying the effectiveness of the new pre-processing methodology against a baseline method on a similarity-annotated dataset of tweet pairs. Experimental results provides evidence that the new pre-processing methodology outperforms the common practice of pre-processing in terms of correlation and error rates. Furthermore, the results demonstrate the importance of pre-processing and data quality in leveraging the performance of STSS in microblogs, such as Twitter. The set of semantic and syntactic features considered in a tweet are also listed. The subsequent process of deriving the corresponding feature vectors that represent a tweet post is described in Chapter 6.

The main contribution of this Chapter is:

- Design of a heuristic-driven methodology for pre-processing microblogging posts, particularly tweets, which is intended for STSS measures. Experimental results provide evidence that the proposed pre-processing methodology enhances the performance of a similarity computation measure.

Chapter 6 – TREASURE –A Microblogging STSS Measure Development Methodology

6.1 Overview

This chapter describes the statistical semantic approach and components developed for implementing a Short Text Semantic Similarity STSS measure for microblogging posts, particularly tweets, known as TREASURE (**T**weet simila**R**ity m**EASURE**). TREASURE is a novel STSS approach, which measures the semantic similarity between pairs of tweets by extracting semantic *and* syntactic features. The hybrid feature set utilised by TREASURE is implemented to generate a meaningful representation for each tweet.

Although tweet similarity is essential for a variety of applications, as described earlier in Chapter 2, Section 2.3, there is not much research on computing semantic similarity for tweets based on word embedding models; rather, existing research towards tweet similarity computation is either based on shared keywords or formal lexical resources (i.e. thesaurus). Moreover, the use of existing measures to computing tweet similarity has three major drawbacks. First, sentence similarity measures configured on WordNet will perform poorly on a Twitter-based dataset as most terms are not present in the ontological hierarchy. Second, corpus-based semantic measures that are trained and designed for an application domain cannot be adapted easily to other domains. Third, some approaches require intensive involvement from humans to manually preprocess the noisy text in tweets, which is an immensely arduous and tedious task. This lack of adaptability corresponds to the informal nature of the communication platform and common user generated conventions used in most OSN. To address these drawbacks, this research aims to develop a *hybrid* approach to similarity measurement of microblogging posts that: 1) Undertake a new pre-processing methodology that aims to model a tweet by extracting semantic and syntactic features. 2) Implements a new short-text semantic similarity (STSS) measure, namely TREASURE, for tweets.

This chapter describes the methodology for developing TREASURE, which includes a design of the main architecture including the semantic and syntactic components, their corresponding sub modules, the word embedding models, and the algorithm for the similarity computation process.

In summary, based on the critical review of previous studies and state-of-the-art approaches and their associated weaknesses in handling microblogs computational linguistic challenges provided in Chapter 2, TREASURE features the following characteristics:

- *Symmetric* –the similarity degree between two candidate tweets, T_1 and T_2, should be the same as that between T_2 and T_1.
- *Fully unsupervised* –does not require any kind of user manual intervention.
- *Hybrid feature set* –extracts and utilizes both semantic and syntactic features present in a tweet pair.
- *Dynamic pipeline* –creates a dynamic joint vector representing the tweet pair rather than a static high dimensional bag-of-words (BOW).
- *Adaptable* –readily replicated across the range of potential application domains in the context of microblogging OSN.

In this chapter, Section 6.2 provides an overview of the new STSS architectural design for measuring tweet similarity. Section 6.3 describes the development methodology followed in implementing TREASURE STSS measure. Section 6.4 provides a detailed description of the semantic components that handles the words semantic co-occurrence relationships computations. The different modules incorporated in this component including the training process of the artificial neural network and the weighting schema are also presented and discussed. Section 6.5 describes the syntactic component that handles the computation of the similarities between a candidate pair based on structural and contextual analysis. The different syntactic modules and their contributions to the similarity are also detailed in this section. Section 6.6 provides a demonstration of the semantic similarity computations, whereas the syntactic similarity computations are presented in Section 6.7. The combined weighted contributions of these two similarities to generate the overall similarity measure and threshold considerations are discussed in Section 6.8. An illustrative example of deriving the semantic and syntactic similarities for a selected tweet pair and demonstrating the process of computing the overall similarity score is provided in Section 6.9. Finally, section 6.10 summarizes the chapter, draws some conclusions, and highlights key contributions.

6.2 TREASURE Architecture Overview

This section illustrates the design and development of the main architecture components in the TREASURE STSS measure. TREASURE features a hybrid approach that consists of two components. The first consists of the semantic modules, which handle semantic word analogy computations and weighting schema. The second consists of the syntactic modules, which take into consideration the morphological structure of words posted in microblogs, particularly Twitter.

Unlike semantic similarity methods, which only take into consideration the similarity derived through topological or statistical semantic computations, TREASURE not only considers *semantic* interpretation, but also accounts for the contribution of the *morphological structure* of terms occurring in a tweet. Syntactic features are particularly important in social contexts such as Twitter because, although tweets are unstructured texts, users in Twitter often express their meaning using common conventions and certain punctuations due to the restriction over character limit. Therefore, ignoring such features leads to missing nuggets of information in the representation of the feature vector for each transformed tweet.

6.3 Methodology of Implementing TREASURE STSS

TREASURE was designed, developed, and evaluated following the general processes in the classical Waterfall software development lifecycle (SDLC) model over newer models, such as Agile (Constantine and Lockwood, 1999). This is attributed to the progress being more easily measured in the Waterfall model, as the full scope of the research project is known in advance. The main stages are shown in Figure 6.1.

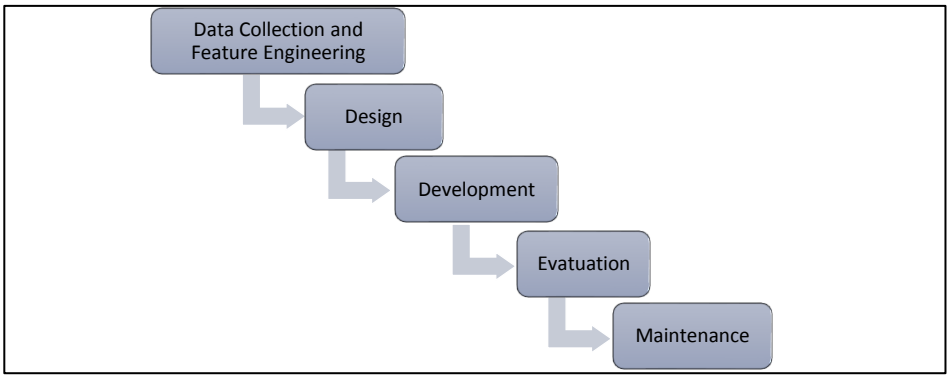


Figure 6.1: TREASURE development phases according to the Waterfall SDLC model

The TREASURE architecture was designed by integrating the pre-processing module (described in Chapter 5) and the semantic and syntactic components. The proposed TREASURE architecture is shown in Figure 6.2.

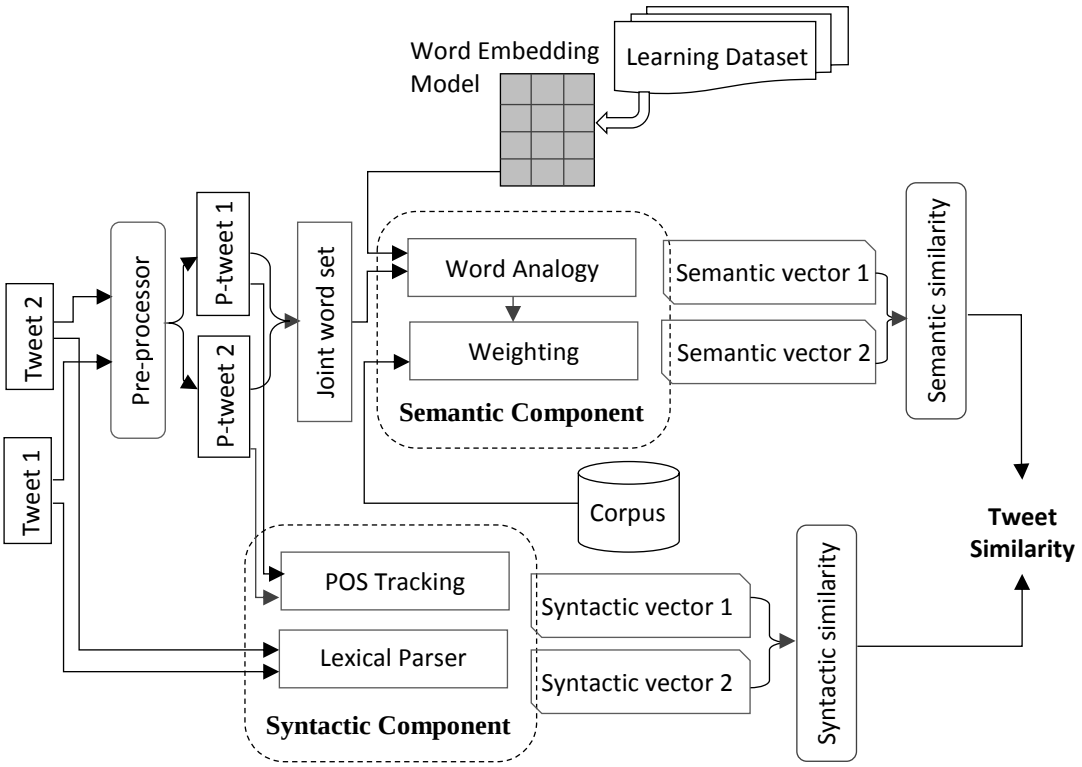


Figure 6.2 The TREASURE STSS architectural design

A tweet is composed of maximum 280 characters considered to be a sequence of words hashtags, mentions, and URLs. The combination of words and hashtags in a tweet, along with their syntactical structure, make a tweet convey a specific meaning. Figure 6.2 presents the process undertaken for tweet similarity computation between a tweet pair being assessed for similarity. After going through pre-processing stages, the proposed method generates a dynamic joint representation of the pair of tweets consisting of the unique words within them. For each tweet, a semantic and a syntactic vector is constructed. The semantic vector is derived using a pre-trained word embedding model and the value of each term is calculated by applying a weighting scheme using a corpus. The syntactic vector is formed in the syntactic component, which extracts features that describe the syntactical structure of a tweet. The semantic and syntactic similarities are computed by calculating the distance between their corresponding vectors. Finally, the overall similarity between a pair of tweets is derived by combining the output of the semantic similarity and syntactic similarity.

The subsequent sections present a detailed description of each component in the proposed tweet similarity algorithm.

TREASURE's main elements consist of the pre-processing steps (discussed in Chapter 5) to generate semantic-rich tweets, the semantic components, and the syntactic component. The subsequent sections describe the implementation for each component in detail.

6.4 Component 1: Implementing the Semantic Decomposition Modules

This component consists of the following modules:

1. The *word analogy* module, which derives words semantic co-occurrence relationships based on word embedding models that contain dense word vector representations (Section 6.4.1).
2. The *word embedding model* generated through unsupervised learning using an artificial neural network to learn word co-occurrences from a large corpus of microblogging posts (Section 6.4.2).
3. The *weighting schema* that determines a term's contribution to the meaning based on its significance according to this scheme (Section 6.4.3).

6.4.1 Word Analogy

Word embedding projects in computational linguistics encode meanings of words to low dimensional vector spaces. Unlike traditional distributional semantic vector space models such as latent semantic analysis (LSA) and latent Dirichlet allocation (LDA), these recent techniques generate dense, continuous valued vectors, called *embeddings.* Word embedding approaches have become the state-of-the-art performances in many intrinsic NLP tasks such as cluster analysis (Dai et al., 2017) and semantic textual similarity (De Boom et al., 2015a) due to their potential in capturing the semantic relations among words. The process of learning embeddings include neural network-based predictive methods, such as Word2Vec (Mikolov et al., 2013a, Bojanowski et al., 2016) and count-based matrix factorization methods, such as GloVe (Pennington et al., 2014). The word analogy module implements a shallow word embedding model, Word2Vec, which is used as the source algorithm for learning dense word vectors. The artificial neural networks used to generate the pre-trained models is a skip-gram architecture as shown in Figure 6.3.

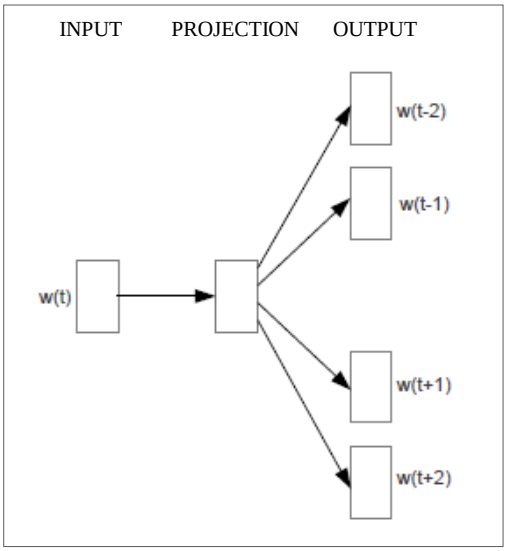


Figure 6.3 Skip-gram model architecture (Mikolov et al., 2013a)

The skip-gram model predicts surrounding words $c_1, c_2, \ldots, c_n$ given the current word w (n is the size of the context window), such as $P(c1|w)$, $P(c2|w)$, and etc. The resulting trained embedding model consists of a word embedding vector denoted by $\check{v}$, for each word w in the model.

Given two words w_1 and w_2, the word analogy module computes the semantic similarity $S_{sem}(w_1, w_2)$. This is obtained by calculating the cosine coefficient between the two corresponding word embedding vectors $\check{v}_1$ and $\check{v}_2$ for w_1 and w_2 in the semantic embedding space. For example, the cosine similarity between $\check{v}_{Obama}$ and $\check{v}_{president}$ in the Google News pre-trained Word2Vec model is 0.31.

6.4.2 Word Embedding Models

The observations from the literature reviewed in Chapter 2 around word embedding in the context of Twitter-based semantic textual analysis revealed potential capabilities of such techniques for microblogging posts analysis. Furthermore, tweets are challenging for classical vector representations and topic modelling methods due to the inadequate information and lack of context for manipulation by a computational method (Alnajran et al., 2018a). Therefore, TREASURE performs semantic computations by obtaining knowledge on word similarities from word embedding models. In this section, the word embedding models used and trained for computing words semantic relationships are described in detail.

6.4.2.1 Google News Pre-trained Model

Mikolov et al. (2013b) trained a Skip-gram Word2vec model on a large dataset of general news articles. The model consists of three million vocabulary words. The generated word embeddings are used to calculate word similarities in the developed sematic similarity method. This model is used for evaluation on the labelled STS.tweet_news dataset. The model's corpus metadata and training hyper-parameters are shown in Table 6.1.

Table 6.1 Corpus metadata and model hyper-parameters for Google News pre-trained model

Metadata and hyper-parameters	**Google News Embedding Model**
Words in the corpus	100 billion words
Unique tokens in the trained embedding model	V = 3M
Training algorithm	Skip-gram/negative sub-sampling
Vector dimension	d = 300
Negative samples	k = 5
Minimum frequency threshold	min_count = 5
Learning context window	w' = 5
Training time	1 day
Trained model size	3G

The Google News Pre-trained Model is implemented in the word analogy module for measuring the similarities between pairs in STS.tweet_news dataset based on the following considerations:

- Both corpora are on the general news domain. The Google News Pre-trained model learned distributed representations of words from traditional news Web documents and STS.tweet_news pairs are composed of news tweets as well as news headlines.
- Although both corpora share similar domains, out-of-vocabulary (OOV) words are prevalent in STS.tweet_news pairs, which are not found in the Google News pre-trained model, being trained on general text corpora. Moreover, users tend to share news in microblogs differently in a more informal manner. However, the lack of news tweets corpora that spans the period where the STS.tweet_news pairs were assembled in order to be used for training an artificial neural network to generate word vectors has led to the choice of the Google News Pre-trained model.

The Google News pre-trained model represents word vectors according to their co-occurrences in a formal and structured context (e.g. Web documents) compared to their colloquial use in social contexts, such as in tweets. Tweets share unique lexical

and structural features that are different from general texts found in traditional documents. The user generated content found in microblogs, particularly Twitter, is usually a fertile environment for noise and common user conventions and emoticons (detailed in Chapter 2). The informal nature of this social medium and the character limit restriction has lead people to cut out conjunctions, pronouns, and substitute expressive terms with emoji in order to ultimately use the allowed range of characters in delivering the intended meaning. This social norm of words employment will consequently generate different word representations.

Therefore, neural embedding models trained on traditional text documents often fall short for capturing the semantic relationships between words present in the social context (i.e. Twitter-based NLP applications) (Wang et al., 2017). The subsequent section describes the process of training an artificial neural network on the political EU_Referendum dataset (the collection of this dataset is described in Chapter 5) to learn distributed word representations and generate a Twitter-based word embedding model.

6.4.2.2 The Political Word Embedding Model

Due to the observations discussed in section 6.4.2.1, the Google News pre-trained model is not considered a good candidate to be used by the word analogy module to capture semantic relationships for the EU_Referendum political tweets. The special features of the EU_Referendum dataset require an embedding model that analyses and models the behavior of words used in this social context.

This section describes the processes undertaken in producing a pre-trained word embedding model learned from a corpus of political tweets. Figure 6.4 shows a layered representation of the model's training process. The processes undertaken in each layer and the model's training configurations are further described in the subsequent sections.

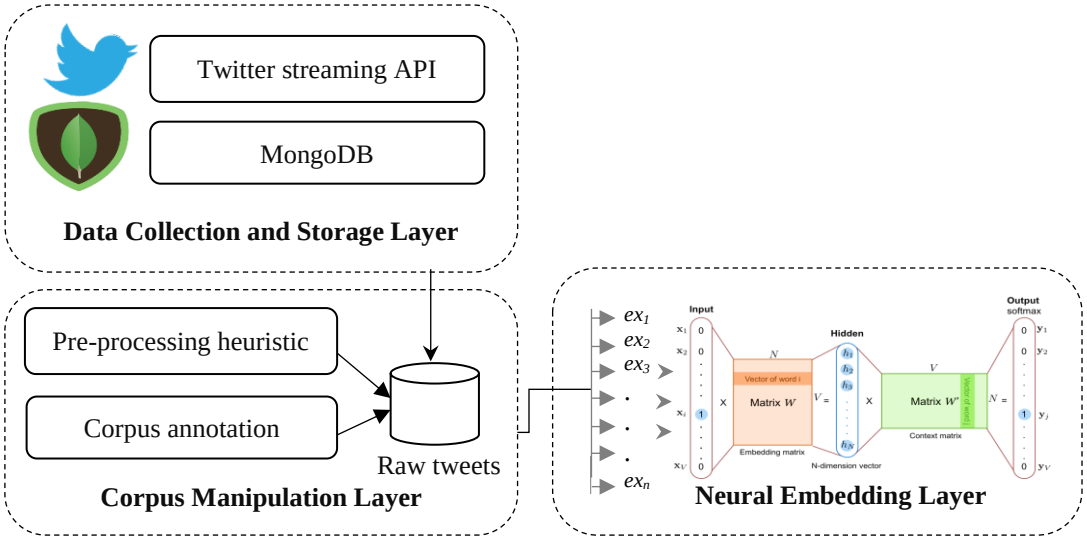


Figure 6.4 Layers of the phases involved in training the EU_Referendum word embedding model

1) *Data Collection and Storage Layer:* this layer involves setting up the Twitter Streaming API and its configuration on the political domain for data collection. The streamed tweets are stored in MongoDB NoSQL database on the flow. That is, in a real-time mode rather than storing them to an external file and transferring them to Mongo DB in batches afterwards (Chapter 5, Sections 5.2, 5.3, and 5.4).
2) *Corpus Manipulation Layer:* the input to this layer is the raw tweets obtained from the previous layer. Corpus manipulation includes pre-processing steps (Chapter 5, Section 5.6) including *n*-gram identification and corpus annotation. Theoretically, training a word embedding model assuming all words in the corpus are isolated from each other is memory intensive (Mikolov et al., 2013b). Additionally, many phrases have a single meaning that is not simply a composition of the meaning of its individual words, such as 'New Jersey'. Therefore, the Chi-squared test is used to identify phrases in the corpus based on frequently occurring bigrams that are commonly embedded in discourse, such as '*vote leave*' and '*stronger in*' (described in Section 6.4.3). After detecting common bigrams in the corpus, the next process involves annotating the corpus with the identified phrases in the previous step. The words that make a phrase are joined using an *underscore* character. For example, '*…visited New York and San Francisco…*' would become *'…visited new_york and san_francisco…*' The resulting corpus then consists of unigrams and explicitly tagged bigrams.

3) *Neural Embedding Layer:* in this layer, the actual training of the word embedding model is performed on the pre-processed and annotated corpus. The goal is to learn the weights of the neural networks hidden layer, which are actually the distributed word representations.

This section describes the methodology used in building and training the word embedding model learned from the political tweets dataset on the EU_Referendum described in Chapter 5.

A. Vocabulary Trimming

A vocabulary of 12.3 million words and phrases are included in the corpus. However, this vocabulary may contain rarely occurring words that lack enough context. Therefore, the minimum word frequency threshold is set to *min_count* = 3. Words and phrases that do not satisfy the *min_count* are discarded due to two reasons: 1) the neural model does not have adequate training examples to learn meaningful embedding vectors for those words. 2) When performing corpus statistics, words occurring less than 3 times in the entire corpus are often typos (Li et al., 2017). The value of the *min_count* threshold has been determined empirically. The application of the minimum frequency threshold has generated a vocabulary V = 86K unique words and phrases in the training embedding model.

B. Model Architecture and Hyper-parameter Configuration

In this research, a Word2Vec Skip-gram artificial neural network model with negative sub-sampling is used (Mikolov et al., 2013b). The use of the Skip-gram model and sub-sampling frequently occurring words decreases the number of training examples, and consequently, reduces the computational burden of the training process. Word2Vec is a back propagation neural network composed of one hidden layer that learns by back-propagating the error to the hidden layer and thus update the input vectors of words. The learning process is unsupervised, in which the goal is to learn the weights between the input layer and the hidden layer, which are actually the embedding vector representations of words. This is similar to the unsupervised feature learning in training an auto-encoder. The architecture of the implemented neural network model is shown in Figure 6.3, Section 6.4.1.

1) Input layer: in this layer, the training examples (i.e. word pairs) are fed into the network. It has been found that a context window size of $w' = 5$ a good trade-off between efficiency and accuracy (Li et al., 2017). Empirical experiments were

conducted by the researcher on different window sizes $w' \in \{3, 4, 5, 6\}$ and have shown $w' = 5$ provides the best embedding vectors for tweets. The output probabilities predict the likelihood of a word occurring in the domain of the input word (i.e. the word's context window). For example, training the network on the word 'TTIP'[3], which is a typical acronym in the event of Brexit, the output probabilities are higher for words like 'trade' and 'union'. Considering the tweet, *T*, '*Brexit issue no organization afford to ignore*' as an example tweet in the annotated corpus described in Section 6.4.2.2, the training samples for *T* at $w' = 5$ are shown in Table 6.2.

Table 6.2 Illustrative example of the model's training input for w′ = 5

Sliding window ($w' = 5$)	**Target word**	**Context**
[brexit issue no organization afford to]	*brexit*	issue, no, organization, afford, to
[brexit issue no organization afford to ignore]	*issue*	brexit, no, organization, afford, to, ignore
[brexit issue no organization afford to ignore]	*no*	brexit, issue, organization, afford, to, ignore
[brexit issue no organization afford to ignore]	*organization*	brexit, issue, no, afford, to, ignore
[brexit issue no organization afford to ignore]	*afford*	brexit, issue, no, organization, to, ignore
[brexit issue no organization afford to ignore]	*to*	brexit, issue, no, organization, afford, ignore
[issue no organization afford to ignore]	*ignore*	issue, no, organization, afford, to

Subsampling is performed to eliminate very frequent words with marginal information content (such as *the*). The probability, *p*, of which a given word is kept in the vocabulary, is calculated as follows:

$$p(w_i) = \left(\sqrt{\frac{z(w_i)}{0.001}} + 1\right) \times \frac{0.001}{z(w_i)}$$

$$p(w_i) = \begin{cases} 1, & z(w_i) < 0.0026 \\ 0.5, & z(w_i) = 0.00746 \\ 0.033, & z(w_i) = 1.0 \end{cases}$$

Equation 6.1 Word probability in a vocabulary (Mikolov et al., 2013b)

Where $z(w_i)$ is the fraction of the total occurrence of the word w_i in the corpus. The sample value of 0.001 is the default sampling parameter (Mikolov et al., 2013b).

2) Hidden layer: in this layer, the dimensions of the embedding vectors is set to $d = 300$. That is, the configured model is learning word vectors with 300 features instead of the high dimensional vocabulary size. The hidden layer is thus represented

[3] Transatlantic Trade and Investment Partnership

by a weight matrix A (86K×300*d*), with 86K rows (1 per each record in the vocabulary) and 300 columns (1 per each hidden neuron).

3) Output layer: a vector for each word in the vocabulary is fed to the output layer. To optimize the computation of this layer, a 'negative sampling' is performed to avoid updating every neuron's weights for each vector in the vocabulary during training. Rather, only a small ratio of the weights are modified by each training vector. The researcher randomly selects five negative words, in which their weights are updated as well as the weights of the word in the training iteration. It has been reported by Mikolov et al. (2013b) that negative sampling value of five words works well for the EU_Referendum dataset size range. The selection of the negative samples is based on a unigram distribution approach, in which more frequent words are more likely to be sampled.

C. Model Complexity and Software Specifications

The model's training complexity is O(V), where V is the vocabulary size. Training the Word2Vec model on the political tweets dataset has taken 27 minutes running on Intel core i7 CPU and 16GB RAM. The statistical information on the learning corpus, trained embedding model, training configurations, and processor and memory specifications are shown in Table 6.3.

Table 6.3 Metadata and hyper-parameters for the EU_Referendum political tweets

Metadata and hyper-parameters	**Political Tweets Embedding Model**
Raw tweets	4 million
Words in the corpus	12.3 million
Unique tokens in the trained embedding model (min_count < 3 omitted)	V = 86K
Training algorithm	Skip-gram / negative sub-sampling
Negative samples	$k = 5$
Vector dimension	$d = 300$
Minimum frequency threshold	min_count = 3
Learning context window	$w' = 5$
Training time	17 minutes
Training complexity	O(V)
Trained model size	136MB

Word embedding models generate word vector representations based on performing iterations over the training corpus in order to learn words co-occurrences in a predefined context window size. Thus, even highly dissimilar words tend to share commonalities in their distributed word vector representations. This behavior should be taken into account in calculating $S_{sem}(w_1, w_2)$ in order to avoid introducing noise to

the semantic vector. Li et al. (2006) performed depth scaling of words in hierarchical semantic nets such that similarity of words at upper layers are scaled down and similarity of words at lower layers is scaled up. Similarly, scaling is performed on the similarity of words in TREASURE where the cosine coefficient of their corresponding vectors in the pre-trained embedding models is less than a certain threshold. A scaling parameter is defined as α, where $\alpha \in [0, 1]$. The optimal value of α is dependent on the word embedding model used and can be determined through the use of a benchmark word pairs dataset with human similarity ratings. Empirical experiments were conducted to determine the optimal threshold value for the pre-trained embedding models used in the word analogy module, which turned out to be $\alpha = 0.3$ for the proposed measure.

6.4.3 Weight Transformation

Unlike most text similarity algorithms, TREASURE retains all function words. However, as these words occur frequently, they contribute less to the meaning of a tweet than other words. Similarly, different words in a tweet contribute differently towards the meaning of a tweet. The significance of a word is determined according to the assumption that words occurring more frequently in a corpus contain less information than less frequently occurring words (Barry et al., 2007). Thus, the extent in which terms contribute to the overall meaning in a tweet is determined by how frequently they occur in a given corpus of tweets. The terms that occur more frequently tend to have less value compared to less frequent terms. However, common weighting techniques such as TF-IDF falls short in favoring discriminatory traits over nondiscriminatory ones in a tweet. This is due to the short and constrained nature of tweets, which creates an upper limit on the term frequency reducing its importance in the weighting scheme. Moreover, the massive size and creative vocabulary generated by Twitter users makes the representation of tweets in TF-IDF vectors sparse and less accurate. Therefore, the weight of a term (i.e. information it carries) is derived from calculating its probability in a corpus using a compound method as follows:

1. Chi-squared test is computed to capture two-word phrases (i.e. bigrams) that are not likely occurring together by random chance, which is computed according to Equation 5.1 in Chapter 5.
2. The probabilities of the bigrams and unigrams (i.e. words) in the corpus are

computed as the relative frequency as shown in Equation 6.2.

$$\hat{p}(g) = \frac{n+1}{N+1}$$

Equation 6.2 *n*-gram probability in a corpus

Where n is the frequency of the n-gram g in the corpus, and N is the total number of n-grams in the corpus (increased by 1 to avoid the case of undefined value). Weight of g in the corpus is defined in Equation 6.3.

$$W(g) = 1 - \frac{\log(n+1)}{\log(N+1)}$$

Equation 6.3 *n*-gram weight in a corpus

So $W \in [0, 1]$.

The semantic similarity $S_{sem}(w_1, w_2)$ between words w_1 and w_2 is therefore a function of word embedding e and word weight h as shown in Equation 6.4.

$$S_{sem}(w_1, w_2) = f(e, h)$$

Equation 6.4 Semantic similarity function

Where e is the cosine angle between embedding vectors $\check{v}_1$ and $\check{v}_2$ for words w_1 and w_2 in the pre-trained embedding model, h is the weight of w_1 and w_2 calculated following Equation 6.3. The author assumes that Equation 6.4 can be rewritten using two independent functions as in Equation 6.5.

$$S_{sem}(w_1, w_2) = f_1(e) \,.\, f_2(h)$$

Equation 6.5 Semantic similarity using independent functions

Where f_1 and f_2 are transfer functions of word embedding similarity and weighting scheme respectively.

6.5 Component 2: Implementing the Syntactic Decomposition Module

This component consists of the following modules:

1. The *part-of-speech* (POS) *tracking* module, which captures derivational morphology structures of content words.
2. The *lexical parser* module, which extracts expressive punctuation marks, Twitter-specific user conventions, and special symbols.

6.5.1 POS Tracking

Word embedding models capture statistical semantics between words based on the distributional hypothesis that words occurring in similar contexts tend to have similar meanings, where all words are processed in a similar manner. Such models also discard derivational morphology between words, such as the noun ‘beauty’ and the

adjective ‘beautiful’. To incorporate structural information, a syntactical feature vector is constructed for each tweet to capture stop words, nouns, verbs, adjectives, adverbs, and digits respectively. Unlike most existing methods that ignore function words in similarity computation, the proposed approach includes these as they carry structural information (Li et al., 2006), which contributes to the meaning in short texts such as tweets. However, function words contribute less to the meaning of a tweet as they appear frequently and therefore their value will be scaled down as discussed in Section 6.4.2. The POS tracking module tags each token in a tweet and populates its corresponding vector. For example, T_1 ‘what a nicely written story!’ and T_2 ‘is chapter 2 well structured?’ are represented in the syntactical vector space as [1, 1, 1, 0, 1, 0] for T_1 and [1, 1, 1, 0, 1, 1] for T_2 following the POS features shown in Table 6.4.

Table 6.4 The syntactical features in a tweet

Id	Syntactical group	Feature
1	*POS tags*	Stop word
2		Noun
3		Verb
4		Adjective
5		Adverb
6		Digit
7	*Twitter conventions*	Hashtag
8		Mention
9	*Punctuation marks*	Interrogation
10		Exclamation
11	*Special symbols*	Currency
12		Ratio

The syntactic similarity between T_1 and T_2 is the cosine between their vectors, which is 0.89. This computation is performed for candidate tweets and their syntactic similarity is derived by calculating the cosine angle (Aggarwal and Zhai, 2012) between their corresponding syntactic feature vectors.

6.5.2 Lexical Parser

Common Twitter conventions and punctuations are most likely to be removed in methods of semantic inferences in social data. However, in this research, the author’s hypothesis is that these symbolic structures are of no less importance than words in social contexts. Therefore, these symbolic conventions and punctuation provide information that cannot be discarded. This is particularly true in Twitter as users do not often follow a grammatical structure in tweets due to the informal nature of the social network. For example, consider the two tweets T_1 ‘going to Rome this weekend!’ and T_2 ‘going to Rome this weekend?’, although both tweets are

constructed from the same words, punctuating them differently changes the complete function of the tweet. The exclamation mark in T_1 expresses the user's excitement, whereas T_2 is an interrogative sentence expressing the user's uncertainty. Another common use of punctuations in informal contexts such as Twitter (albeit out of scope) is the sarcastic case. To further elaborate the role of expressive punctuations (i.e. interrogation and exclamation marks) in Twitter, the tweet 'Do I really need to mention this again!' has a latent rhetorical interrogation mark that indicates intended sarcasm.

While highlighting the role of expressive punctuation marks in Twitter demonstrates their importance in delivering the overall meaning of a tweet, common Twitter conventions (e.g. *#hashtags* and *@mentions*) are taken into account as well. Hash-tagging timely events and mentioning users over the network are frequently apparent in Twitter and almost every tweet contains at least one of them. The lexical parser module breaks down the tokens in a tweet and produces a list of the hashtags and mentions. Furthermore, special symbols (e.g. $ and %) are prevalent in tweets and carry syntactic information that cannot be ignored. The syntactical feature vector discussed in Section 6.5.1 is thus extended to accommodate further syntactical features, which are expressive punctuation marks, Twitter-based conventions, and special symbols. The complete list of syntactical features are provided in Table 6.4.

6.6 Computing the Semantic Similarity between Tweets

A tweet is decomposed into words and symbolic structures. Unlike classical methods that represents a sentence using a high dimensional static features (i.e. keywords) such as bag-of-words (BOW), TREASURE dynamically forms semantic and syntactic vectors solely based on the compared tweets. Recent research achievements in the complex field of computational linguistics and social network analysis are adapted as well to construct an efficient method of transforming a tweet into a representative semantic and syntactic feature vectors (Mikolov et al., 2013b, Naili et al., 2017, Alnajran et al., 2018c).

Given two tweets, T_1 and T_2, the proposed tweet similarity measure (TREASURE) forms a joint word set, from which the lexical semantic vectors are derived. The joint word set takes the following form:

$T = T_1 \cup T_2 = \{ w_{1_{T_1}}, w_{2_{T_1}} \ldots . w_m \}$.

Where *m* is the number of unique words in *T*, which is the joint word set that consists of all the unique words from T_1 and T_2. Unlike existing methods that consider different forms of a word such as *mouse* and *mice*, *cat* and *cats* which are considered as four distinct words in the joint word set *T* (Li et al., 2006), the proposed measure inserts the root of the word in *T*, for two reasons:

1. Unlike derivational morphology discussed in Section 6.5.1, in which the grammatical category of a word is changed, inflectional morphology does not change the essential meaning of a word.
2. Adding different forms of a words in the joint word set creates sparse vectors and introduces noise to the similarity computation algorithm.

Thus, the joint word set, *T*, for the two tweets, T_1 '*EU Referendum briefing on living and working in the UK #ProtectJobs*' and T_2 '*You must stay in the #EU to protect your job!*', is:

T = {EU Referendum briefing on living and working in the UK Protect Job you must stay to}.

Tracing shared words in the candidate tweets back to their morphemes in the joint word set creates a compact set with no redundant information, in this example, *you* represents both *you* and *your*. The joint word set, *T*, can be considered as the semantic features in the candidate tweets. Therefore, each pair of tweets is semantically represented by the use of *T* as follows: the joint word set is used to derive the lexical semantic vector, denoted by $\check{s}$, where each entry corresponds to a word in *T*. Thus, the dimension of the semantic vector, $\check{s}$, is equal to the length of the joint word set (i.e. number of words). The lexical semantic vector is denoted by v_{sem}, and values in the lexical semantic vector, $\check{s}_i (i = 1, 2, 3, \dots, n)$, is derived by computing the semantic similarity of the corresponding words embedding vectors $\check{v}_i$ in the tweet. Considering T_1 as an example:

***Case* 1**. If w_i is contained in the tweet T_1, $\check{s}_i$ is set to 1.

***Case* 2**. If w_i does not appear in T_1, the cosine coefficient is computed between the word embedding vector $\check{v}_i$ for w_i and each embedding vector corresponding to every word in the tweet T_1, using the method presented in Section 6.4.1. The highest similarity score ς obtained denotes the most similar word in T_1 to w_i if ς exceeds α threshold discussed in Section 6.4.4; otherwise, $\check{s}_i$ is set to 0.

Following the weighting schema discussed in Section 6.4.3, the value of an entry in

the semantic vector becomes:

$$s_i = \check{s}_i \,.\, W(g_i)$$

Equation 6.6 Entry value in the semantic vector

Where $W(g_i)$ is the weight of an *n*-gram (i.e. a word or a two-word phrase) in the joint word set. The product of the similarity and weight of g_i allows this entry of the semantic vector to contribute to the overall similarity based on their individual value. The semantic similarity between two tweets is derived by computing the cosine coefficient between the two semantic vectors corresponding to the tweets under consideration:

$$S_{sem}(T_1, T_2) = \frac{v_{sem}(T_1)\ .\ v_{sem}(T_2)}{\|v_{sem}(T_1)\|\ \|v_{sem}(T_2)\|}$$

Equation 6.7 The semantic similarity of T_1 and T_2

It is worth noting that TREASURE does not take into account the order of the words occurring in a tweet. This is based on two considerations: first, in tweets, unlike formal English sentences, users often use relaxed informal expressions that lack English grammatical structure rules. The character limit restriction impose misplacing adjectives and adverbs (e.g. *old silly fool* instead of *silly old fool*) and cutting off elements such as pronouns and conjunctions (e.g. *voting leave?* instead of *are you voting for leave?*) while supporting their meaning with emoticons for an ultimate usage of characters. Therefore, although English is not a free word order language, the free grammar nature of Twitter reduces the significance of word order in analyzing the semantic and syntactic structure and its contribution to the overall similarity between two tweets. Second, the proposed approach is composed of multiple modules to account for the necessary semantic and syntactic fragments of a tweet and thus, deferring computational costs and incorporating a word order similarity module would scale up the complexity even further.

6.7 Computing the Syntactic Similarity between Tweets

The syntactic similarity between two tweets is a combination of various syntactical features as discussed in Section 6.5. As a tweet enters the syntactic decomposition module, it is lexically parsed, tokenized, and tagged according to the POS it contains. The tweet is then represented by a syntactic feature vector, which transforms the syntactic information held in the tweet into a numeric vectorized representation. Consider a pair of tweets, T_1 and T_2, and their corresponding syntactic feature vectors,

$v_{syn}(T_1)$ and $v_{syn}(T_2)$ as follows:

T_1: An absolute disgrace! & again British kids get nothing!! #Brexit

T_2: Is @David_Cameron secretly taking us into another war while eyes are on #Brexit?

$v_{syn}(T_1)$: [3, 4, 1, 2, 0, 0, 0, 3, 1, 0, 0, 0]

$v_{syn}(T_2)$: [4, 3, 2, 0, 1, 0, 1, 0, 1, 1, 0, 0]

The syntactic feature vector, $v_{syn}(T_1)$, is derived by obtaining the syntactic features (as shown in Table 6.4, Section 6.5.1) for T_1, and similarly for T_2. The syntactic similarity between $v_{syn}(T_1)$ and $v_{syn}(T_2)$ is therefore a function of POS tags and lexical parsing of common Twitter convention. It is derived by computing the cosine coefficient between the syntactical feature vectors $v_{syn}(T_1)$ and $v_{syn}(T_2)$ as follows:

$$S_{syn}(T_1, T_2) = \frac{v_{syn}(T_1)\ .\ v_{syn}(T_2)}{\|v_{syn}(T_1)\|\ \|v_{syn}(T_2)\|}$$

Equation 6.8 The syntactic similarity of T_1 and T_2

The overall similarity between a pair of tweets is a combination of semantic and syntactical similarity at variable contributions, which are determined by empirical experiments.

6.8 Overall Tweet Similarity of TREASURE

As discussed in Section 6.2, the semantic and syntactic analogies between tweets play different roles in conveying the meaning of tweets. Therefore, the overall similarity between a pair of tweets is a combination of both semantic and syntactic similarities; each contributes according to its significance to the overall similarity score. The semantic similarity represents the potential meaning between words constructing a tweet, while the syntactic similarity provides information about the morphological structure of the words and common Twitter conventions used. Hence, the overall tweet similarity is defined in Equation 6.9 as a combination of semantic similarity and syntactic similarity.

$$S(T_1, T_2) = \delta\, S_{sem} + (1 - \delta) S_{syn}$$
$$= \delta \frac{v_{sem}(T_1)\ .\ v_{sem}(T_2)}{\|v_{sem}(T_1)\|\ \|v_{sem}(T_2)\|} + (1 - \delta) \frac{v_{syn}(T_1)\ .\ v_{syn}(T_2)}{\|v_{syn}(T_1)\|\ \|v_{syn}(T_2)\|}$$

Equation 6.9 Overall Similarity of T_1 and T_2

Where $\delta \leq 1$ determines the relative contributions of semantic and syntactic

information to the overall similarity score. However, it has been reported that syntactic information carry subordinate value for semantic processing of text (Wiemer-Hastings, 2000); δ should therefore be a value larger than 0.5, i.e., $\delta \in (0.5, 1]$ (Li et al., 2006).

6.9 Illustrative Example: Similarities for a Selected Tweet Pair

To illustrate how to compute the overall tweet similarity for a pair of tweets using the pre-trained word embedding model, the researcher provide below a detailed description of the measure for two example tweets:

T_1: Sterling falls substantially on #Brexit concerns!

$r_{sem}(T_1)$ = [sterling, falls, substantially, on, #brexit, concerns]

T_2: Is the pound falling on renewed Brexit worries?

$r_{sem}(T_2)$ = [is, the, pound, falling, on, renewed, brexit, worries]

The joint word set is:

T = {sterling falls substantially on brexit concerns is the pound falling renewed worries}.

The semantic features for T_1 and T_2 can be extracted from the joint word set, T. The process of deriving the semantic vector for T_1, using the proposed method, is shown in Table 6.5.

Table 6.5 Process for deriving the weighted semantic vector, W(š)

i	***T*(*w*$_i$)**	*sterling*	*falls*	*substantially*	*on*	*brexit*	*concerns*	***š***	**Weight (W(*T*(*w*$_i$)))**	**W(š)**
1	sterling	1						**1**	0.5452	0.5452
2	falls		1					**1**	0.6166	0.6166
3	substantially			1				**1**	0.7859	0.7859
4	on				1			**1**	0.279	0.279
5	brexit					1		**1**	0.2426	0.2426
6	concerns						1	**1**	0.5664	0.5664
7	is							**0**	0.2693	0
8	the				0.4765			**0.4765**	0.1967	0.1
9	pound	0. 6455						**0.6455**	0.5184	0.3346
10	falling		1					**1**	0.6001	0.6001
11	renewed							**0**	0.7301	0
12	worries						0. 5059	**0.5059**	0.5930	0.3

In the first row, the words in tweet T_1 are listed, whereas the first column contains the words, w_i, where $i \in \{1, 2, 3, \ldots, 12\}$, in the joint word set T. The words are sorted according to the order they appear originally. For each word in the joint word set, T, the values in the semantic vector are derived as follows:

1. If the identical word exists in T_1, the corresponding cell at the cross point is set

to 1.

2. If the root of the word exist in T_1, such as 'falls' and 'falling', the corresponding cell at the cross point is set to 1.
3. Else, the similarities between the word and every word in T_1 are computed and the cell at the cross point of the word with the highest similarity is set to the resulting similarity value, if this value exceeds the predefined threshold which is set to 0.3[4].
4. The word is assigned 0 if the highest similar word in T_1 is below 0.3.

For example, the word '*pound*' is not in T_1, but the most similar word is '*sterling'*, with a similarity of 0.65. Thus, the cell at the cross point of '*pound*' and '*sterling*' is set to 0.65. In the same manner, the word 'on' does not exist in T_1 and the most similar word to it holds a similarity value of less than 0.3, and therefore 0 is assigned. Other column cells are left empty, as their values are not required in demonstrating the similarity computation process. The semantic vector $\check{s}$ is obtained by selecting the largest value in each column. The resulting values are multiplied by the weight of the corresponding word in T, to account for the significance of the term. As a result, the semantic vectors for T_1, and similarly, T_2, are:

$v_{sem}(T_1)$ = {0.5452 0.6166 0.7859 0.279 0.2598 0.5664 0 0.1 0.3346 0.6001 0 0.3}

$v_{sem}(T_2)$ = {0.3519 0.6166 0 0.279 0.2598 0.2865 0.2693 0.1967 0.5184 0.6001 0 0.593}

From $v_{\text{sem}}(T_1)$ and $v_{\text{sem}}(T_2)$, the semantic similarity between the two tweets is S_{sem} = 0.781.

The syntactic vectors $v_{\text{syn}}(T_1)$ and $v_{\text{syn}}(T_2)$ are derived from the syntactical features that correspond to each tweet. The process of deriving the syntactic vectors, $v_{\text{syn}}(T_1)$ and $v_{\text{syn}}(T_2)$, as per the feature set shown in Table 6.4, Section 6.5.1, is shown in Table 6.6. Unlike semantic vectors, these are count-based vectors that record the number of occurrences for the different morphological structures and syntactical features in a tweet.

$v_{\text{syn}}(T_1)$ = {1 2 1 0 1 0 1 0 0 1 0 0}

$v_{\text{syn}}(T_2)$ = {2 3 3 0 0 0 0 0 1 0 0 0}

[4] Empirically derived threshold, word analogy values of less than 0.3 are intuitively too dissimilar. This value may change for different embedding models.

and, thus, $S_{syn} = 0.7646$.

Table 6.6 Process for deriving the syntactic vectors

Syntactic features	T_1	T_2
Function word	1	2
Noun	2	3
Verb	1	3
Adjective	0	0
Adverb	1	0
Digit	0	0
Hashtag	1	0
Mention	0	0
Interrogation	0	1
Exclamation	1	0
Currency	0	0
Ratio	0	0

Finally, the similarity between tweets "*Sterling falls substantially on #Brexit concerns!*" and "*Is the pound falling on renewed Brexit worries?*" is 0.78, using 0.8 for δ[5].

Although T_1 and T_2 do only share words *on* and *Brexit*, the algorithm is still aware of the similarity between the tweet pair. Traditional BOW methods (Barry et al., 2007) would result in a similarity of 0.2887, which is very low similarity measure, while the TREASURE measure computes a relatively high similarity. Thus, this example demonstrates that the proposed method can capture the meaning of the tweet regardless of the amount of common words.

6.10 Chapter Summary

This chapter has detailed the methodology for implementing the components of TREASURE. These integrated components will be evaluated in order to determine TREASURE STSS measurement accuracy. Consequently, gathering adequate evidence to answer one of the main research questions, which is '*Is it possible to intelligently measure the degree of semantic equivalence between OSN microblogging posts using an automated semantic computation method?*' Further evidence will be gathered in Chapter 7, where testing/evaluation methodology, experiments and results are carried out in order to fully address this research question.

The main novel contributions in this chapter are:

- A new pre-trained word embedding model based on unsupervised leaning of words co-occurrences from a large corpus in the EU Referendum political rich

[5] Empirically derived value through experiments on tweet pairs.

domain of controversial views. Unlike existing pre-trained models learned from traditional documents, this trained model provides a statistical semantic model that captures the behaviour and relationships between words used in the social context. This shall contribute to the success of different microblogging-based NLP applications in relevant domains.

- A novel hybrid statistical approach for microblogging STSS measurement that determines the overall similarity score based on the semantic relationships between n-grams as well as the inflectional morphology structure and common user conventions.
- A novel architectural design for English tweets STSS measurement, known as TREASURE that integrates semantic and syntactic components incorporating several corresponding modules, which can be extended to other microblogging OSNs and adapted to different languages.

Chapter 7 - TREASURE Evaluation Methodology and Results

7.1 Overview

In this chapter, the evaluation methodology for TREASURE is proposed in order to evaluate the effectiveness of the TREASURE STSS measure. In chapter 6, a novel TREASURE architectural design was proposed that incorporates collective integrated components and modules such as the word analogy, word embedding, weighting scheme, lexical analysis, and the similarity calculation algorithm. TREASURE uses semantic and syntactic features extracted from a pair of tweets to derive the corresponding feature vectors and compute subsequent similarity calculations in order to produce an overall similarity score.

The following sections outline the evaluation methodology used within three experiments designed to evaluate TREASURE.

1. **Experiment (1)** – this experiment was conducted with human participants to generate a benchmark of similarity-annotated tweet pairs on the political domain, from the EU Referendum dataset (the produced benchmark will be referred to as the EU_Referendum benchmark), which is a rich source of controversial views (data collection, pre-processing methodology, and features extraction are described in Chapter 5). The experimental methodology and design for this experiment is provided in Section 7.3.
2. **Experiment (2)** – this experiment uses the generated EU_Referendum benchmark to evaluate the strength of linear or monotonic association between TREASURE measurements and the human judgements derived from the EU_Referendum benchmark to test the first hypothesis, $\boldsymbol{H_A}$ (discussed later in this section). In this experiment, the pre-trained word embedding model on the EU_Referendum dataset (described in Chapter 6) is used to obtain semantic relationships between words. The experimental methodology and design for this experiment is provided in Section 7.5.
3. **Experiment (3)** – this experiment was conducted to assess the generalizability of TREASURE to a different domain, which is general news in twitter. The benchmark used in this experiment is SemEval-2014 STS.tweet_news (Guo et al., 2013) (described in Chapter 5) to test the second hypothesis, $\boldsymbol{H_B}$ (discussed later in this section). The Google News word embedding pre-trained model

(Mikolov et al., 2013b) learned from traditional Web documents was used in this experiment to obtain semantic relationships between words (described in Chapter 6). The experimental methodology and design for this experiment is provided in Section 7.5.

The results of the second and third experiments are used to compare TREASURE's evaluation results to the state-of-the-art as well as previous semantic similarity measures in order to test the third hypothesis, $\boldsymbol{H_C}$.

Therefore, the aim of the second experiment is to answer the research question related to Hypothesis A, $\boldsymbol{H_A}$, (a statistically significant correlation exists between TREASURE and human similarity judgments), which is:
Question A: Can TREASURE provide similarity measures that approximate human cognitive interpretation of similarity for microblogging posts?

The third experiment was conducted to test Hypothesis B, $\boldsymbol{H_B}$, (TREASURE can be generalized to different microblogging domains), which was designed to answer the following research question:
Question B: Does TREASURE demonstrate a statistically significant performance degradation when applied to a different domain?

The second and third experiments shall provide adequate evidence to test Hypothesis C, $\boldsymbol{H_C}$, (TREASURE achieves the highest correlation to human judgments among existing measures), designed to answer the following research question:
Question C: Does TREASURE demonstrate a statistically significant correlation with regard to existing STSS methods in the context of microblogs?

In order to test the aforementioned hypotheses, a set of intrinsic evaluation metrics are defined and justified. The use of these metrics require benchmark datasets that are ideally produced by human judgements with a good level of inter-judge agreement. The aim of the intrinsic evaluation is to test the three hypotheses, which are related to: the correlation of TREASURE with human judgements ($\boldsymbol{H_A}$), the generalizability of TREASURE to different domains ($\boldsymbol{H_B}$), and the effectiveness of TREASURE with regard to state-of-the-art STSS measures ($\boldsymbol{H_C}$).

7.2 TREASURE Overall Evaluation Methodology

The effectiveness of TREASURE in approximating human typical cognitive

perceptions on similarities in the context of microblogging social media was evaluated with reference to two benchmark datasets. The first benchmark is the SemEval-2014 STS.tweet_news that is labelled with human similarity ratings. The second benchmark was produced from the political EU Referendum dataset through an experiment with human experts to gather human similarity judgements on a set of tweet pairs using closed-ended questionnaires. The mean of the human ratings is computed and compared to TREASURE estimations by assessing the strength of linear association between the benchmarks (actual) and TREASURE (estimated).

7.2.1 Rationale for the Selection of the Evaluation Datasets

This section describes and justifies the two datasets used to evaluate TREASURE and test the research hypotheses provided in Section 7.2.2. Multiple benchmark datasets have been published for evaluating short-text similarity measures (O'shea et al., 2013) however, there are not many benchmark datasets produced on raw tweets.

Towards obtaining evidence to test the first hypothesis, $\boldsymbol{H_A}$, a dataset was collected from Twitter on the political domain of the EU Referendum (described in Chapter 5). A preliminary subset of 30 raw tweet pairs was derived from the EU Referendum dataset (Section 7.3.1.1). Benchmarks of 30 sentence pairs are commonly used in similar studies to evaluate semantic similarity measurement (Li et al., 2006, O'Shea et al., 2008a). This subset is used to produce a benchmark with human similarity ratings gathered from 32 participants through closed-ended questionnaires as described in Section 7.3.2 The description includes the experimental design, methodology, population, and sampling.

Furthermore, SemEval-2014 STS.tweet_news benchmark is utilised to evaluate the genralizability of TREASURE when applied in a different domain. This dataset is composed of 750 similarity-labelled pairs of tweets and news headlines on the general news domain. The use of this dataset will provide insightful evidence on the generalizability of TREASURE to a different and more general domain area.

7.2.2 Hypotheses

The main hypotheses of the experiments were:

$\boldsymbol{H_{A1}}$: A statistically significant correlation exists between TREASURE and human similarity judgments.

This hypothesis relates to TREASURE's ability to provide similarity measurements that are similar to humans' judgements.

$\boldsymbol{H_{A0}}$: A statistically insignificant correlation exists between TREASURE and human similarity judgments.

That is, TREASURE estimated similarity values and human actual scores do not demonstrate a strong linear relationship.

$\boldsymbol{H_{A1}}$**:** TREASURE can be generalized to different microblogging domains.

This hypothesis relates to the generalizability of TREASURE and the ability to apply it to different domains in the context of microblogging social media.

$\boldsymbol{H_{B0}}$**:** TREASURE cannot be generalized to different microblogging domains.

That is, TREASURE is domain specific and cannot be extended to measure the similarities for microblogging posts in different application domains.

$\boldsymbol{H_{C1}}$**:** TREASURE achieves the best correlation to human judgments amongst existing measures.

This hypothesis relates to the performance of TREASURE compared to existing related work.

$\boldsymbol{H_{C0}}$**:** TREASURE does not achieve the best correlation to human judgments amongst existing measures.

That is, there exist other STSS measures that perform better than TREASURE in the context of microblogs.

All the hypothesis ($\boldsymbol{H_A}$, $\boldsymbol{H_B}$, and $\boldsymbol{H_C}$) were tested using the subjective user evaluation judgements.

7.3 Experiment 1: Gathering Human Similarity Ratings on Tweet Pairs

This section describes the experimental design and instruments used for collecting human similarity ratings in order to produce a reliable EU_Referendum similarity benchmark, which will be used for the intrinsic evaluation of TREASURE. The human subjective similarity judgements on pairs of tweets were gathered using a closed-ended questionnaire. These judgements form a subjective qualitative control that is used to assess the strength of association between TREASURE and the human

judgements.

This section describes the methodology undertaken in constructing the following elements related to the human rating experiment:

1. The tweet pairs –this includes deriving a subset of 30 tweet pairs from the EU_Referendum dataset through an unsupervised sampling methodology.
2. The questionnaire design – this includes the design of the task instructions and the Likert scale such that minimal confusion is introduced to attain consistency between raters in order to achieve a reliable benchmark.

7.3.1 The Unsupervised Sampling Methodology for Deriving Tweet Pairs

A benchmark is ideally generated by human judges with a good level of inter-rater agreement (Schütze et al., 2008). However, the production of similarity judgments for the whole dataset of collected tweets is a labor-intensive process. Furthermore, manually generating pairs of tweets from the EU_Referendum dataset, which contains four million tweets is extremely expensive, if not impossible, and may introduce bias. Therefore, an unsupervised approach is required to derive a representative sample set of the political tweets in order to reduce the expensive process of judges' recruitment for generating the benchmark dataset.

An unsupervised semantic-based cluster analysis approach (SBCA) is implemented (described in Chapter 8) using the proposed similarity measure. The goal of using this cluster analysis to provide a suitable dataset for the human similarity experiment is twofold:

1. Generating pairs of tweets using the resulting clustroids and tweets (i.e. observations) at different distances to the clustroids to form pairs of tweets. The selected pairs of tweets are used for constructing the benchmark dataset of human judgments on similarity. This benchmark is then used for intrinsic evaluation of TREASURE, but will also be valuable for the wider research community.
2. Analysis of the generated clusters provides an extrinsic evaluation of the proposed tweet similarity method as it has been used in allocating tweets to the most similar cluster (i.e. clustering distance measure).

The clustering algorithm is implemented following a divisive approach such that all observations in the dataset start in one cluster. The cluster analysis commences by

assigning a random observation, T_r, as a cluster center. A recursive series of splits are subsequently performed based on comparing each observation with the derived clustroids. An observation, T_r, is assigned to a predefined cluster if it satisfies a certain threshold, τ_{sim}. Otherwise, a new cluster is generated and T_r is assigned as the new cluster's clustroid, T_c. This process recursively carries on until all observations in the dataset are assigned in clusters. Unlike most clustering algorithms that require the number of clusters to be determined beforehand, such as k-means, this approach does not apply this condition. Instead, the number of clusters in the dataset is directly proportional to the specified similarity threshold. This linear relationship implies that as the value of the threshold increases, more clusters are generated and vice versa. Based on an experiment conducted on a similarity-labelled Twitter dataset (detailed in Chapter 9), it has been empirically determined that a value of $\tau_{\text{sim}} = 3.0$ yields the most cohesive and separated set of clusters.

However, a cluster analysis of the entire EU Referendum dataset would be a complex and time consuming process (given the dataset size as discussed in Chapter 5 and algorithm complexity as discussed in Chapter 8). Therefore, a subset of the whole corpus of collected tweets is derived, such that the complete timeframe for the data collection process is spanned. Although it has been reported that 10% of a dataset is considered a representative sample set (Severino, 2006), collecting a random 10% of the whole dataset may introduce bias in the resulting tweets and miss out on important events.

Thus, the methodology for building a representative subset is conducted as follows:

1. The corpus of pre-processed tweets is divided into four groups according to the month a tweet has been streamed.
2. For each month during the data collection, the group of corresponding tweets is further split into four groups according to the week of tweet streaming.
3. The result is a corpus of tweets organized into four main groups corresponding to the four months of data collection and each group contains four subgroups according to the week a tweet has been streamed.
4. The representative subset is created by retrieving a random sample of 10% from each of the sixteen subgroups in order to span the entire data collection period.

This sampling methodology resulting in 13.7K tweets, not only ensuring a

representative set is collected in terms of size, but in content as well. The clustering algorithm is applied on the representative sample of tweets using the proposed similarity measure, TREASURE with a similarity threshold, τ_{sim} = 3.0. The unsupervised approach generated eleven non-overlapping clusters as summarized in Table 7.1. The representative tweets for each cluster are referred to as a "clustroids" instead of a "centroids" because tweets were clustered in a *non-Euclidean* space, and thus clustroids do not necessarily reside in the centre of a cluster (further elaborated in Chapter 8).

Table 7.1 Cluster analysis of political tweets on the EU_Referendum dataset

Cluster id	**Representative tweet (clustroid)**	**Cluster size**
1	*Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today*	2731
2	*EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats*	1840
3	*Sterling slides on renewed Brexit worries*	1719
4	*Brexit Emerges As Threat to TTIP Deal*	1682
5	*It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union*	1524
6	*Should the United Kingdom remain a member of the EU or leave the EU?, Opinium poll: Remain: 49% (-3) Leave: 51% (+3)*	1243
7	*Erdogan is an Islamic extremist who will flood the EU w #jihadists. Kick Turkey out of NATO and no admission to the EU. #Brexit*	987
8	*Both #HillaryClinton and #Obama continue to call on UK not to leave EU? If not EU #terror movement limited!*	688
9	*Brexit introduce controlled immigration system, deport those who support extremism*	604
10	*Terrorism is the scariest think. And it's ways more scarier if it's in the EU, in your home. Stay strong Brussels! #prayersforBrussels*	421
11	*It's just utterly stupid. Thank god UKIP will never get in power and Brexit will fucking fail.*	295

7.3.1.1 Deriving the Tweet Pairs for Human Similarity Annotation

In psychology, the capacity of information, *i*, that can be received, processed, and remembered in the immediate memory of a typical human cognitive system is seven plus or minus two (Miller, 1956), that is $i \in r$, where r = {5, 6, 7, 8, 9}. The methodology of producing the benchmark of similarity judgments from the EU Referendum dataset is based on this psychological theory. In order to make the annotation task as simple as possible for participants to complete, the experiment has been designed according to the results of the cluster analysis described in Section 7.3.1.

1. Each representative tweet, T_c, which is essentially the clustroid corresponding to each of the five biggest generated clusters are used to form one part in the pairs of tweets. Five clusters are used in order to avoid complexity and keep the experiment simple for the participants to follow as in Miller (1956) psychological experiment.
2. For each representative tweet, six tweets are randomly selected from the dataset and assigned to make up a pair.
3. This subsampling process is performed for each representative tweet in the biggest five generated clusters.
4. The resulting 30 pairs of tweets are used to form the human similarity EU_Referendum benchmark as shown in Table 7.2.

Table 7.2: Tweet pairs used in the similarity annotation experiment

Pair id	**Tweet**	**Representative tweet**
1	Brussels attacks may sway Brexit vote: Strategists	*Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today*
2	On one hand, there are decent human beings that send their sympathies to the Brussels victims and their families. And then there's Brexit.	
3	#Brussels attacks: Terrorism could break the EU and lead to Brexit	
4	Terrorism is the scariest think. And it's ways more scarier if it's in the EU, in your home. Stay strong Brussels! #prayersforBrussels	
5	Brussels Attacks Spur Brexit Campaign: Anti-Immigration Parties Link Terror To EU Open Borders	
6	The world is seriously fucked up right now.	
7	@caddenlimos connecting low paid workers doing non skilled jobs from Poland with terrorism in Belgium	*EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats*
8	Has anyone from the Brexit Brigade addressed the issue of what will happen to existing EU citizens living and working in the UK	
9	@thebobevans Today's atrocity foreseable under EU policy. Trust UK security services to protect UK citizens. Brexit	
10	#Brexit supporters claim EU needs UK more than we need it. 45% of UK exports go to EU, 10% of EU exports come here	
11	Could 2m+ 18-34 Year Old Workers Emigrating After a Brexit Cause a Recruitment Nightmare?	
12	We must stay in #EU to protect jobs	
13	Sterling slides on renewed Brexit worries - A decline of 1% marks the 25th day this year the pound has moved	*Sterling slides on renewed Brexit worries*
14	London-based crowdfunding platform Seedrs poll on	

	the EU referendum finds 47% of investors and 43% of entrepreneurs	
15	Brussels bombing rose Brexit risk. How to trade Pound in case of Brexit: GBP $USD #FX	
16	Sterling has dipped cause markets believe Brexit will happen-GOOD-spivs in the city will adjust after playing their gambling games	
17	In most scenarios #Brexit will impose a significant long-term cost on the UK economy #OEBrexit	
18	it's not just an economic argument	
19	Jobs rely on trade NOT a political union. Security relies on sharing information NOT a political union. #Brexit #StrongerIn #VoteLeave	*#Brexit Emerges As Threat To TTIP Deal*
20	#Brexit, a new threat to TTIP transatlantic trade talks	
21	Naive to think Brexit would solve the #ttip problem...only way to protect #NHS is for govt to exclude it from TTIP	
22	Benign Brexit would require accepting high levels of immigration and deep trade agreement with EU	
23	Brexit Risks Rising	
24	Negotiating trade agreements after #Brexit would be complicated for UK as there's no @wto for #services: @angusarmstrong8 at @FedTrust event	
25	UK's NHS will NOT survive staying in the EU	*It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union | via @Telegraph*
26	What would #Brexit mean for the #pharma industry?	
27	To the "expats" in spain who are moaning about immigration can i just say this to you? Jog the fuck on you UTTER hypocrites	
28	How can we save NHS inside EU	
29	We send £350 million to Brussels every week - enough to build a new NHS hospital every week. Let's #VoteLeave and #TakeControl	
30	The EU referendum is a vote for the EU or the NHS, we can't have both	

This sampling methodology is performed to prevent any bias being introduced by selecting the pairs included in the test data and also to avoid reliance on the TREASURE to perform the selection, which has not been evaluated by human experts yet.

7.3.2 The Questionnaire Design

This section describes the design of the questionnaire in terms of the instructions and guidance provided to the participants and the semantic descriptions for the Likert scale that will be used by participants to assign a similarity score for each tweet pair.

7.3.2.1 The Similarity Likert Scale

A Likert scale is a psychometric scale that ranges from a group of categories –least to

most– asking people to indicate how much they agree or disagree, approve or disapprove, or believe to be true or false (Allen and Seaman, 2007). Semantic descriptors (sometimes referred to as semantic anchors) are absolute descriptions identifying the similarity scale (Miller and Charles, 1991). O'Shea et al. (2010) provided evidence that semantic scale descriptors contribute to more consistent human judgments. The definitions in the similarity scale present in (Agirre et al., 2012) are set for general sentences pairs, in which similarities are more easily interpreted and distinguished than tweets.

7.3.2.2 Adaptation of the Similarity Scale

This section describes the adapted Likert scale for tweet-pair similarities and the descriptions associated with each level in that scale. A set of descriptors need to be identified to give the best approximation to intervals in a Likert scale for tweets. The 4-point scale validated semantic anchors defined by Charles (2000) show a very close agreement between the actual score and desired scores. Agirre et al. (2012), on the other hand, used an intuitively chosen scale point definitions for a 6-point scale, but this was not validated. The Likert scale points defined by Agirre were mapped in the constructed human similarity annotation experiment with the use of Charles' validated semantic anchor descriptors in order to produce an adapted 6-point Likert decimal scale.

The similarity scale points and definitions adaptation is performed in order to come up with semantic anchors that can better interpret the broader semantics in the tweets themselves and produce a reliable benchmark that has a good level of inter-rater agreement (Gwet, 2014). The adapted 6-point similarity scale for tweets is shown in Table 7.3. The first decimal point is used to introduce finer degrees of similarity (O'Shea et al., 2010).

Table 7.3: Adapted semantic anchors for tweets

Scale point	**Semantic anchor**
0.0	The overall meaning of the sentences is unrelated (on different topics).
1.0	The overall meaning of the sentences is vaguely similar (on the same topic).
2.0	The overall meaning of the sentences is clearly similar (share some details).
3.0	The overall meaning of the sentences is very much alike (missing/different important information).
4.0	The overall meaning of the sentences is strongly related (unimportant details differ).
5.0	The overall meaning of the sentences is identical (equivalent).

7.3.2.3 Instructions and Guidelines Provided to Participants

The participants were provided with an introduction to the study and the aim of undertaking this research. Due to the nature of the language used in microblogs, participants were told that they might find some of the words that are used in tweets offensive and that they can withdraw from the experiment at any time, if they wish. For the similarity annotation task (Appendix E, Section b), participants were provided instructions about the similarity rating process, containing the operational definition of similarity for participants to assign a value from 5.0 – 0.0 to each pair – the greater the similarity of meaning the higher the number. Potential variation arises from encouraging the use of the first decimal place (Rubenstein and Goodenough, 1965) as opposed to instructions which may encourage the use of integers only (Miller and Charles, 1991). Thus, participants were advised that they could use the first decimal place and the major scale points were also defined using the adapted semantic anchors shown in Table 7.3.

7.3.3 Sampling the Population for Participants

The aspiration to represent the general population is restricted due to three reasons:

1. Participants would be performing the similarity judgement task without supervision in order to avoid possibility of bias in their responses.
2. The tweet pairs are rich in political interrelated information and thus require adequate political background to be able to interpret the latent semantics. The younger population, although maybe more familiar with Twitter terminology, generally have less political background to qualify them in judging such rich semantic pairs.
3. A statistical analysis study[6] of the distribution of twitter users in the UK from 2012 to 2018, by age group revealed that an average of 55% of Twitter users are aged 25-54.

Thus, it was decided to restrict the sample to adults with graduate-level education. The sample was also restricted to include only native English speakers to ensure that the language used in the experiment is completely comprehensible and thus similarity judgments would not be influenced by anticipating text meaning or false interpretations. The 32 total participants volunteered without compensation. The use

[6] https://www.statista.com/statistics/271351/twitter-users-in-the-united-kingdom-uk-by-age/

of 32 participants is commonly considered a representative population sample in similar studies (O'Shea et al., 2008a, O'Shea et al., 2010, O'shea et al., 2013). Furthermore, a power analysis showed that 80% power for a large effect (effect size is identified in section 7.5.1) would require a total sample size of 32 participants (Faul et al., 2007). The human similarity rating experiment does not require collecting any personal information from any participant, such as age or gender, and therefore no sensitive personal data is held.

7.3.4 Results of Experiment 1: The EU_Referendum Benchmark

The production of the EU_Referendum similarity benchmark involved asking participants to complete a questionnaire, rating the semantic similarity of the tweet pairs on the scale from 0.0 (minimum similarity) to 5.0 (maximum similarity), as in Charles (2000) and Agirre et al. (2012). Tweets are listed according to their corresponding cluster to make up tweet pairs. These pairs are listed in a randomized order within each cluster. The two tweets making up each pair are the cluster representative tweet and the randomly selected tweet to prevent introducing any bias to the benchmark data (Section 7.3.1.1, Table 7.2). The participants were asked to complete the similarity annotation questionnaire in their own time and to work through from start to end according to the given instructions (the similarity annotation questionnaire is present in Appendix E, Section b). As discussed in Section 7.3.2.2, these instructions contain linguistic anchors for the 6 main scale points 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, adapted using (Agirre et al., 2012, Charles, 2000) (Table 7.3). The use of these anchors allows the application of similarity statistical measurements as they yield psychometric properties analogous to an interval scale (Charles, 2000). Each of the 30 tweet pairs was assigned a semantic similarity score calculated as the mean of the judgments obtained by the participants. These can be seen in Table 7.4, where all human similarity scores are provided as the mean score for each pair.

Table 7.4 The EU_Referendum similarity benchmark results

Pair Id	Tweet Pair	Human Similarity (Mean)	TREASURE Similarity Measure
1	a. Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today b. Brussels attacks may sway Brexit vote: Strategists	3.6	3.71
2	a. Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today b. On one hand, there are decent human beings that send their sympathies to the Brussels victims and their families. And then	3.85	3.78

	there's Brexit.		
3	a. Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today b. #Brussels attacks: Terrorism could break the EU and lead to Brexit	3.53	3.62
4	a. Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today b. Terrorism is the scariest think. And it's ways more scarier if it's in the EU, in your home. Stay strong Brussels! #prayersforBrussels	3.51	3.67
5	a. Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today b. Brussels Attacks Spur Brexit Campaign: Anti-Immigration Parties Link Terror To EU Open Borders	2.83	3.73
6	a. Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today b. The world is seriously fucked up right now.	0.45	2.73
7	a. EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats b. @caddenlimos connecting low paid workers doing non skilled jobs from Poland with terrorism in Belgium	1.93	2.54
8	a. EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats b. Has anyone from the Brexit Brigade addressed the issue of what will happen to existing EU citizens living and working in the UK	3.54	3.43
9	a. EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats b. @thebobevans Today's atrocity foreseable under EU policy. Trust UK security services to protect UK citizens. Brexit	0.53	2.28
10	a. EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats b. #Brexit supporters claim EU needs UK more than we need it. 45% of UK exports go to EU, 10% of EU exports come here	0.49	2.39
11	a. EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats b. Could 2m+ 18-34 Year Old Workers Emigrating After a Brexit Cause a Recruitment Nightmare?	2	2.46
12	a. EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats b. We must stay in #EU to protect jobs	3.52	2.99
13	a. Sterling slides on renewed Brexit worries b. Sterling slides on renewed Brexit worries - A decline of 1% marks the 25th day this year the pound has moved	4.77	4.44
14	a. Sterling slides on renewed Brexit worries b. London-based crowdfunding platform Seedrs poll on the EU referendum finds 47% of investors and 43% of entrepreneurs	0.83	2.79
15	a. Sterling slides on renewed Brexit worries b. Brussels bombing rose Brexit risk. How to trade Pound in case of Brexit: GBP $USD #FX	2.63	3.59
16	a. Sterling slides on renewed Brexit worries b. Sterling has dipped cause markets believe Brexit will happen-GOOD-spivs in the city will adjust after playing their gambling games	3.94	3.52
17	a. Sterling slides on renewed Brexit worries b. In most scenarios #Brexit will impose a significant long-term cost on the UK economy #OEBrexit	2.27	2.63

18	a. Sterling slides on renewed Brexit worries b. it's not just an economic argument	0.7	1.56
19	a. #Brexit Emerges As Threat To TTIP Deal b. Jobs rely on trade NOT a political union. Security relies on sharing information NOT a political union. #Brexit #StrongerIn #VoteLeave	0.99	2.84
20	a. #Brexit Emerges As Threat To TTIP Deal b. #Brexit, a new threat to TTIP transatlantic trade talks	4.92	3.98
21	a. #Brexit Emerges As Threat To TTIP Deal b. Naive to think Brexit would solve the #ttip problem...only way to protect #NHS is for govt to exclude it from TTIP	3.32	3.8
22	a. #Brexit Emerges As Threat To TTIP Deal b. Benign Brexit would require accepting high levels of immigration and deep trade agreement with EU	1.96	3.15
23	a. #Brexit Emerges As Threat To TTIP Deal b. Brexit Risks Rising	0.9	2.55
24	a. #Brexit Emerges As Threat To TTIP Deal b. Negotiating trade agreements after #Brexit would be complicated for UK as there's no @wto for #services: @angusarmstrong8 at @FedTrust event	2.93	3.31
25	a. It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union | via @Telegraph b. UK's NHS will NOT survive staying in the EU	4.74	4.45
26	a. It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union | via @Telegraph b. What would #Brexit mean for the #pharma industry?	0.93	2.97
27	a. It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union | via @Telegraph b. To the "expats" in spain who are moaning about immigration can i just say this to you? Jog the fuck on you UTTER hypocrites	0.3	3.31
28	a. It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union | via @Telegraph b. How can we save NHS inside EU	3.67	3.9
29	a. It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union | via @Telegraph b. We send £350 million to Brussels every week - enough to build a new NHS hospital every week. Let's #VoteLeave and #TakeControl	3.05	3.15
30	a. It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union | via @Telegraph b. The EU referendum is a vote for the EU or the NHS, we can't have both	3.91	4.06

In Table 7.4, some pairs are observed to have a significant difference between the actual (mean raters) and estimated (TREASURE) measurements, such as pairs 6, 9, 10, 23, and 27. In all these cases, TREASURE recorded a similarity score that is higher than the actual similarity between the tweet pair. This is attributed to the mechanism of the word analogy module, which computes the semantic relationships between words based on their co-occurrences in a lexical corpus. The EU_Referendum dataset (described in Chapter five) was used to train a neural network to generate word embedding vectors for each word in the dataset. Due to the corpus being domain-specific, words tend to occur in similar contexts. For example, the fact that offensive

and swear words (pairs 6 and 27) commonly co-occur with the EU Referendum terminologies such as Brexit and the NHS, their corresponding word vectors share similar weight representations. Consequently, the overall similarity of the tweet pair increases as a result of the similarities between the individual word vectors.

The subsequent section provides an analysis of the benchmark production in terms of the reliability of the actual ratings that were gathered from 32 participants and whether their ratings share a good level of agreement. The level of agreement among raters will determine the quality of the benchmark and the ability to use it in an intrinsic evaluation of TREASURE and other similar studies developed by the wider research community.

7.3.4.1 The Similarity Benchmark Reliability Analysis

The similarity judgments used to produce the human similarity benchmark from the EU_Referendum dataset were generated by human observers instructed to rate 30 pairs of tweets for semantic similarity following the 6-point Likert scale described in Section 7.3. The average of raters' judgments can only be trusted after demonstrating reliability. Inter-rater reliability (IRR) is the level of consensus among raters. Statistical measures are used to provide a logistical evidence that the agreement among raters' subjective assessments is beyond a simple chance (Klaus, 1980). That is, evaluating whether common instructions given to different observers of equivalent set of phenomena yields the same readings within a tolerable margin of error. The agreement observed among independent observers is the key to reliability (Hayes and Krippendorff, 2007). According to (Hayes, 2009), the more agreement among observers on the data they generate, the more comfortable we can be that their produced data can be exchangeable with data produced by other observers, reproducible, and trustworthy.

Varieties of measures are employed in existing academic research to compute inter-rater reliability. The lack of uniformity among studies is unlikely due to technical disagreement between researchers, but rather due to less sufficient information on how this test is calculated and how the results should be interpreted (De Swert, 2012). In this research, Krippendorff's alpha (Hayes and Krippendorff, 2007) (KALPHA), often denoted by α, is used as it has been suggested to be the standard reliability measure (Hayes and Krippendorff, 2007). It handles different sample sizes, generalizes across

scales of measurement; can be used with any number of coders, and satisfies the important criteria for a good measure of reliability. Krippendorff's alpha, α = .80 is generally brought forward as the norm for a good reliability test, with a minimum of .67 or even .60 (De Swert, 2012). Thanks to the work of Hayes and Krippendorff (2007), who made computing Krippendorff's alpha test easily accessible by developing a macro to make KALPHA calculation possible in SPSS. Figure 7.1 shows the computed alpha result for Krippendorff's test on the EU_Referendum human similarity benchmark.

The test gives a good inter-rater agreement, at α ≈ 0.8 for the production of the EU_Referendum human similarity benchmark presented in Section 7.3.4. Additionally, the bootstrapping procedure indicates that there is zero chance that the KALPHA would be below .70 if the whole population would be tested.

```
Krippendorff's Alpha Reliability Estimate


            Alpha     LL95%CI     UL95%CI       Units    Observrs       Pairs
Interval    .7805      .7644       .7766     30.0000     32.0000 14880.0000

Probability (q) of failure to achieve an alpha of at least alphamin:
   alphamin          q
      .9000     1.0000
      .8000     1.0000
      .7000      .0000                    KALPHA ≈ 0.8
      .6700      .0000
      .6000      .0000
      .5000      .0000

Number of bootstrap samples:
  10000

Judges used in these computations:
Columns   1 -  14
P1        P2        P3        P4        P5        P6        P7        P8
P9        P10       P11       P12       P13       P14
Columns  15 -  28
P15       P16       P17       P18       P19       P20       P21       P22
P23       P24       P25       P26       P27       P28
Columns  29 -  32
P29       P30       P31       P32

Examine output for SPSS errors and do not interpret if any are found

------ END MATRIX -----
```

Figure 7.1 The Krippendorff's alpha test result for the EU Referendum similarity benchmark

Therefore, the Krippendorff's alpha test results indicate that an intrinsic evaluation of the proposed similarity measure (TREASURE) can be conducted against the expert judgments with a relatively good confidence that the subjects are reliable enough to

make conclusions towards the measure's performance.

7.4 The Evaluation Methodology using Human Rating Benchmarks

This section describes the methodology carried out in conducting the evaluation for the second experiment using the EU_Referendum benchmark and the third experiment using the STS.tweet_news benchmark in order to answer the research questions outlined in Section 7.1.

The first experiment was conducted with human subjects, which produced the EU_Referendum benchmark as described in Section 7.3. On the other hand, the STS.tweet_news benchmark was published with pairs associated with human similarity ratings that were previously gathered by Guo et al. (2013). The subsequent sections describe the use of these benchmarks for intrinsically evaluating TREASURE to consequently address different questions.

7.4.1 Parameter Setting

As described in Chapter 6, TREASURE requires two parameters to be determined at the outset:

1. A threshold for deriving the semantic vectors from the word embedding model (Chapter 6, Section 6.4.2.2).
2. A weighting factor, δ, for determining the significance between semantic information and syntactic information (Chapter 6, Section 6.8).

The parameters in the evaluation experiments where empirically found using the benchmark datasets, evidence and methodology of previous publications (Li et al., 2006, Wiemer-Hastings, 2000) and intuitive consideration as follows: since syntax plays a relatively small role for semantic processing of text, the semantic computation is weighted higher, 0.8 for δ_{sem}, and consequently, 0.2 for the syntactic contribution, δ_{syn}. With regard to the semantic vector threshold, it has been determined considering two aspects: 1) detecting and utilizing similar words semantic characteristics to the greatest extent, and 2) keeping the noise low. These factors imply using a small semantic threshold, but not too small. A small threshold allows the model to capture sufficient sematic information of words distributed representations obtained by the neural embedding model. However, as the word embedding model represents word co-occurrence relationships, a too small threshold will introduce excessive noise to the model causing a deterioration of the overall performance. Based on these

considerations, different parameter values were experimentally observed and the appropriate values were identified using the tweets pairs' benchmark datasets. In this way, the researcher empirically found 0.3 for semantic vector threshold works well for the Google News as well as the EU_Referendum pre-trained word embedding models. Similarly, 0.8 for δ_{sem} works well for weighting the contribution of semantic and 0.2 for δ_{syn} syntactic information to the overall similarity in the EU_Referendum and STS.tweet_news benchmarks used in this research. Thus, both thresholds should be extended to different application domains in microblogging OSN.

7.4.2 Rationale for the Selection of Evaluation Metrics

This section presents the appropriate metrics used for evaluating TREASURE and explains the considerations taken into account for selecting these metrics.

7.4.2.1 Pearson and Spearman's Rank Correlation Coefficient

Pearson correlation is a parametric measure of linear association between two variables X and Y. It is denoted by the character *r* and has a value between -1 and +1 (1 is strong positive linear correlation, 0 is no linear correlation, and -1 is strong negative linear correlation). Pearson correlation can be obtained through computing Equation 7.1.

$$r = \frac{N \sum \mathrm{xy} - (\sum \mathrm{x} \sum \mathrm{y})}{\sqrt{[N \sum x^2 - (\sum x)^2][N \sum y^2 - (\sum y)^2]}}$$

Equation 7.1 Pearson's correlation coefficient (Pallant, 2013)

Where, *N* is number of observations, ∑xy is the sum of the products of paired scores, ∑x is the sum of x scores, ∑y is the sum of y scores, $\sum x^2$ is the sum of squared x scores, and $\sum y^2$ is the sum of squared y scores.

The usage of the Pearson correlation coefficient has been a common method for assessing the performance of STSS systems (Reimers et al., 2016). Pearson's *r* is obtained through computing the correlation between human judgments and machine assigned semantic similarity scores (Agirre et al., 2016a). As such, systems that record higher Pearson correlation coefficient are generally considered "accurate" STSS systems and would often be among the top choices for the system designer of an STSS based evaluation task. However, this common practice of STSS evaluation through Pearson correlation has been questioned previously. Zesch (2010) reported several limitations of the Pearson correlation as follows:

- Sensitive to outliers.

- Limited to measuring linear relationships.
- The two variables need to be approximately normally distributed.

Agirre et al. (2013) stated in the discussion of the results of the SemEval-2013 task about semantic textual similarity (STS): "Evaluation of STS is still an open issue" and that beside the Pearson correlation coefficient "...other alternatives need to be considered, depending on the requirements of the target application."

Zesch recommended the usage of Spearman's rank correlation coefficient (often referred to as Spearman's *rho*) in order to overcome the aforementioned limitations. Spearman's *rho* is a non-parametric test that is used to measure the monotonic relationship between two variables. It is not sensitive to outliers, non-linear relationships, and non-normally distributed data. This is because Spearman's correlation employs a ranking scheme instead of using the actual values to compute a correlation.

However, most evaluation methods of STSS systems including the SemEval semantic textual similarity shared tasks only report the Pearson correlation coefficient. Therefore, the experiment results were also evaluated via computing Pearson and Spearman's correlation coefficient to avoid uncertainty. The equation for calculating Spearman's rank correlation is as follows:

$$\rho = 1 - \left(\frac{6 \sum d^2}{n(n^2 - 1)}\right)$$

Equation 7.2 Spearman's rank correlation (Pallant, 2013)

Where, ρ is Spearman's rank correlation, d is the difference between the ranks of corresponding variables, and n is the number of observations. Although Pearson's and Spearman's coefficients tend to perform different calculations, the outcome of both of them is interpreted in the same way that is mentioned above.

7.4.2.2 Statistical Tests

The evaluation metrics described in Section 7.4.2.1 provide insights on the strength of the relationship association between the two variables (actual *vs*. estimated). In this research, the statistical test is used to measure the significance of this relationship (linear or monotonic depending on the normality distribution of the values) and thus, test the hypotheses.

Selecting the appropriate statistical technique for testing the hypothesis is the most difficult part when conducting research (Pallant, 2013). This is attributed to the lack of

a universal methodology that clearly guide researchers on the right statistical test choice (Kinnear and Gray, 1999). The challenge of this choice refers to the variations in the nature of research, as is depends on the type of research questions that needs to be addressed. In terms of the STSS measures, it also depends on the scale of similarity assignment, the variables to be analysed, the underlying assumptions for specific statistical techniques, and the nature of the data itself (Pallant, 2013).

Statistical techniques are generally divided in statistics into two diff erent approaches: parametric and non-parametric. The parametric test, such as t-tests and the Pearson's correlation coefficient, tend to make assumptions regarding the population, in which the sample has been drawn. These assumptions often relate to the shape of the population distribution. As per Gravetter and Wallnau (2016), parametric tests are inferential statistical analysis based on assumptions regarding the population and require numerical score. On the other hand, non-parametric techniques, such as Spearman's correlation coefficient do not employ such strict requirements nor do they make distribution assumptions, and therefore sometimes referred to as distribution free tests. These tests are most often used with categorical and ordinal data as they do not require that the data is normally distributed and are not based on a set of assumptions about the population (Nolan and Heinzen, 2011).

The normal distribution can be investigated either by observing the histograms or by performing the normality *goodness-of-fit* tests. The "test of normality" provides insight on the normality of the data and can be done by using Kolmogorov-Smirnov (K-S) test when the sample size is greater than 50 or Shapiro-Wilk test when the sample size is smaller than 50. It is generally agreed that significant values greater than 0.05 indicate that the data is similar to a normal distribution, otherwise the data significantly deviate from a normal distribution. However, as these tests are based on significance testing, making a judgement based solely on them can be misleading (Field, 2012). These tests can produce false significant *p*-values in large samples for small and unimportant effects even if these samples generally follow a normal distribution. Similarly, they will lack power to detect normality violations in small samples. Therefore, it is recommended to plot the data and make an informed normality decision based on both visual and statistical tests.

The normality histograms for the STS.tweet_news and the EU_Referendum datasets are available in Appendix F. The Kolmogorov-Smirnov normality test for the

STS.tweet_news dataset shown in Table 7.5 indicates that the variables are significantly different from a normal distribution, while the normality histograms show that the data generally follow a normal distribution.

Table 7.5 Test of normality for the STS.tweet_news dataset

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
STS.tweet_news_ACTUAL	.148	750	.000	.916	750	.000
STS.tweet_news_TREASURE	.072	750	.000	.968	750	.000
STS.tweet_news_LCH	.074	750	.000	.896	750	.000
STS.tweet_news_WUP	.071	750	.000	.928	750	.000
STS.tweet_news_WPATH	.064	750	.000	.980	750	.000
STS.tweet_news_PATH	.061	750	.000	.983	750	.000
STS.tweet_news_STASIS	.062	750	.000	.976	750	.000
STS.tweet_news_LIN	.058	750	.000	.972	750	.000
STS.tweet_news_RES	.050	750	.000	.987	750	.000
STS.tweet_news_JCN	.064	750	.000	.978	750	.000
a. Lilliefors Significance Correction						

Based on several observations, the data is considered to follow a normal distribution.

1. As the STS.tweet_news dataset is considered to contain large samples (n = 750), very small, inconsequential departures from a distribution might be deemed significant in a goodness-of-fit test (K-S test).
2. According to the central limit theorem (CLT), as sample sizes get larger (> 30 or 40) (Ghasemi and Zahediasl, 2012), the less the assumption of normality matters because the sampling distribution tends to be normal (Field, 2012).
3. The normality histograms demonstrate approximately normal distributions.

Hence, the parametric Pearson correlation coefficient will be used to examine the strength of linear association between the actual and estimated values, and the parametric paired sample t-test will be used to test the significance of this association. On the other hand, the EU_Referendum dataset (n = 30) is assumed to violate the assumption of normal distribution, which is generally the case for ordinal data generated according to a Likert scale. Table 7.6 shows the Shapiro-Wilk test (since the sample size is less than 50) shows that the data is not normally distributed for most of the samples and the histograms show that the data generally do not follow within the normality curve.

Table 7.6 Test of normality for the EU_Referendum dataset

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
EU_Referendum_ACTUAL	.159	30	.051	.920	30	.027
EU_Referendum_TREASURE	.095	30	.200*	.977	30	.742
EU_Referendum_STASIS	.097	30	.200*	.958	30	.279
EU_Referendum_WPATH	.131	30	.197	.972	30	.590
EU_Referendum_JCN	.179	30	.016	.890	30	.005
EU_Referendum_WUP	.127	30	.200*	.979	30	.803
EU_Referendum_LIN	.083	30	.200*	.982	30	.879
EU_Referendum_PATH	.147	30	.099	.930	30	.048
EU_Referendum_RES	.143	30	.119	.906	30	.012
EU_Referendum_LCH	.222	30	.001	.896	30	.007

*. This is a lower bound of the true significance.
a. Lilliefors Significance Correction

Therefore, the Spearman nonparametric test will be utilised for the strength of association and the non-parametric two-sample test will be used to test the hypothesis. This test is the nonparametric alternative to the repeated measure t-test, however, it converts scores to ranks and compares them instead of comparing the actual means of the two systems under study. It is worth noting that the Pearson correlation coefficient will also be calculated as it is generally used in evaluating STSS systems as discussed in Section 7.4.2.1.

7.5 Experiments 2 and 3: TREASURE Intrinsic Evaluation Results and Discussion

This section describes the evaluation results of TREASURE with reference to the EU_Referendum and the STS.tweet_news benchmark datasets following the considerations discussed in Section 7.4. The analysis of the results is demonstrated using correlations coefficients and inferential statistical analysis in order to derive sufficient evidence to test the three hypotheses outlined in Section 7.2.2. Section 7.5.1 provides analysis on the *strength* of association between the human subjective judgements on similarities and TREASURE'S produced similarity predictions. Section 7.5.2 analyses the *significance* of this association and addresses the first main research question (the second main research question is addressed in Chapter 9) set out in Chapter 1.

7.5.1 Correlation Results and Comparative Analysis

In statistics, the *effect size* is defined as "*information about the magnitude and direction of the difference between two groups or the relationship between two variables" (Durlak, 2009).* The Effect size will be measure according to (Cohen,

1988) criteria of 0.1 = small effect, 0.3 = medium effect, 0.5 = large effect.

The proposed STSS similarity measure (TREASURE) demonstrated a good correlation coefficient compared to human judgments for both datasets under consideration. TREASURE achieved 0.83 Spearman's correlation with reference to the EU_Referendum benchmark and a Pearson correlation of 0.776 was achieved with reference to the STS.tweet_news benchmark. The average performance of TREASURE is 0.8, which is the best correlation among state-of-the-art measure for tweet similarity.

7.5.1.1 The Comparison Criterion between Different Semantic Similarity Measures

TREASURE is compared against different levels of textual semantic similarity computation approaches in order to provide a thorough insight on the performance of TREASURE.

1. Concepts-based semantic similarity measures – the WordNet taxonomy is utilised to demonstrate the results of the concept-based measures to compute words semantic similarities:
    - Rada et al. (1989) proposed a similarity measure called "Distance" to assess the conceptual distance between a set of concepts, which is essentially the average minimum path length over all pairwise combinations of nodes between two graphs in a hierarchical taxonomy (edge-based approach). (**PATH**)
    - Wu and Palmer (1994) proposed a semantic similarity measure to improve some aspects in the PATH measure applied to an ontology. The authors considered the depth of the lexical taxonomy in the measure, because two concepts in lower levels of ontology are more specific and are more similar. (**WUP**)
    - Resnik (1995) proposed a new approach to measuring semantic similarity in an *is-a* taxonomy, based on the notion of information content (node-based approach). The information shared by two concepts is indicated by the information content of the concepts that subsume them in a lexical taxonomy. One key to the similarity of two concepts is the extent to which they share information in common,

indicated in an *is-a* taxonomy by a highly specific concept that subsumes them both. (**RES**)

- Jiang and Conrath (1997) propose a hybrid model that is derived from the edge-based notion by adding the information content as a decision factor (combined edge-based and node-based). In this approach, the lexical taxonomy structure is combined with corpus statistical information so that the semantic distance between nodes in the lexical taxonomy can be better quantified with the computational evidence derived from distributional analysis of corpus data. (**JCN**)
- Leacock and Chodorow (1998) tackled the problem of word sense disambiguation for the hypernomy and hyponymy semantic relations through combining local syntactic information with semantic information from WordNet. (**LCH**)
- Lin (1998) presents a definition of similarity that is claimed to be universal, which is derived from a set of assumptions. The universality of the definition is demonstrated by its applications in different domains as long as the domain has a probabilistic model. (**LIN**)
- Zhu and Iglesias (2017) main idea of semantic similarity method is to encode both the structure of the lexical taxonomy and the statistical information of concepts. It aims to give different weights to the shortest path length between concepts based on the information they share. The path length is used to describe difference and the common information is considered as commonality. (**WPATH**)

2. Formal English sentences STSS measure –focus on the semantic similarity between sentences that are composed of proper English words, which can be found in dictionaries.
    - Li et al. (2006) STSS measure generates an overall similarity score for a pairs of sentences, which is a combination of semantic and syntactic similarities. The semantic similarity part is based on computing the semantic relationships between words using WordNet, whereas the syntactic part is based on computing the word order similarity for the sentence pair. (**STASIS**)

3. Informal OSN-based STSS measure –like TREASURE, the focus is on the semantic similarity between short-text that is obtained from social networks, which consists of out-of-vocabulary (OOV) words and special characteristics.
    - Sultan et al. (2014) (**DLS@CU**) calculates the semantic similarity between two tweets based on the proportion of their aligned content words. The word alignment between two words is computed using the paraphrase database (PPDB[7]). If the two words, w_i and w_j, or their lemma are identical, then the similarity between them, $sim(w_i, w_j)$ is 1. If the two words are present as a pair in PPDB, the $sim(w_i, w_j)$ is 0.9. Otherwise $sim(w_i, w_j)$ is 0. DLS@CU was ranked the top performing STSS measure on SemEval-2014 semantic similarity task achieving 0.764 on the STS.tweet_news benchmark.

Table 7.7, Figure 7.2, and Figure 7.3 show the correlation coefficients, mean, and standard deviation for the ten semantic similarity measures on the EU_Referendum and the STS.tweet_news benchmarks. The correlation scatterplots are provided in Appendix G.

Table 7.7 Pearson (r), Spearman (ρ) correlations achieved by different STSS measures, mean (μ), and standard deviation (σ)

Category	Semantic similarity measure	EU_Referendum		STS.tweet_news	μ	σ
		ρ	r	r		
Concept-based	PATH	0.6	0.653	0.74	0.697	0.062
	WUP	0.601	0.579	0.54	0.56	0.028
	RES	0.074	0.004	0.313	0.159	0.218
	JCN	0.599	0.636	0.75	0.693	0.081
	LCH	0.147	0.087	0.319	0.203	0.164
	LIN	0.563	0.589	0.656	0.623	0.047
	WPATH	0.55	0.605	0.699	0.652	0.066
Sentence-based	STASIS	0.79	0.744	0.683	0.714	0.043
OSN-based	DLS@CU	-	-	0.764	0.764	-
	TREASURE	0.83	0.825	0.775	0.8	0.035

[7] A paraphrase database containing over 220 million paraphrase pairs (http://paraphrase.org).

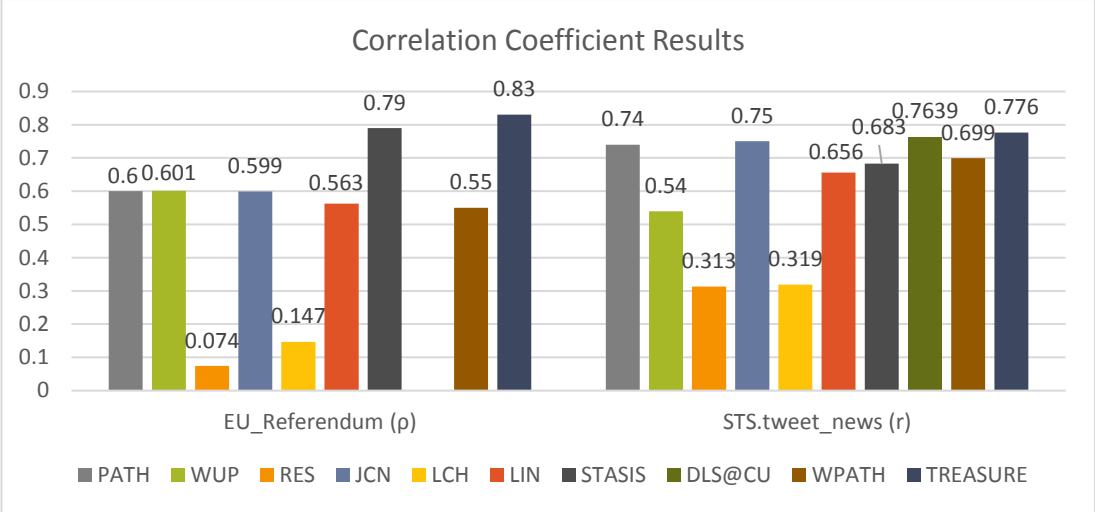


Figure 7.2 Correlation coefficient for different semantic similarity measure

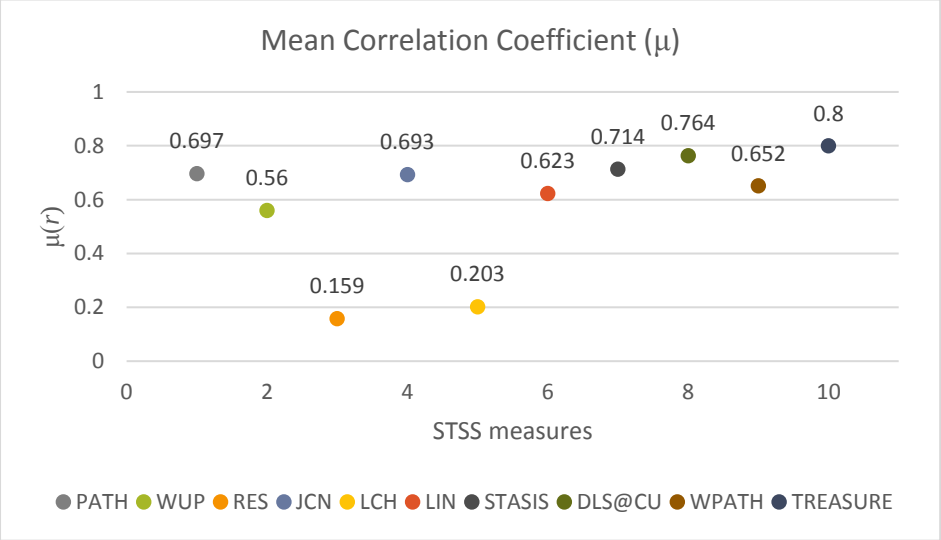


Figure 7.3 Mean correlation for different semantic similarity measure as shown in Table 7.7

Section 7.5.1 provided an intrinsic evaluation of TREASURE STSS measure to determine the strength of association between the measure's results (estimated) and the human similarity ratings (actual) obtained from two benchmarks, which are the EU_Referendum and the STS.tweet_news. The next section provides a statistical analysis of the results of TREASURE on both benchmarks in order to determine the significance of the linear and monotonic associations between the actual and estimated values.

7.5.2 Inferential Statistical Analysis

This section addresses the research questions set out in Section 7.1 through performing inferential statistical analysis according to the observations considered in Section 7.4.2. A statistical test concludes that the differences between two scores is

statistically significant, if the significance level α (p-value) is equal to or less than .05 (Pallant, 2013), presented as Asymp. Sig. (2-tailed). The classification of the data in terms of normality has been conducted through the tests demonstrated in Section 7.4.2.2. Accordingly, inferential statistical analysis techniques are employed for further investigation and testing of hypotheses.

7.5.2.1 Testing Hypothesis A

The aim of the second experiment is to test hypothesis $\boldsymbol{H_A}$, related to the following research question:

> *Question A: Can TREASURE provide similarity measures that approximate human cognitive interpretation of similarity for microblogging posts?*

The subjective evaluation of TREASURE on the EU_Referendum benchmark generated from experiment (1) described in Section 7.3, aims to evaluate strength of association between TREASURE (estimated) and the human similarity judgements (actual). As the actual and estimated values are non-normally distributed, a non-parametric test is carried out to assess the significance of this association to test the following hypothesis (test further justified in Section 7.4.2.2):

$\boldsymbol{H_{A0}}$: μd = 0 (that there is no monotonic association between the human similarity judgements and TREASURE measurements on the domain-specific dataset)

$\boldsymbol{H_{A1}}$: μd ≠ 0 (that there is a monotonic association between the human similarity judgements and TREASURE measurements on the domain-specific dataset)

Table 7.8 shows the Spearman's correlation and significance test results carried out to determine if there is sufficient evidence at the α level, determined earlier in this section, to conclude that there is a monotonic association between the estimated and actual similarity scores on the political domain of the EU Referendum.

Table 7.8 The non-parametric correlation significance for the domain-specific dataset

Correlations

			EU_Referendum_ACTUAL	EU_Referendum_TREASURE
Spearman's rho	EU_Referendum_ACTUAL	Correlation Coefficient	1.000	.830**
		Sig. (2-tailed)	.	.000
		N	30	30
	EU_Referendum_TREASURE	Correlation Coefficient	.830**	1.000
		Sig. (2-tailed)	.000	.
		N	30	30

**. Correlation is significant at the 0.01 level (2-tailed).

According to the results present in Table 7.8, there is a strong, positive monotonic correlation between the human similarity judgments and TREASURE measurements on the domain-specific microblogging posts (ρ = .83, n = 30, p < .001), indicating that ***H_{A1}* can be accepted.**

7.5.2.2 Testing Hypothesis B

The aim of the third experiment is to test hypothesis ***H_B***, related to the following research question:

> *Question B: Does TREASURE demonstrate a performance degradation when applied to a different domain?*

The subjective evaluation of TREASURE on the STS.tweet_news benchmark described in Section 7.3, aims to provide insights on the performance of TREASURE STSS measure applied in a generalized domain. The strength of linear relationship between TREASURE (estimated) and the human similarity judgements (actual) was determined to be strong as discussed in Section 7.6.1. In this section, as the actual and estimated values were considered to follow a normal distribution, a parametric test is carried out to assess the significance of this relationship in order to test the following hypothesis (test further justified in Section 7.4.2.2):

H_{B0}: μd = 0 (that there is no linear relationship between the human similarity judgements and TREASURE measurements on the general domain dataset)

H_{B1}: μd ≠ 0 (that there is a linear relationship between the human similarity judgements and TREASURE measurements on the general domain dataset)

Table 7.9 shows the Pearson's correlation coefficient and significance test results carried out to determine if there is sufficient evidence at the α level to conclude that there is a linear relationship between the estimated and actual similarity scores on the general-domain STS.tweet_news dataset.

Table 7.9 The parametric correlation significance for the general-domain dataset

Correlations

		STS.tweet_news_ACTUAL	STS.tweet_news_TREASURE
STS.tweet_news_ACTUAL	Pearson Correlation	1	.775**
	Sig. (2-tailed)		.000
	N	750	750
STS.tweet_news_TREASURE	Pearson Correlation	.775**	1
	Sig. (2-tailed)	.000	
	N	750	750

**. Correlation is significant at the 0.01 level (2-tailed).

According to the results present in Table 7.9, there is a strong, positive linear relationship between the human similarity judgments and TREASURE measurements on the general-domain microblogging posts ($r \approx .78$, $n = 750$, $p < .001$), indicating that ***H_{B1}* can be accepted.**

7.5.2.3 Testing Hypothesis C

The aim of this hypothesis is to test the significance of the difference between the correlation of TREASURE and other STSS measures in order to test ***H_C*** and address the following research question:

> *Question C: Does TREASURE demonstrate a statistically significant correlation with regard to existing STSS measures in the context of microblogs?*

Towards deriving the evidence, results of intrinsic evaluation performed in Section 7.5.1 are utilised. The evaluation results show that TREASURE achieves the highest mean correlation coefficient among other STSS measures that might have also demonstrated a strong correlation with reference to the EU_Referendum and the STS.tweet_news benchmarks. In this section, the difference between the correlations of TREASURE and the other highly correlated measures (Table 7.7), and whether the former demonstrates a statistically significantly higher correlation is investigated. The tests are carried out with TREASURE and the measures with the highest correlations from each category as discussed in Section 7.5.1. In this section, the significance tests are performed in order to test the following hypothesis:

H_{C0}: $\mu d = 0$ (that TREASURE dose not demonstrates a significantly higher correlation compared to existing STSS measures)

H_{C1}: $\mu d \neq 0$ (that TREASURE demonstrates a significantly higher correlation compared to existing STSS measures)

The first step in the comparison process involves converting the two correlation values under consideration into the standard form of *z* scores. The transformation of *r* to *z* is performed according to Table 11.1 in (Pallant, 2013). After transforming *r* to its corresponding *z*, Z_{obs} is obtained according to Equation 7.3.

$$Z_{obs} = \frac{z_1 - z_2}{\sqrt{\frac{1}{N_1 - 3} + \frac{1}{N_2 - 3}}}$$

Equation 7.3 Observed value of *Z* calculation (Pallant, 2013)

If the obtained Z_{obs} value is between -1.96 and +1.96, then the correlation coefficients cannot be considered statistically significantly different. Likewise, if Z_{obs} is not within this range, coefficients are statistically significantly different.

Table 7.10 shows the significance of the mean correlations differences between TREASURE and other STSS measures that are highly correlated with the EU_Referendum and the STS.tweet_news benchmarks.

Table 7.10 Significance of the difference between TREASURE and other STSS measures

	TREASURE	**DLS@CU**	**STASIS**	**PATH**
$\mu(r)$	.8	.764	.714	.697
Z_{obs}	0	1.99	4.1	4.48

The calculated values of Z_{obs} between the mean correlation of TREASURE and the other semantic similarity measures present in Table 7.10 are all over +1.96 (Z_{obs} > +1.96). Therefore, the test results provided that there is sufficient evidence to conclude that a statistically significant differences exist between the mean correlation coefficient of TRASURE and existing semantic similarity measures, that $\boldsymbol{H_{C1}}$ **can be accepted**.

7.6 Discussion

TREASURE achieved the best correlation compared to the other measures for both benchmarks used. With the use of uniform experiment settings and constant threshold parameter values, it can be observed that TREASURE performed better on the EU_Referendum benchmark than the STS.tweet_news benchmark. This can be attributed to three reasons:

1. **Characteristics of the test dataset** – the architecture of the developed algorithm is composed of semantic-based modules and syntactic-based modules. The latter is designed to extract syntactic features from raw tweets while the former generates semantic feature vectors upon performing certain steps of preprocessing. All tweet pairs in the EU Referendum political dataset retain Twitter-based user conventions and share relatively similar level of noise. This means that the syntactical feature vector is not biased with data in

one tweet that make up a pair. This is not the case in the STS.tweet_news benchmark, where each pair is formed of a typical tweet, which may contain hashtags and special symbols, and a corresponding news headline that is a typical sentence composed of formal English text. The lack of uniformity of the tweet pairs in the STS.tweet_news benchmark results in a performance deterioration of the syntactical similarity computation module, which consequently causes the accuracy of the overall similarity score to slightly degrade. Another factor that is worth discussing is the highly polarised tweets in the EU_Referendum dataset. Due to its nature, the referendum tweets are prone to different offensive and sensitive terminology as shown in Table 7.4. The fact that these terminology frequently occur in tweets, which are *pro* or *against* Brexit for varying reasons (e.g. NHS, trade, academia, etc.) has negatively influenced the performance of TREASURE as discussed in Section 7.3.4.

2. **Word embedding pre-trained model** – the core of the semantic processing is the word analogy module, which calculates the semantic relationships between words. This module computes the semantic relationship between word vectors generated by a neural embedding model. The effectiveness of this model depends on two factors: 1) *quality* (positive examples such as "cloudy sky" are more informative than negative examples such as "cloudy book") and 2) *quantity* (i.e. vocabulary coverage) of the learning text corpus. The Google News pre-trained model was used in the evaluation of the similarity algorithm on STS.tweet_news, whereas the political pre-trained model was used in the evaluation of the measure on EU_Referendum benchmark. While the Google News pre-trained model features a higher vocabulary coverage from a large corpus of Google News, it misses on some of the OOV words such as hashtags, slangs (e.g. *uhhhh, yummie, hmmm, WTF, damn, aww, ouch,* etc.), and event-specific vocabulary occurring in incredible velocity in tweets. This is due to the fact that the training corpora contain news articles, which are generally written in a formal structured language, in which words can be mapped to English dictionaries. Thus, the model learns distributed representations for words used in such documentation and misses out of vocabulary (OOV) words that are commonly used in tweets due to the character length restriction.

Therefore, although the model exhibits a large set of examples and vocabulary size, it does not provide a vectorized modelling for OOV words. This means that an embedding model, which is learned from tweets data is required in order to cater for the informal language used in social media contexts (Li et al., 2017). Therefore, the evaluation of the developed measure with reference to the EU_Referendum benchmark was performed using a word embedding model that was pre-trained with a corpus of political tweets instead of the Google News model. The correlations results shown in Table 7.7 demonstrate that, under the given experimental setting, a correlation enhancement of 5% when a Twitter-based neural embedding model is used to predict the semantic equivalence between tweets, rather than using a model trained on general data.

3. **Production of the gold standard labels** – similarity is highly subjective between humans and is linked to psychological and mental behaviors. Thus, in order to perform statistical tests and derive accurate conclusions on a measure that predicts human typical cognitive system, it is imperative to compare it against a benchmark produced by human experts with a good level of inter-judge agreement. The STS.tweet_news benchmark similarity ratings were assembled using AMT crowdsourcing (Buhrmester et al., 2011), gathering 5 scores per sentence pair. The similarity label score is represented as the mean of those five scores. It is worth noting that five annotators is a relatively low number of raters in order to generate a reliable benchmark (O'shea et al., 2013). This can be observed through example pairs where the similarity prediction measure produces a score that is intuitively more logical than the gold standard. For example, the pair *This is interesting: "What We Don't Know Is Killing Us"* and *Editorial: What We Don't Know Is Killing Us* is assigned a similarity score of 3.6, while the measure predicted score is 4.85. Such cases contribute to the decrease of correlation even though the measure intuitively seems to perform better than the gold standard. The non-logical labelled similarities observed can be attributed to a benchmark reliability problem of low inter-judge agreement. In contrast, the EU_Referendum benchmark was produced by 32 human observers who share a certain set of characteristics (nativeness, age, and education level). The generated benchmark features a good degree of reliability, at $\alpha = 0.8$. That is, the similarity measure can be statistically

evaluated against relatively uniform human psychometric properties that can be reproducible using other set of observers.

Table 7.10 in Section 7.5.1 shows that TREASURE achieves a significantly higher correlation among existing textual similarity measures in predicting the semantic similarity of tweets. For the SemEval-2014 semantic similarity shared task, the algorithm developed in (Sultan et al., 2014) achieved the best correlation coefficient on the STS.tweet_news benchmark among 38 other participating systems, at $r = 0.764$. The comparison of TREASURE similarity computation algorithm with the top scoring competitor shows that the former performed better when tested on the same dataset, at $r = 0.775$. Compared to STASIS, TREASURE achieved 9.2% better correlation on the STS.tweet_news and 8.1% on the EU_Referendum benchmarks. Comparing with concept similarity algorithms, JCN provides the closest performance to TREASURE, at $r = 0.75$, while RES recorded the least correlation for the STS.tweet_news benchmark, at $r = 0.313$. For the EU_Referendum benchmark, PATH comes after STASIS with 17.2% less correlation compared to TREASURE. Again, RES's results demonstrate a non-significant correlation on the EU_Referendum benchmark, at $r = 0.004$. The average of the measures correlation coefficient indicates that TREASURE outperforms the three type of measures under comparison, which are concept-based, formal, and informal short-text semantic similarity measurements for two Twitter-based benchmarks. STASIS (based on WordNet) achieved a very good correlation when evaluated on sentences composed with dictionary word definitions and DLS@CU (uses PPDB (Ganitkevitch et al., 2013)) performed as well on image descriptions, at $r = 0.816$ and $r = 0.821$ respectively. However, their performance has deteriorated when applied in the context of social data. It can be observed from the analysis results that such measures, which are based on lexical taxonomies achieved less correlation to human judgements when used for informal short text analysis. This is mainly attributed to the high proportion of OOV words present in microblogging posts. These words are more prevalent in the EU_Referendum benchmark, which is the reason behind the decrease in the correlations obtained by evaluation on this benchmark. TREASURE, unlike these algorithms, obtains its semantic calculations by learning distributed word representations from co-occurrences in large corpora of microblogging posts. This way, it is able to derive semantic relationships for the nature of modern language used in social media user generated context, which is absent in

traditional English knowledge bases such as WordNet.

7.7 Chapter Summary

This chapter has outlined and detailed the experimental methodology used to evaluate the new TREASURE STSS measure and illustrated the results to validate the architectural design proposed in Chapter 6 through conducting three experiments:

1. An experiment with human experts to produce an evaluation benchmark on a domain-specific microblogging dataset (Section 7.3).
2. An experiment to evaluate the correlation of TREASURE achieved with reference to the benchmark produced by the first experiment (Section 7.5).
3. An experiment to evaluate the generalizability of TREASURE through investigating its achieved correlation on a general-domain microblogging dataset (Section 7.5).

The Performance of TREASURE was evaluated by testing three hypotheses as follows:

- $\boldsymbol{H_A}$ – A statistically significant correlation exists between (TREASURE) and human similarity judgments.
- $\boldsymbol{H_B}$ – (TREASURE) can be generalized to different microblogging domains.
- $\boldsymbol{H_C}$ – (TREASURE) achieves the highest correlation to human judgments among existing measures.

The results from the experiments, using inferential statistical analysis with reference to subjective measures, show a significant evidence to support all of the hypotheses.

The main novel contributions in this chapter are:

- A new reliable benchmark of microblogging pairs labelled with similarity judgments by human experts with a good level of inter-rater agreement in the domain of Politics.
- A novel experimental methodology to produce a benchmark with human similarities from a large dataset of raw microblogging posts.
- An adapted set of semantic anchors (instructions) for tweet pairs that minimises confusion among raters in order to reduce the variance in the assigned similarity scores.
- An evidence that TREASURE STSS measure achieves a statistically significant correlation coefficient in the specific domain of politics at *p*-value

< .01, and demonstrates a strong monotonic association with human similarity judgements.

- An evidence that TREASURE STSS measure can be generalized to a different domain while achieving a statistically significant correlation coefficient, at *p*-value < .01, and demonstrating a strong linear relationship with human similarity judgements.
- An evidence that TREASURE STSS measure achieved a statistically significantly higher correlation ($Z_{obs} > +1.96$) compared to existing STSS semantic similarity measures.

Chapter 8 - The Semantic-Based Cluster Analysis (SBCA) Algorithm

8.1 Overview

Unsupervised machine learning has been a problem of intense discussion due to its potential in knowledge extraction for various applications and domains. As discussed in Chapter 3, much research have been conducted to tackle this problem for Information Retrieval (IR) systems by clustering context-rich documents. The problem is more complex in microblogging online social networks (OSN), where users generate highly unstructured content, such as tweets, which are short text posts that are often composed of informal English language. Due to the special characteristics of these tweets, traditional cluster analysis algorithms may not produce accurate results.

Little research has been undertaken towards clustering Twitter posts however; these existing methods (Garg and Rani, 2017, Inouye and Kalita, 2011, Bates, 2015) feature one or more of three weaknesses:

1. Require the number of clusters to be determined beforehand
2. Perform keyword-based clustering, which ignores the semantic relations between tweets
3. Model the text in a high dimensional vector space model (VSM) and use Euclidean distance to calculate similar microblogging posts.

In this chapter, a semantic-based cluster analysis (SBCA) algorithm is developed using the TREASURE (**T**weet simila**R**ity m**EASURE**) short-text semantic similarity (STSS) measure, described in Chapter 6 and evaluated in Chapter 7. The SBCA algorithm implements a novel approach towards the problem of semantic cluster analysis for microblogging posts. Unlike conventional partition-based clustering (discussed in Chapter 3) such as *k*-means, which requires the number of clusters to be determined beforehand, this new algorithm partitions the dataset through performing recursive iterations to produce the optimal number of clusters using a proximity measure. The proposed approach tackles the problem from a natural language processing (NLP) perspective, and uses TREASURE as the proximity measure to compute the semantic similarities between tweets.

This chapter aims to describe the development methodology of the novel SBCA unsupervised learning algorithm, which was designed to detect meaningful clusters (i.e. themes) in microblogging posts. The external evaluation methodology and experimental analysis of SBCA, which is conducted with reference to a multi-class benchmark are further elaborated in Chapter 9.

The rest of the chapter is outlined as follows; first the author briefly discusses the clustering algorithm's objective function in Section 8.2. In Section 8.3, the author describes the implementation methodology taking into consideration the proximity measure (8.3.1), the data structures (8.3.2), the clustroids' computation (8.3.4) and the algorithm's pseudocode (8.3.5), which is demonstrated with a flowchart. The author discusses SBCA's time and space complexities in Section 8.4 and summarizes the chapter in Section 8.5.

8.2 SBCA Objective Function

In unsupervised machine learning, "typical objective functions in clustering formalize the goal of attaining high intra-cluster similarity (documents within a cluster are similar) and low inter-cluster similarity (documents from different clusters are dissimilar)" (Schütze et al., 2008). This is a particular objective when all the features of the dataset under consideration are continuous numeric values such that distances between them can be measured in a Euclidean space. However, when clustering unstructured data such as microblogging posts, reaching the minimum/maximum value for the objective function does not necessarily imply that the intra-cluster instances are semantically homogeneous. Therefore, this cluster analysis problem requires a subjective evaluation criterion to determine the quality of the generated clusters (SBCA evaluation methodology is further elaborated in Chapter 9).

8.3 SBCA Implementation

In this section, the author describes the technical considerations carried out in the implementation of the semantic-based unsupervised algorithm proposed (SBCA) for clustering microblogging posts.

8.3.1 Proximity Measure

The SBCA algorithm incorporates TREASURE as the proximity measure upon which tweets are either grouped or separated according to a similarity threshold (detailed in

Chapter 9), τ_{sim} = 0.6[8] using Equation 8.1[9]. TREASURE demonstrates the core component of SBCA, which is the distance measure that will determine the semantic degree of intra-cluster and inter-cluster similarities between tweets. The TREASURE STSS measure is considered particularly applicable for clustering microblogging posts due to two reasons:

1. It is particularly designed to capture the similarities between Twitter posts, the most popular microblogging platform, and can be extended to other kinds of microblogging social networks (TREASURE evaluation results discussed in Chapter 7).
2. TREASURE is composed of both semantic and syntactic components to capture a comprehensive set of features from the text. The semantic modules compute the semantic relationships between words based on an artificial neural network embedding model learned from a large corpus of tweet examples. Whereas the syntactical modules capture structural and syntactical features that are common in microblogs, which contributes to the overall similarity score (TREASURE components and development methodology are described in Chapter 6).

TREASURE generates a similarity score following a 6-point Likert scale, $S \in [0,5]$, such that a score of 0.0 indicates no perceived similarity (i.e. largest distance) and 5.0 indicates the maximum perceived similarity, which in this case means the corresponding vectors are represented in the same point in a high dimensional vector space model (i.e. no distance) for the semantically identical vectors. As demonstrated in Chapter 7, participants (in the tweets similarity experiment) had the option to use the first decimal point in similarity ratings to show finer degrees of similarity. Thus, TREASURE was implemented in a way that produces real-value similarity scores such that it could simulate the human finer perceptions on similarities.

For a given pair of tweets, T_1 and T_2, the conversion process of the similarity measure (TREASURE STSS measurement), S, into a distance measure, d, is performed in two steps:

1. The similarity, S, is normalized to [0, 1] using the following equation:

[8] Empirically derived threshold by experiments on labelled tweet pairs (detailed in Chapter 9).
[9] This threshold is used by the SBCA proximity measure in deciding whether a tweet, *T*, will be assigned to an existing cluster or a new cluster is initiated for *T*.

$$S_{norm} = \frac{S(T_1, T_2)}{S_{max}(T_1, T_2)}$$

Equation 8.1 Similarity normalization

According to the 6-point Likert scale discussed in Chapter 7, the value of S_{max} in Equation 8.1 is five.

2. The corresponding distance measure, *d*, is then obtained using the following equation:

$$d(T_1, T_2) = 1 - S_{norm}(T_1, T_2)$$

Equation 8.2 Converting similarity to distance measure

Thus, the similarity threshold, $\tau_{sim} = 3$, is normalized using Equation 8.1, resulting into $S_{norm} = 0.6$, then converted to the corresponding distance measure using Equation 8.2, which finally comes to $\tau_{dis} = 0.4$.

8.3.2 Data Structures

A data structure is defined as, "a group of data elements used for organizing and storing data" (Tenenbaum, 1990). The data has to be organized in a manner that supports the efficiency of an algorithm, and data structures such as stacks, queues, linked lists, heaps, and trees provide different capabilities to organize data (Tenenbaum, 1990). In many existing studies, researchers tend to pay much attention to the type of algorithm implemented rather than the data structures used in the implementation. However, the right choice of the data structure used for a particular algorithm is always of the utmost importance as it may significantly improve the algorithm's runtime burden. For example, considering an algorithm designed to find the most similar pair in a dataset. The common implementation of this algorithm uses a 2-dimensional array to store the pairwise distances between pairs. The runtime complexity of traversing this 2-dimensional array to find the pair with the smallest distance is $O(n^2)$. An alternative implementation maps the pairwise distances to a *heap*, which is a binary tree that provides an efficient implementation of a priority queue. The runtime complexity is O(log *n*) for inserting an element into the heap and O(1) for retrieving the minimum distance pair, which is the node at the top of the heap binary tree.

The SBCA algorithm implements a local data structure for each cluster, namely a *dictionary*, *k*, a global *string* array, A_c, for the set of clustroids, and a global *dictionary* array, A_k, for the set of clusters. Instead of implementing a 2-dimensional array for

each cluster to store pair-wise distances and travers each row to find the tweet that has the minimum sum of distances, which is carried out in O(n^2), SBCA implements local dictionaries. These dictionaries consist of key-value pairs, where the key represents the short text part (i.e. tweet) and the value represents the sum of distances to other instances in the same cluster, which is carried out in O(n). The global array stores the tweets representing the centre (i.e. clustroids) for each generated cluster. The subsequent section describes the methodology undertaken for deriving a representative tweet for a cluster when a new tweet instance is assigned to that cluster.

8.3.3 Deriving Clustroids Based on Cluster Sizes

Clustering data points in a Euclidean space represents a cluster by its centroid, which is the center of gravity or the average of the points in the cluster (Leskovec et al., 2014). However, when the space in non-Euclidean, which is common in clustering unstructured text, distances cannot be based on location of points. Unlike continuous numerical data, microblogging posts are unstructured text that are not represented in a Euclidean space. This implies that cluster instances do not point to locations where the average distance can be calculated to produce a cluster centroid. In such case, a problem arises when each cluster requires a representative data point, but a collection of points cannot be represented by their centroid because the space is non-Euclidean. Multiple studies represent short text in a VSM (Laniado and Mika, 2010, Mozetič et al., 2018), which impose the curse of dimensionality problem (Leskovec et al., 2014). These approaches generate very sparse vectors that require intensive computational resources in order to compute the centroids in a high dimensional space.

The proposed algorithm aims to provide a globally optimal solution to the cluster analysis problem of microblogging posts. SBCA selects a point from the cluster instances to represent that cluster. This nominated data point, in some sense, lies in the center by picking up the tweet text that is, ideally, closest to all the points of that cluster. In this case, the cluster representative point is called the clustroid instead of centroid. The clustroid can be selected in various ways, each aiming to minimize the distances from the clustroid and every point in the cluster. An effective method is selecting the clustroid to be the point that minimizes the sum of distances to the other points in the cluster (Leskovec et al., 2014). After initializing a new cluster, the process of assigning data points to that cluster is illustrated in the algorithm's pseudocode as described in Section 8.2.4. For each cluster, SBCA derives the representative instance

(i.e. clustroid) through traversing all the instances in a cluster to determine the data point that is the most similar to the cluster instances.

In the proposed algorithm, deriving the clustroid is determined based on two interrelated constraints, *cluster size* (i.e. number of data points in a cluster) and *distance* (i.e. intra-cluster pairwise distances). The distance is computed depending on the cluster size, which is identified by the instances contained in that cluster. At any time in running SBCA, the clusters sizes would fall into one of the following four categories, where A is the global array and k is the local dictionary as discussed in Section 8.3.2:

1. *Singleton cluster* –this is the case when a new cluster is initialized, as it contains only one tweet, T, which is determined to be the clustroid, C. Thus,

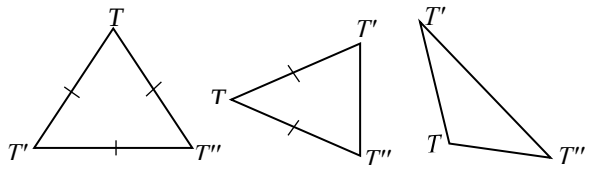
   $$C = T,\ k = \{\text{key: } T, \text{ value: } 0\},\ A = [C]$$

   Where *value* refers to the distance, which is zero because in this case, there is only one instance in the cluster.

2. *Doubleton cluster* –when a new instance, T', is assigned into a singleton cluster, the previous instance remains the clustroid of the cluster, C. Thus,

   $$C = T,\ k = \{\text{key: } T, \text{ value: } d(T, T'), \text{ key: } T', \text{ value: } d(T, T')\},\ A = [C]$$

3. *Tripleton cluster* –when a new instance, T'', is assigned into a doubleton cluster, the clustroid in this case is determined based on the distances between the triplet instances. SBCA identifies a cluster representative instance (i.e. clustroid) through modelling the three candidate instances based on a triangle geometric analysis in order to cover all possible cases (Bird, 2014). Each instance is assigned at an angle according to their pairwise distances calculated by inverting the TREASURE similarity to a distance measure to generate a triangle. Based on the pairwise distances between the three data points, which are candidate clustroids, the generated triangle can be one of the three cases shown in Figure 8.1. The pairwise distances between the candidate clustroids are modelled according to the three main types of triangles, where T, T', and T'' denote the three queued instances in the cluster, which are candidate clustroids.



(a) Equilateral (b) Isosceles (c) Scalene

Figure 8.1 Sides-based triangle classification

A triangle is a figure enclosed by three straight lines, where the sum of its three angles, $\angle ABC = 180°$ (Bird, 2014), where A, B, and C are the interior angles of the triangle. An angle degree refers to the direction of a triangle side, whereas the magnitude of the sides demonstrate the distance between two angles. In this research, the authors focus on the distance between instances rather than the direction (i.e. angle degree). Towards determining the new clustroid for a tripleton clusters, the distances between candidate clustroids represent a triangle straight lines, which can fall into one of the following cases:

Case 1. *Equilateral triangle* –figure 8.1.(a) represents $\Delta T'TT''$, a triangle in which all sides are equal. This means that the distances, *d*(*T,T'), d*(*T, T''),* and *d*(*T', T'')* are equal. In this case, the last assigned clustroid, *C*, remains unchanged, which is in this case, *T*, the first instance in the cluster.

$$\because\ d(T,T') = d(T,T'') = d(T',T'')\ \therefore C = T$$

Equation 8.3 Clustroid in *Equilateral triangle*

Case 2. *Isosceles triangle* – Figure 8.1.(b) represents $\Delta TT'T''$, a triangle in which only two sides are equal. This case represents one of two sub cases:

1. Size of the equal sides is less than the size of the third side such that,

$$d(\mathrm{T}',\mathrm{T}'') > \frac{d(\mathrm{T},\mathrm{T}') + d(\mathrm{T},\mathrm{T}'')}{2}$$

$$\because\ (TT' < T'T'') \wedge (TT' = TT'')\ \therefore C = T$$

Equation 8.4 Clustroid in *Isosceles triangle* (case 2.1)

   The clustroid, *C*, is set as the point that minimizes the sum of distances to other points, which is *T* in this case (Leskovec et al., 2014).

2. Size of the equal sides is greater than the size of the third side, Equation 8.4 becomes,

$$d(\mathrm{T}', \mathrm{T}'') < \frac{d(\mathrm{T}, \mathrm{T}') + d(\mathrm{T}, \mathrm{T}'')}{2}$$

$$\because \ (TT' > T'T'') \wedge (TT' = TT'') \ \therefore C = T'$$

Equation 8.5 Clustroid in Isosceles triangle (case 2.2)

In this sub case, even though *T* resides at an equally distant point to *T′* and *T″*, it does not represent the majority of the cluster's instances. Thus, *T′* instead is assigned as the new clustroid.

Case 3. *Scalene triangle* –the most common case where candidate clustroid instances have different pair-wise distances, such as Figure 8.1.(c)., which shows $\Delta TT'T''$, a triangle with unequal sides. In this case, the sum of distances is computed for each instance and the one with the minimum value is considered the representative instance, *C* (Leskovec et al., 2014).

$$\exists C \ \in \Delta TT'T'', C \ := \arg\min_{d} \sum f(x)$$

Equation 8.6 Clustroid in Scalene triangle

Where *f(x)* is the distance function *d*, between each instance, *x*, and other candidate instances, such that the point that satisfies the minimum sum of distances is set as the new clustroid. In the case present in Figure 8.1.(c), *C* = *T*.

Thus,

$C = min\sum_{i=1}^{m} d(T_i)$, k = {key: T, value:$\sum_{i=1}^{m} d(T_i)$, key: T', value: $\sum_{i=1}^{m} d(T_i)$, key: T'', value: $\sum_{i=1}^{m} d(T_i)$}, A = [C]

where *m* is the number of instances in the cluster.

4. *Multiple-instance cluster* –these clusters contain quadruple or more instances. When a new post is assigned into a tripleton cluster, the pair-wise distances between the new post and the cluster's instances are computed, *k* values are updated with the new sum of distances, and the new clustroid is derived from *k*, where the sum of pairwise distances is the minimum. As more instances are assigned into the cluster, the clustroids are derived in the same manner discussed here.

8.3.4 The SBCA Algorithm

The proposed algorithm (SBCA) performs recursive iterations over the collection of

data points (i.e. microblogging posts) and generates non-overlapping clusters. It implements a crisp partitioning methodology where each data point belongs to one and only one cluster. Table 8.1 presents a pseudocode of the implemented SBCA algorithm. It demonstrates the recursive iterations performed from initiating a new cluster to the stage where all data points are assigned to clusters.

Table 8.1 The SBCA algorithm pseudocode

Algoritm 2 SBCA for microblogging posts using TREASURE

function SBCA(E, τ):
Input: Let A_k be the array of cluster's dictionaries, k, A_c be the array of clustroids, C, and E be the dataset of microblogging posts, T_i, where $i = \{1, 2, 3, 4, \ldots, n\}$, $len(E) = n$, considered for cluster analysis, the distance threshold τ_{dis}.
Output: assignment of T to the relevant cluster dictionary, k, satisfying $d(T, C) < \tau_{dis}$, where C is the clustroid.
T ← *first*(*E*)
$k_1 \leftarrow T$
$c_1 \leftarrow T$
$A_k \leftarrow k_1$
$A_c \leftarrow c_1$
while *not* at end of *E* **do***:*
loop through each cluster center, A_c, where $c_i \in k_i$, $i = \{1, 2, 3, \ldots, len(A_c)\}$.
    T ← *next(E)*
    distance ← 1 – ($S(T, C_i)$/ $S_{max}(T, C_i)$)
    if *distance*[10] < τ_{dis} **then**
        assign *T* to k_i
        k_i, c_i = **UpdateSums**(*T*, k_i)
    else
        initialize new *'k*
        'k ← *T*
        'c ← *T*
        $A_k \leftarrow$ *'k*
        $A_c \leftarrow$ *'c*
end function SBCA(E, τ)

function UpdateSums(*T*, *k*):
Input: *T* is the new instance that will be assigned to the dictionary, corresponding to cluster *k*.
Output: *k* updated with new sums of distances for each instance after the insertion of *T*, and the new clustroid with the minimum sum.
min = 0
$C \leftarrow T$
foreach *j***,** *sum* in *k:*
j is an instance in k where $j \in \{1, 2, 3, \ldots, len(k)\}$
    sum ← sum + (1 – ($S(j, T)$/ $S_{max}(j, T)$))
    if min = *0*
      min = sum
    else
      if sum < min
        min = sum
        $C \leftarrow j$
return *k* **,** *C*
end function UpdateSums(*T*, *k*)

In Figure 8.2, a flowchart illustrates the overall process of the SBCA algorithm in

[10] Where *distance* τ_{dis} = 0.4 was derived from empirically determined similarity threshold.

assigning an instance, *T*, to a cluster in case the distance between *T* and the cluster's representative data point, *C*, is less than or equal to the distance threshold, $d(T, C) < \tau_{dis}$, or initiating a new cluster otherwise.

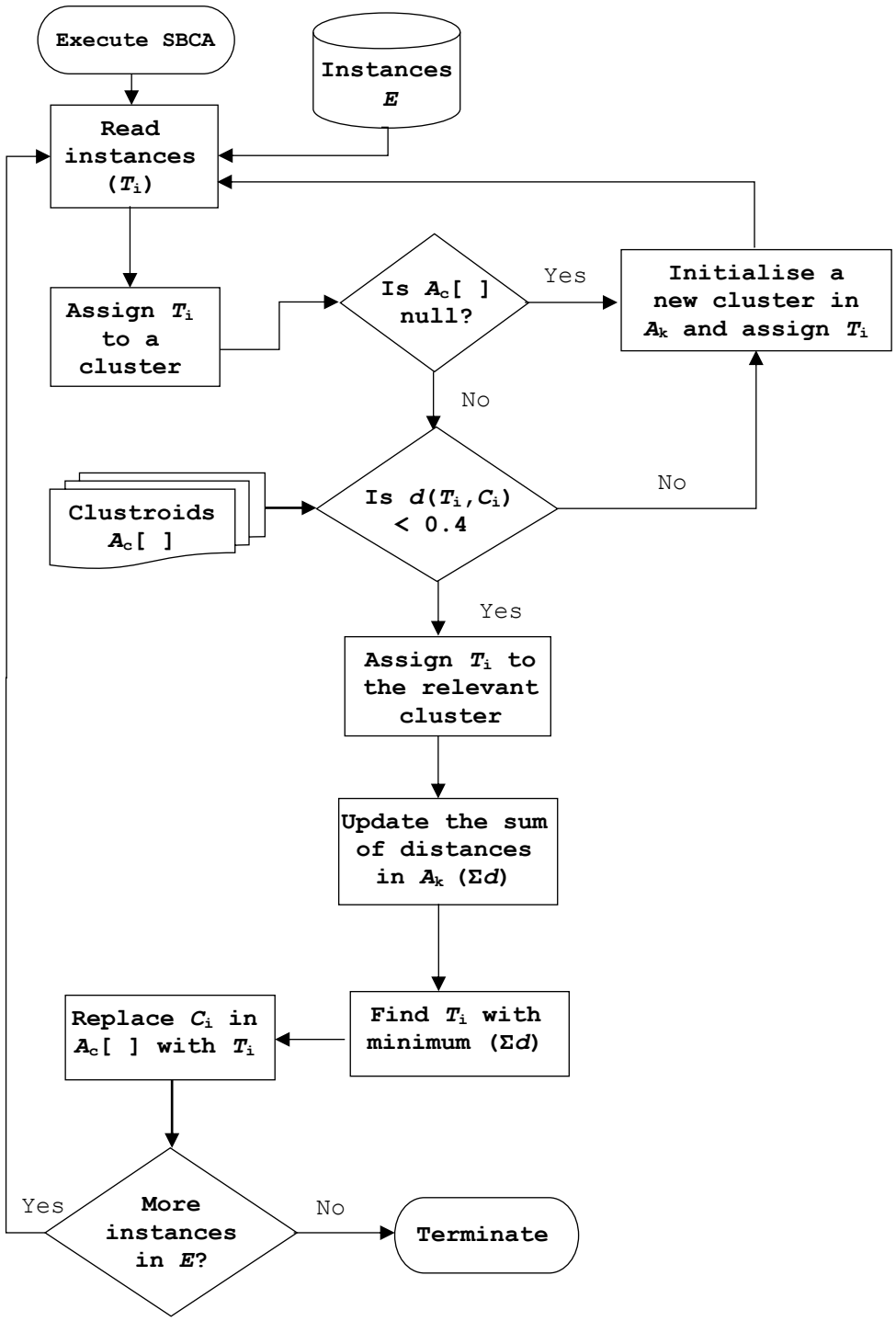


Figure 8.2 SBCA algorithm flowchart

The next section describes the SBCA algorithm's computational demand in terms of memory consumption and execution time.

8.4 SBCA Time and Space Complexity

In terms of complexity, the SBCA algorithm shares the same time complexity as *k*-means partition-based clustering (worst case is $O(n^2)$), which is generally considered a low computational cost algorithm (Salem et al., 2017). The space requirements for the SBCA algorithm are modest because only the data points are stored. Therefore, the specific storage requirements are

$$\text{Space complexity} = O((K+f)n) \text{ , hence } O(n)$$

Where K is the number of clusters, f is the number of features (i.e. attributes), and n is the number of data points. The run time requirement of SBCA is linear to the number of data points. In particular, the time complexity is

$$\text{Time complexity} = O(I*K*f*n), \text{ worst case would be } O(n^2)$$

Where I is the number of iterations required to update the sum of pairwise distances in each cluster. Therefore, SBCA is basically linear in the number of data points. This makes the SBCA algorithm quite efficient for clustering microblogging posts.

Compared to hierarchical approaches, the agglomerative (bottom-up) algorithm has a time complexity of $O(n^3)$, whereas the divisive (top-down) algorithm runs in even more time at $O(2^n)$ (Sharma et al., 2017), which means that the SBCA algorithm scales better to large datasets such as microblogging posts.

8.5 Chapter Summary

This chapter has detailed the methodology of implementing the SBCA algorithm including the proximity measure using TREASURE STSS measure (developed in Chapters 6 and evaluated in Chapter 7), clustroid computation, implementation pseudocode, and computational complexity. The algorithm's generated clusters will be evaluated with reference to benchmark datasets of microblogging posts using external evaluation criteria, which are further elaborated in Chapter 9. Experimental analysis will be carried out in order to answer the main research question outlined in Chapter 1, "Is it possible to automatically discover semantic themes in OSN microblogging posts based on an automated semantic computation method?" The testing/evaluation methodology, experiments and results are detailed in the next chapter.

The main novel contributions in this chapter are:

- A novel semantic-based clustering algorithm (SBCA) that incorporates TREASURE STSS new proximity measure for detecting semantic themes within microblogging posts. This SBCA algorithm can be used not only to generate clusters in a batch processing mode where all instances are contained in a corpus, but also in real-time as microblogging posts are being streamed.

Chapter 9 – The Semantic-Based Cluster Analysis (SBCA) Evaluation Methodology and Results

9.1 Overview

This chapter presents the design of an evaluation methodology for the semantic-based cluster analysis (SBCA) algorithm, which was proposed in Chapter 8. SBCA aims to dynamically detect non-overlapping semantic "themes" (i.e. meaningful clusters) in microblogging posts, particularly tweets, without having to determine a fixed number of clusters beforehand as with other partition-based clustering algorithms such as *k*-means. Typical objective functions in clustering numerical values formalize a single goal of attaining high intra-cluster cohesion and low inter-cluster cohesion. However, clustering textual instances such as tweets, require a subjective function for evaluating the semantic similarities of elements within clusters. This subjective function is obtained in a Twitter-based benchmark with tweets classified into categories. Due to the lack of such benchmarks, an experiment is performed to gather human classifications of tweets into clusters to form a benchmark from the EU_Referendum dataset (described in Chapter 5). The produced benchmark is used to evaluate the clusters generated by the SBCA algorithm.

In the SBCA algorithm, the TREASURE (**T**weet simila**R**ity m**EASURE**) short-text semantic similarity (STSS) measure proposed in Chapter 6 is used as the proximity measure, which plays a central role in the SBCA algorithm. Therefore, the subjective evaluation of the SBCA algorithm performs as an extrinsic evaluation of TREASURE (i.e. an indirect evaluation through a target application).

The evaluation methodology is carried out through undertaking three experiments designed to evaluate the SBCA algorithm as follows:

1. **Experiment (1)** – this experiment was conducted utilising the STS.tweet_news benchmark dataset (described in Chapter 5), which consists of similarity ratings for tweet pairs. This experiment was performed in order to determine the optimal value of TREASURE similarity threshold, τ_{sim}, which will determine if an instance will be assigned to an existing cluster or to a new cluster. The experimental methodology and evaluation of this experiment are provided in Section 9.2.

2. **Experiment (2)** – this experiment was conducted with human participants to generate a benchmark of tweets classifications into semantic categories utilising the EU Referendum dataset, which is a rich source of controversial views (data collection, pre-processing methodology, and features extraction are described in Chapter 5). The experimental methodology and design for this experiment are provided in Section 9.3.
3. **Experiment (3)** –this experiment used the threshold determined by experiment (1) in order to detect semantic themes within the EU Referendum dataset. The resulting clusters were evaluated using the benchmark generated from experiment (2). The experimental methodology and evaluation of this experiment is provided in Section 9.4.

The aim of conducting experiments 1, 2, and 3 is to answer the second main research question (the first main question was addressed in Chapter 7) outlined in Chapter 1, which is:

Is it possible to automatically discover semantic themes in OSN microblogging posts based on an automated semantic computation method?

Towards answering this main question, the SBCA algorithm was evaluated through application of different external evaluation criteria (described in Sections 9.2.2) with reference to a benchmark dataset in order to answer the subsequent questions that correspond to the second main research question (the first main research question was addressed in Chapter 7).

1. *Can the SBCA algorithm generate pure clusters?*
2. *Can the SBCA algorithm generate accurate clusters by undertaking correct separation and combining decisions with reference to a benchmark?*

9.2 Experiment (1): Deriving the Optimal SBCA Parameter Value

This experiment was implemented in order to derive the optimal similarity threshold value, τ_{sim}, for the proximity measure (TREASURE) used in the SBCA algorithm. The resulting coarse-grained or fine-grained clusters is determined by the value of this threshold. A higher value of τ_{sim} is expected to generate a larger number of granularities with low intra-cluster variance and high inter-cluster variance. That is, as τ_{sim} approaches the upper bound of the similarity scale [0, 5], $\tau_{sim} \rightarrow 5$, nearly each instance in the dataset will end up in a singleton cluster. In contrast, a lower value of τ is expected to generate less granularities with higher intra-cluster variance and lower

inter-cluster variance. Hence, as τ_{sim} approaches the lower bound of the similarity scale [0, 5], $\tau_{sim} \rightarrow 0$, all instances in the dataset will end up in a single cluster. Therefore, the aim of this experiment is to determine the optimal value of τ_{sim} using the STS.tweet_news similarity labelled dataset.

9.2.1 Experiment (1) Evaluation Methodology using the STS.tweet_news Benchmark

The STS.tweet_news benchmark dataset consists of tweet pairs that are annotated with similarity ratings, which was used to evaluate TREASURE (TREASURE evaluation is present in Chapter 7). The lack of Twitter-based benchmarks that are annotated with actual multi-class classification of tweets that can be used to evaluate an unsupervised clustering algorithm has led to running the SBCA algorithm on the STS.tweet_news similarity benchmark dataset. The application of the evaluation metrics discussed in the subsequent section for different values of τ_{sim} is carried out to determine the optimal value for detecting semantic themes in Twitter feeds, which can be extended to different microblogging posts.

9.2.1.1 Rational for the selection of the external evaluation criteria

The STS.tweet_news benchmark dataset does not consist of classes from which each instance belongs. Therefore, it is imperative to design an evaluation methodology such that a similarity labelled benchmark can be utilised for the purpose of cluster analysis evaluation. The evaluation of the proposed clustering algorithm on the STS.tweet_news benchmark in order to determine the optimal value of the similarity threshold, τ_{sim}, is performed through four external evaluation criteria as follows:

1. **Rand index (RI)–**considers the assignment of tweets to clusters according to a series of decisions. That is, two tweets should be assigned to the same cluster if and only if they are similar. A true positive (TP) decision assigns two similar tweets to the same cluster, whereas a true negative (TN) decision assigns two dissimilar tweets to different clusters. There are two types of errors that can be committed by a clustering algorithm. A false positive (FP) decision assigns two dissimilar tweets to the same cluster, whereas a false negative (FN) decision assigns two similar tweets to different clusters. The *Rand index* is used to measure the percentage of decisions that are correct, which is simply

accuracy. Equation 9.1 is used to compute the *Rand index* of the SBCA resulting clusters.

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}$$

Equation 9.1 Rand Index (Schütze et al., 2008)

2. **Precision (*P*) and Recall (*R*)** –*P*/*R* are the most common measurements for evaluating classifiers, which can be used to evaluate the grouping decisions determined by a clustering algorithm. Precision is interpreted as, out of the instances that were grouped in the same cluster, how many of them are actually semantically similar. Whereas recall determines the percentage of actually similar instances that ended up in the same cluster. Therefore, in addition to the *Rand index*, precision and recall are used, which are formally presented in Equation 9.2 and Equation 9.3 respectively.

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Equation 9.2 Precision (Schütze et al., 2008)

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Equation 9.3 Recall (Schütze et al., 2008)

3. **F-measure** – this metric is defined as the weighted harmonic mean of precision and recall. While the *Rand index* gives equal weight to FPs and FNs, separating similar documents is sometimes worse than putting pairs of dissimilar documents in the same cluster. Therefore, *F measure* can be used to penalize false negatives more strongly than false positives by selecting a value $\beta > 1$, thus giving more weight to recall.

$$F_\beta = \frac{(\beta^2 + 1)\mathrm{PR}}{\beta^2 \mathrm{P} + \mathrm{R}}$$

Equation 9.4 *F*-measure (Schütze et al., 2008)

For each of the aforementioned evaluation metrics, the SBCA algorithm is executed for six consecutive cases. Each case uses a different value of τ_{sim} in order to determine the optimal parameter threshold value for the proximity measure (TREASURE). The proportion of correctly clustered observations determines the accuracy of the clustering algorithm. The higher this proportion, the better the algorithm.

Thus, the SBCA algorithm is evaluated on six different similarity thresholds τ_{sim},

spanning the three similarity ranges used in (Dai et al., 2017), which are:

- The *lower* bound, [0 – 2]
- The *neutral* bound, (2 – 3]
- The *upper* bound, (3 – 5]

From each range, two threshold values are used in the evaluation of the SBCA algorithm, such that, if a tweet, *T*, and a clustroid, *C*, has a similarity, $S(T, C) > \tau_{sim}$, *T* is assigned to the cluster where C is the representative tweet for. Otherwise, *T* is assigned to a new cluster (the SBCA algorithm is detailed in Chapter 8).

The next section describes the SBCA results for each value of τ_{sim} using the aforementioned evaluation metrics along with a discussion on the value that provided the most accurate clusters according to the STS.tweet_news similarity labelled benchmark.

9.2.2 Experiment (1) Results and Discussion

The results of the evaluation metrics described in Section 9.2.1.1 can be derived using a contingency matrix of the decisions undertaken by the SBCA algorithm against the actual decisions as defined in Table 9.1.

Table 9.1 Contingency matrix

	Same cluster	**Different clusters**
Same class	**TP**	**FN**
Different classes	**FP**	**TN**

Case 1. (τ_{sim} = 1.5) –In this case, if $S(T, C) > 1.5$, $C \leftarrow T$. Otherwise, $C_{new} \leftarrow T$. Table 9.2 shows the decisions of the SBCA algorithm for τ_{sim} = 1.5 with reference to the STS.tweet_news similarity labelled benchmark.

Table 9.2 The contingency matrix for τ_{sim} = 1.5

	Predicted: Yes	**Predicted: No**	
Actual: Yes	TP = 328	FN = 315	643
Actual: No	FP = 9	TN = 98	107
	337	413	

From Equation 9.1, **RI** = (328+98)/750 = 0.568

From Equation 9.2, **P** = 328/(328+9) = 0.973

From Equation 9.3, **R** = 328/(328+315) = 0.51

From Equation 9.4, **F** = (2*0.973*0.51)/(0.973+0.51) = 0.669, where β = 1

Case 2. (τ_{sim} = 2) –In this case, if S(*T*, *C*) > 2, *C* ← *T*. Otherwise, C_{new} ← *T*. Table 9.3 shows the decisions of the SBCA algorithm for τ_{sim} = 2.0 with reference to the STS.tweet_news similarity labelled benchmark.

Table 9.3 The contingency matrix for τ_{sim} = 2.0

	Predicted: Yes	**Predicted: No**	
Actual: Yes	TP = 342	FN = 249	591
Actual: No	FP = 9	TN = 150	159
	351	399	

From Equation 9.1, **RI** = (342+150)/750 = 0.656

From Equation 9.2, **P** = 342/(342+9) = 0.974

From Equation 9.3, **R** = 342/(342+249) = 0.579

From Equation 9.4, **F** = (2*0.974*0.579)/(0.974+0.579) = 0.726, where β = 1

Case 3. (τ_{sim} = 2.5) –In this case, if S(*T*, *C*) > 2.5, *C* ← *T*. Otherwise, C_{new} ← *T*. Table 9.4 shows the decisions of the SBCA algorithm for τ_{sim} = 2.5 with reference to the STS.tweet_news similarity labelled benchmark.

Table 9.4 The contingency matrix for τ_{sim} = 2.5

	Predicted: Yes	**Predicted: No**	
Actual: Yes	TP = 351	FN = 195	546
Actual: No	FP = 21	TN = 183	204
	372	378	

From Equation 9.1, **RI** = (351+183)/750 = 0.712

From Equation 9.2, **P** = 351/(351+21) = 0.944

From Equation 9.3, **R** = 351/(351+295) = 0.643

From Equation 9.4, **F** = (2*0.944*0.643)/(0.944+0.643) = 0.765, where β = 1

Case 4. τ_{sim} = 3 –In this case, if S(*T*, *C*) > 3, *C* ← *T*. Otherwise, C_{new} ← *T*. Table 9.5 shows the decisions of the SBCA algorithm for τ_{sim} = 3.0 with reference to the STS.tweet_news similarity labelled benchmark.

Table 9.5 The contingency matrix for τ_{sim} = 3.0

	Predicted: Yes	**Predicted: No**	
Actual: Yes	TP = 380	FN = 77	457
Actual: No	FP = 36	TN = 257	293
	416	334	

From Equation 9.1, **RI** = (380+257)/750 = 0.849

From Equation 9.2, **P** = 380/(380+36) = 0.913

From Equation 9.3, **R** = 380/(380+77) = 0.832

From Equation 9.4, **F** = (2*0.913*0.832)/(0.913+0.832) = 0.871, where β = 1

Case 5. τ_{sim} = 3.5 –In this case, if S(*T*, *C*) > 3.5, *C* ← *T*. Otherwise, C_{new} ← *T*. Table 9.6 shows the decisions of the SBCA algorithm for τ_{sim} = 3.5 with reference to the STS.tweet_news similarity labelled benchmark.

Table 9.6 The contingency matrix for τ_{sim} = 3.5

	Predicted: Yes	**Predicted: No**	
Actual: Yes	TP = 337	FN = 25	362
Actual: No	FP = 124	TN = 264	388
	461	289	

From Equation 9.1, **RI** = (337+264)/750 = 0.801

From Equation 9.2, **P** = 337/(337+124) = 0.731

From Equation 9.3, **R** = 337/(337+25) = 0.931

From Equation 9.4, **F** = (2*0.731*0.931)/(0.731+0.931) = 0.819, where β = 1

Case 6. τ_{sim} = 4 –In this case, if S(*T*, *C*) > 4, *C* ← *T*. Otherwise, C_{new} ← *T*. Table 9.7 shows the decisions of the SBCA algorithm for τ_{sim} = 4.0 with reference to the STS.tweet_news similarity labelled benchmark.

Table 9.7 The contingency matrix for τ_{sim} = 4.0

	Predicted: Yes	**Predicted: No**	
Actual: Yes	TP = 170	FN = 3	173
Actual: No	FP = 173	TN = 404	577
	337	413	

From Equation 9.1, **RI** = (170+404)/750 = 0.765

From Equation 9.2, **P** = 170/(170+173) = 0.504

From Equation 9.3, **R** = 170/(170+3) = 0.983

From Equation 9.4, **F** = (2*0.504*0.983)/(0.504+0.983) = 0.666, where β = 1

Table 9.8 shows an ensemble of the evaluation results for different τ_{sim} values. From these results, it can be observed that the higher thresholds τ_{sim} (3.5 and 4.0) have higher recalls, but increase false positives (FP) (the number of dissimilar tweets that were grouped in the same cluster), therefore, precision goes down. In contrast, the lower

thresholds τ_{sim} (1.5 and 2.0) recorded higher precisions, but decrease false negatives (FN) (the number of similar tweets that were grouped in different clusters).

Table 9.8 Evaluation of the SBCA algorithm using different τ_{sim} values

τ	Precision	Recall	F-measure	Accuracy (RI)	Clusters (K)
1.5	97.3%	51%	66.9%	56.8%	6
2.0	97.4%	57.9%	72.6%	65.6%	15
2.5	94.4%	64.3%	76.5%	71.2%	37
3.0	91.3%	83.2%	87.1%	84.9%	52
3.5	73.1%	93.1%	81.9%	80.1%	84
4.0	50.4%	98.3%	66.6%	76.5%	131

The SBCA proximity measure (TREASURE) will be assigned the similarity threshold that provides a trade-off between precision (P) and recall (R). Since the F-measure is defined as the weighted harmonic mean of precision and recall, the threshold that demonstrates the highest F-measure is thus determined as the optimal parameter value for the SBCA algorithm. Table 9.8 shows an excellent performance (F-measure and accuracy) when τ_{sim} = 3.0. Considering the number of clusters, K, it can be observed that there is a linear relationship between τ_{sim} and the number of clusters, such that more clusters are generated as τ_{sim} increases and vice versa. Hence, a low value of τ_{sim} generates a coarse grained clusters, whereas higher values generate finer-grained clusters. Moreover, it can be observed that the number of clusters generated for τ_{sim} at 3.0 is the closest to the mean number of clusters, which is:

$$\mu(K) = (6+15+37+52+84+131)/6 = 54, \text{ which is} \approx 52.$$

The SBCA algorithm generating large number of clusters is attributed to two interrelated factors:

1. The STS.tweet_news dataset consists of 1500 tweets in the general domain of news, which contains tweets related to different events and topics.
2. TREASURE uses the Google News pre-trained word embedding model (described in Chapter 6), which may not contain specific words used in the STS.tweet_news dataset and thus tend to generate lower similarity values causing the SBCA algorithm to generate new clusters.

Experiment (1) provided results that demonstrate an optimal value of τ_{sim} at 3.0 for clustering microblogging posts utilising the STS.tweet_news similarity labelled benchmark. That is, the SBCA algorithm will assign tweets to the same cluster if and only if they share a similarity score > 3.0 ($S > \tau_{sim}$), according to TREASURE STSS

measure integrated in the SBCA algorithm. The next section describes the experiment carried out to detect semantic themes within the EU_Referendum dataset using the similarity threshold determined in experiment (1), which is τ_{sim} = 3.0, for the SBCA proximity measure.

9.3 Experiment 2: Detecting Semantic Themes within the EU Referendum Dataset

This section describes the experimental methodology and the detected semantic themes (i.e. generated clusters) in the EU Referendum dataset. Experiment (3) will provide a subjective evaluation of the generated clusters through running a human experiment to gather judgements on the belongingness of a subset of the results to their relevant clustroids.

The SBCA algorithm incorporating TREASURE as the proximity measure was implemented following the pseudocode presented in Chapter 8. SBCA follows a divisive approach such that all observations in the dataset start in one cluster. The cluster analysis commences by assigning a random observation, T_r, as a cluster center (i.e. clustroid). A recursive series of splits are subsequently performed based on comparing each observation with the derived clustroids. An observation, T_r, is assigned to an existing cluster if it satisfies a certain threshold, τ_{sim}, which is determined to be 3.0 (Experiment 1). Otherwise, a new cluster is generated and T_r is assigned as the new cluster's clustroid, T_c. This process recursively carries on until all observations in the dataset are assigned in clusters. Unlike most clustering algorithms that require the number of clusters to be determined beforehand, such as *k*-means, the SBCA algorithm does not apply this condition. Instead, the number of clusters in the dataset is dynamically determined according to the specified similarity threshold, τ_{sim}. This linear relationship implies that as the value of τ_{sim} increases, more clusters are generated and vice versa, as shown in Table 9.8, Section 9.2.2.

9.3.1 The EU Referendum Dataset Sampling Methodology

A cluster analysis of the entire EU Referendum dataset would be a complex and time consuming process (given the dataset size as discussed in Chapter 5 and algorithm complexity as discussed in Chapter 8). Therefore, a subset of the whole corpus of collected tweets is derived, such that the complete timeframe for the data collection process is spanned. Although it has been reported that 10% of a dataset is considered

a representative sample set (Severino, 2006), collecting a random 10% of the whole dataset may introduce bias in the resulting tweets and miss out on important events. Thus, the methodology for constructing a representative sample is conducted as follows:

1. The corpus of pre-processed tweets is divided into four groups according to the month a tweet has been streamed.
2. For each month during the data collection, the group of corresponding tweets is further split into four groups according to the week of tweet streaming.
3. The result is a corpus of tweets organized into four main groups corresponding to the four months of data collection and each group contains four subgroups according to the week a tweet has been streamed.
4. The representative subset is created by retrieving a random sample of 10% from each of the sixteen subgroups in order to span the entire data collection period.

This sampling methodology resulting in 13.7K tweets, not only ensures a representative subset is constructed in terms of size, but in content as well. The SBCA algorithm is applied on the sampled subset of tweets using TREASURE at the similarity threshold, τ_{sim} = 3.0. For clustering tweets on the EU_Referendum, TREASRE uses the corresponding EU_Referendum pre-trained word embedding model demonstrated in Chapter 6.

The eleven themes generated by the SBCA algorithm are shown in Figure 9.1 along with each theme cluster size.

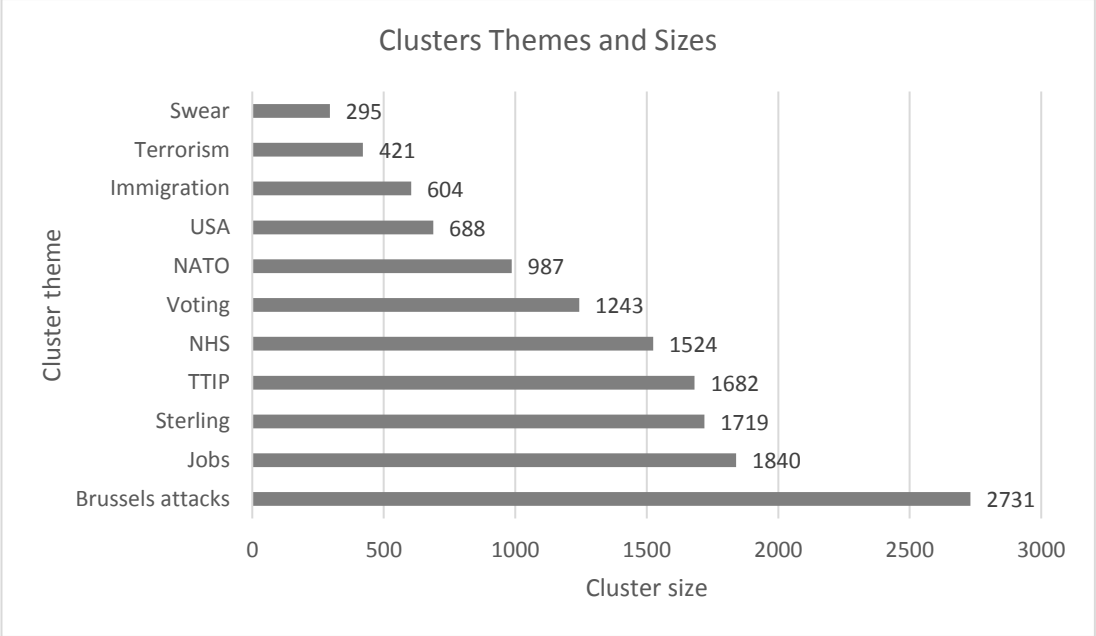


Figure 9.1 The EU Referendum themes detected by the SBCA algorithm

Table 9.9 shows the representative tweets (i.e. clustroid) for each of the eleven generated semantic clusters shown in Figure 9.1.

Table 9.9 The clustroids corresponding to the detected themes shown in Figure 9.1

Cluster id	**Representative tweet (clustroid)**
1	Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today
2	EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats
3	Sterling slides on renewed Brexit worries
4	Brexit Emerges As Threat to TTIP Deal
5	It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union
6	Should the United Kingdom remain a member of the EU or leave the EU?, Opinium poll: Remain: 49% (-3) Leave: 51% (+3)
7	Erdogan is an Islamic extremist who will flood the EU w #jihadists. Kick Turkey out of NATO and no admission to the EU. #Brexit
8	Both #HillaryClinton and #Obama continue to call on UK not to leave EU? If not EU #terror movement limited!
9	Brexit introduce controlled immigration system, deport those who support extremism
10	Terrorism is the scariest think. And it's ways more scarier if it's in the EU, in your home. Stay strong Brussels! #prayersforBrussels
11	It's just utterly stupid. Thank god UKIP will never get in power and Brexit will fucking fail.

The next section provides Experiment (3), which describes the subjective evaluation of the generated clusters through running an experiment with humans to gather classifications of random tweets from the sampled subset (described in Section 9.3.1)

to their relevant clustroids as shown in Table 9.9.

9.4 Experiment 3: Evaluating the SBCA Detected Themes through a Multi-Class Benchmark

This section describes the third experiment, which is divided into two stages. Firstly, a human experiment is conducted to generate a reliable multi-class labelled benchmark from the EU Referendum sampled tweets. Secondly, the generated clusters of semantic themes described in Experiment (2) are subjectively evaluated using the multi-class benchmark produced in the first stage.

9.4.1 Producing the EU_Referendum Multi-Class Benchmark

The experimental design and instruments used for collecting human classifications of tweets from the EU Referendum dataset is similar to the experiment conducted in Chapter 7 for gathering human similarity ratings. The majority of the gathered EU Referendum class annotations will be used as a benchmark for a subjective evaluation of the SBCA and an extrinsic evaluation of TREASURE. The human subjective judgements on mapping tweets to the most relevant class was gathered using a closed-ended questionnaire. These judgements form a subjective qualitative control that is used to assess the quality of the SBCA algorithm in detecting semantic themes within microblogging posts.

This section describes the methodology undertaken in constructing the following elements related to the human experiment:

1. The tweets and clustroids – includes obtaining random tweets from the SBCA generated clusters in which humans will be asked to assign them to their most appropriate category (through mapping a tweet to a clustroid).
2. The questionnaire design – includes the design of the task instructions such that less confusion is introduced to attain consistency between judges in order to produce a reliable benchmark.

9.4.1.1 Deriving Random Tweets from Clusters

In psychology, the capacity of information, i, that can be received, processed, and remembered in immediate memory of a typical human cognitive system is seven plus or minus two (Miller, 1956), that is $i \in r$, where $r = \{5, 6, 7, 8, 9\}$. The methodology of producing the benchmark of classification judgments on the SBCA generated

clusters from the EU Referendum subset is based on this psychological theory. In order to make the classification task as simple as possible for participants to complete, the experiment has been designed according to the results of the SBCA algorithm described in Section 9.3.1.

1. Each clustroid, *C*, which is essentially the clustroid corresponding to each of the five largest generated clusters (shown in Table 9.10) are used to form the categories, which has the themes, *Brussels attacks, Jobs, Sterling. TTIP, NHS.* Only these five clusters are used in the experiment in order to avoid complexity and keep it simple for the participants to follow according to the Miller (1956) psychological study.
2. For each *C*, three tweets are randomly selected to avoid bias and included in the experiment.
3. This subsampling process is performed for each representative tweet in the largest five generated clusters.
4. The resulting 15 tweets are used to form the EU_Referendum multi-class benchmark as shown in Table 9.11.

Table 9.10 Clustroids of the five largest tweets used in the experiment

Category	**Clustroids (*C*)**
A	Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today
B	EU Referendum Briefing on Living and Working in the EU #ProtectJobs #Expats
C	Sterling slides on renewed Brexit worries
D	#Brexit Emerges As Threat To TTIP[11] Deal
E	It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union

Table 9.11 Random tweets selected from the five largest clusters as shown in Table 9.10

Pair id	**Tweet (*T*)**	**Clustroids (*C*)**
1	I'm very sad for the families of the Brussels victims, but not at all surprised it happened! Wake up Europe #Brexit	*Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today*
2	On one hand, there are decent human beings that send their sympathies to the Brussels victims and their families. And then there's Brexit.	
3	#Brussels attacks: Terrorism could break the EU and lead to Brexit	
4	@caddenlimos connecting low paid workers doing non skilled jobs from Poland with	*EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats*

[11] Transatlantic Trade and Investment Partnership (TTIP)

	terrorism in Belgium	
5	Has anyone from the Brexit Brigade addressed the issue of what will happen to existing EU citizens living and working in the UK	
6	We must stay in #EU to protect jobs	
7	Sterling slides on renewed Brexit worries - A decline of 1% marks the 25th day this year the pound has moved	*Sterling slides on renewed Brexit worries*
8	Brussels bombing rose Brexit risk. How to trade Pound in case of Brexit: GBP $USD #FX	
9	Sterling has dipped cause markets believe Brexit will happen-GOOD-spivs in the city will adjust after playing their gambling games	
10	Jobs rely on trade NOT a political union. Security relies on sharing information NOT a political union. #Brexit #StrongerIn #VoteLeave	*#Brexit Emerges As Threat To TTIP Deal*
11	#Brexit, a new threat to TTIP transatlantic trade talks	
12	Naive to think Brexit would solve the #ttip problem...only way to protect #NHS is for govt to exclude it from TTIP	
13	UK's NHS will NOT survive staying in the EU	*It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union | via @Telegraph*
14	How can we save NHS inside EU	
15	I did worry about threat to NHS from TTIP - but EU and @EU_TTIP_team have listened to our concerns @HealthierIn	

This sampling methodology is performed to prevent any bias being introduced by selecting the tweets included in the experiment. The design of the questionnaire and population sampling follows the methodology provided in Section 7.3.2.3 and Section 7.3.3 in Chapter 7.

9.4.2 The Produced EU_Referendum Multi-Class Labelled Benchmark

The production of the EU_Referendum multi-class benchmark involved asking participants to complete a questionnaire, classifying tweets that are listed in a randomized order to their best matching clustroid from the provided list of clustroids (Table 9.10, Section 9.4.1.1). The participants were asked to complete the classification annotation questionnaire in their own time and to work through from start to end according to the given instructions as described in Section 9.4.1.2 (the classification annotation questionnaire is present in Appendix E, Section a). The 32 participants assigned each of the 15 tweets to their best matching cluster category from

Table 9.10 and the majority of the judgments obtained by the participants was determined as the actual class for each tweet. The resulting benchmark can be seen in Table 9.12, where all human classifications are provided as the major category score obtained for each tweet alongside the SBCA classifications.

Table 9.12 The EU_Referendum multi-class benchmark results

Id	Tweets	Human Classifications	SBCA
1	Has anyone from the Brexit Brigade addressed the issue of what will happen to existing EU citizens living and working in the UK	B	B
2	Sterling has dipped cause markets believe Brexit will happen-GOOD-spivs in the city will adjust after playing their gambling games	C	C
3	How can we save NHS inside EU	E	E
4	I'm very sad for the families of the Brussels victims, but not at all surprised it happened! Wake up Europe #Brexit	A	A
5	On one hand, there are decent human beings that send their sympathies to the Brussels victims and their families. And then there's Brexit.	A	A
6	#Brussels attacks: Terrorism could break the EU and lead to Brexit	A	A
7	Sterling slides on renewed Brexit worries - A decline of 1% marks the 25th day this year the pound has moved	C	C
8	@caddenlimos connecting low paid workers doing non skilled jobs from Poland with terrorism in Belgium	B	B
9	Brussels bombing rose Brexit risk. How to trade Pound in case of Brexit: GBP $USD #FX	A	C
10	I did worry about threat to NHS from TTIP - but EU and @EU_TTIP_team have listened to our concerns @HealthierIn	E	E
11	UK's NHS will NOT survive staying in the EU	E	E
12	#Brexit, a new threat to TTIP transatlantic trade talks	D	D
13	We must stay in #EU to protect jobs	B	B
14	Jobs rely on trade NOT a political union. Security relies on sharing information NOT a political union. #Brexit #StrongerIn #VoteLeave	B	D
15	Naive to think Brexit would solve the #ttip problem...only way to protect #NHS is for govt to exclude it from TTIP	D	D

As similarity interpretation is highly subjective, there are two cases where the SBCA algorithm failed to assign tweets instances to the clusters that the majority of human participants agreed upon, according to the multi-class benchmark shown in Table 9.12. For example, as tweet number 14 in Table 9.12 start with *jobs*, it gives an indication that it is related to the *jobs* cluster. However, the SBCA algorithm assigns this tweet to the *trade* cluster due to the high similarity it shares with the terms of the clustroid in the *trade* cluster.

The subsequent section provides an analysis of the multi-class benchmark production in terms of the reliability of the actual judgements that were gathered from the 32

participants and whether their judgements share a good level of agreement or not. The level of agreement among judges (humans) will determine the quality of the benchmark and the ability to use it for a subjective evaluation of the SBCA algorithm and other similar studies developed by the wider research community.

9.4.2.1 The Multi-Class Benchmark Reliability Analysis

The judgments obtained to produce the EU_Referendum multi-class benchmark were generated by 32 human observers instructed to classify 15 tweets to their best match clustroids. The average of classification judgments can only be trusted after demonstrating reliability. The agreement observed among independent observers is the key to reliability (Hayes and Krippendorff, 2007). As with the human similarity benchmark (reliability analysed in Chapter 7), the Krippendorff's alpha statistical test (Hayes and Krippendorff, 2007) (KALPHA) is used to assess the reliability of the EU_Referendum classification benchmark. That is, evaluating whether common instructions given to different observers of equivalent set of phenomena yields the same readings within a tolerable margin of error. As discussed in Chapter 7, Krippendorff's alpha, $\alpha = .80$ is generally brought forward as the norm for a good reliability test, with a minimum of .67 or even .60 (De Swert, 2012). Figure 9.2 shows the computed alpha result for the Krippendorff's test on the EU_Referendum classification benchmark.

```
Krippendorff's Alpha Reliability Estimate


           Alpha     LL95%CI     UL95%CI       Units    Observrs       Pairs
Nominal    .8222       .7845       .8570    15.0000     32.0000   7440.0000

Probability (q) of failure to achieve an alpha of at least alphamin:
   alphamin          q
      .9000     1.0000
      .8000      .1262                    KALPHA = .82
      .7000      .0000
      .6700      .0000
      .6000      .0000
      .5000      .0000

Number of bootstrap samples:
  10000

Judges used in these computations:
Columns   1 -  14
 P1        P2        P3        P4        P5        P6        P7        P8
 P9        P10       P11       P12       P13       P14
Columns  15 -  28
 P15       P16       P17       P18       P19       P20       P21       P22
 P23       P24       P25       P26       P27       P28
Columns  29 -  32
 P29       P30       P31       P32

Examine output for SPSS errors and do not interpret if any are found

------ END MATRIX -----
```

Figure 9.2 The Krippendorff's alpha test result for the EU Referendum classification benchmark

The Krippendorff's alpha test gives a good inter-rater agreement, at α = 0.82 for the production of the EU_Referendum classification benchmark presented in Section 9.4.2. Additionally, the bootstrapping procedure indicates that there is only 12.6% chance that the KALPHA would be below .80 if the whole population would be tested. Therefore, a subjective evaluation of the proposed SBCA algorithm can be conducted against the expert judgments with a relatively good confidence that the subjects are reliable enough to make conclusions towards the algorithm's performance.

9.4.3 Evaluating the SBCA Detected Themes using the EU_Referendum Multi-Class Benchmark

The EU_Referendum multi-class benchmark consists of tweets that are annotated with classes they belong to, which is used in this section to evaluate the SBCA algorithm. The application of the evaluation metrics discussed in the subsequent section for τ_{sim} = 3.0 as determined by Experiment (1) in Section 9.2, is undertaken to subjectively assess the SBCA generated clusters provided in Experiment (2), Section 9.3. The evaluation results will provide insights on the validity of the SBCA algorithm in

detecting semantic themes within microblogging posts and consequently answer the main research question outlined in Chapter 1 and its subsequent questions given in Section 9.1.

9.4.3.1 Rationale for the selection of the external evaluation criteria

Unlike the STS.tweet_news similarity-labelled benchmark, the EU_Referendum multi-class benchmark consists of classes from which each instance (i.e. tweet) belongs. Therefore, the evaluation of the SBCA generated clusters with reference to the EU_Referendum multi-class benchmark will be conducted using the *Purity* external evaluation measure in addition to the criteria described in Section 9.2.1.1.

To compute *purity*, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned tweets instances and dividing by *N*, which is the total number of clustered instances in the dataset. Purity can be formally defined as:

$$\text{Purity}(\Omega, \text{C}) = \frac{1}{N}\sum_{k} \max_{j} \left|k_i \cap c_j\right|$$

Equation 9.5 Purity (Schütze et al., 2008)

Where $\Omega = \{k_1, k_2, k_3 \ldots k_i\}$ is the set of clusters and $C = \{c_1, c_2, c_3 \ldots c_j\}$ is the set of classes. The k_i is interpreted as the set of tweets determined by the SBCA algorithm as belonging to k_i and c_j as the set of tweets determined in the EU_Referendum multi-class benchmark as belonging to c_j in Equation 9.5.

The five external evaluation criteria (*Purity*, *Rand index*, *Precision*, *Recall*, and *F-measure*) are computed to conduct an in-depth validation of the SBCA algorithm with reference to the EU_Referendum multi-class benchmark, where results are discussed in the subsequent section.

9.4.3.2 Evaluation Results and Discussion

This section presents the calculations that were performed for each of the evaluation criteria in order to obtain insights on the performance of the SBCA algorithm in detecting semantic themes embedded within the EU_Referendum rich domain of controversial views and discussions.

Purity calculates the degree of match between the instances in the clusters generated by the SBCA algorithm and in the EU_Referendum multi-class benchmark as demonstrated in Figure 9.3. In the case of a bad clustering, the purity values are close

to zero and a perfect clustering has a purity of one.

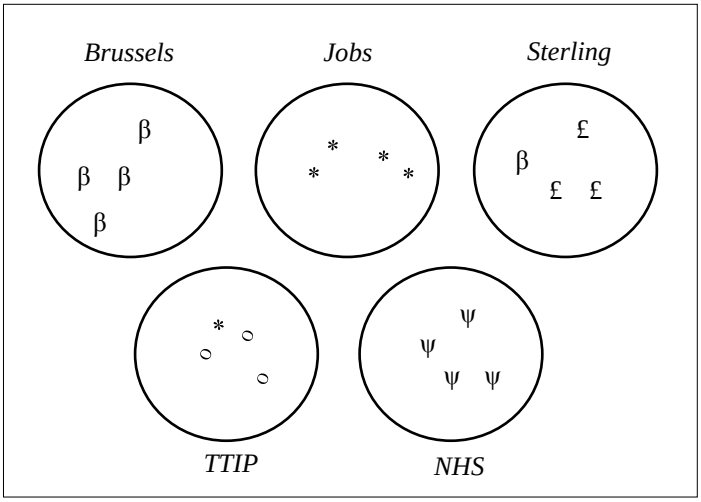


Figure 9.3 Demonstration of the Purity of the clusters generated by SBCA using the EU_Referendum multi-class benchmark shown in Table 9.12

From Figure 9.3, purity is calculated using Equation 9.5 by taking the majority of classes in each cluster such as:

$$\text{Purity(SBCA)} = (1/20)*(4+4+3+3+4) = 0.9$$

Where n = 20 is the total number of instances in each cluster. High purity is easy to achieve when the number of clusters is large. In particular, purity is 1 if each tweet gets its own cluster (i.e. singleton clusters). Thus, purity is not a standalone measure to trade off the quality of the clustering against the number of clusters. A measure that allows making this trade-off is the *Rand index.*

Rand index (*RI)* is a measure of the percentage of accurate decisions undertaken by the SBCA clustering algorithm using Equation 9.1. Table 9.13 demonstrates the matrix derived from the SBCA clusters and the EU_Referendum classes in order to compute the TP, TN, FP, and FN decisions.

Table 9.13 The matrix for computing the SBCA RI derived from Table 9.12

	C_1	C_2	C_3	C_4	C_5
Brussels	4		1		
Jobs		4		1	
Sterling			3		
TTIP				3	
NHS					4

From the matrix provided in Table 9.13, separation and combining decisions are computed and presented in the contingency matrix shown in Table 9.14.

Table 9.14 The contingency matrix for the SBCA and benchmark decisions

	Predicted: Yes	**Predicted: No**	
Actual: Yes	TP = 24	FN = 8	32
Actual: No	FP = 6	TN = 152	158
	30	160	

Thus, Random index, Precision, Recall, and the F-measure are calculated using the derived values of TP, TN, FP, and FN decisions and applying Equations 9.2, 9.3, and 9.4, respectively. The SBCA evaluation results using the five external evaluation criteria are provided in Table 9.15.

Table 9.15 Evaluation of the SBCA algorithm using the five external evaluation criteria

	Purity	**Precision**	**Recall**	**F-measure**	**Accuracy (RI)**
Lower bound	0.0	0.0	0.0	0.0	0.0
Upper bound	100%	100%	100%	100%	100%
SBCA value	90%	80%	75%	77.4%	92.6%

The discussion on the performance of the SBCA algorithm is conducted in terms of the external evaluation criteria as well as the clusters sizes. With regard to the *Purity*, the SBCA is considered to generate 90% pure clusters which is considered a very good level of purity (Vanegas and Bonet, 2018). The F-measure, based on a weighted harmonic mean of precision and recall, recorded 77.4% by the SBCA algorithm on the EU_Referendum dataset. However, because the F-measure does not take into account the true negatives (Mihalcea et al.), it is generally considered limited in capturing the full story (Xiong et al., 2004). Therefore, the accuracy (RI) is also computed in interpreting the results of the SBCA algorithm. The evaluation results demonstrated that the SBCA algorithm achieved an accuracy of 92.6%. Based on a similar study, which aimed to perform fuzzy clustering of health surveillance terms in social media (discussed in Chapter 3), achieved an accuracy of 87.1% (Dai et al., 2017) that was reported as excellent, SBCA is thus considered to achieve an excellent accuracy at 92.6% as demonstrated in Table 9.15. Compared to the SBCA performance on the STS.tweet_news dataset shown in Table 9.8, Section 9.2.2, the clustering algorithm achieved an 7.7% increase in terms of accuracy when applied on the EU_Referendum benchmark. This increase is anticipated to be attributed to the correlation of TREASURE on the EU_Referendum benchmark being higher than its correlation on the STS.tweet_news general domain dataset (discussed in Chapter 7), which was originally related to the different word embedding models used for each dataset

(described in Chapter 6) from which the semantic relationships between words are computed. In terms of the cluster sizes, a sharp decrease can be observed on the clusters generated from the EU Referendum dataset compared to the clusters generated from the STS.tweet_news dataset. The SBCA algorithm generated eleven clusters from the EU Referendum dataset and, at the same similarity threshold τ_{sim} = 3.0, generated 52 clusters from the STS.tweet_news dataset. This difference in the number of clusters is considered to be related to the following reasons:

1. As the STS.tweet_news dataset was aggregated for the purpose of semantic similarity of tweet pairs, it may not be a good candidate for cluster analysis. This is due to the too many general topics and different news and subjects contained within the 1500 instances. Moreover, there are only few tweets sharing similar meanings compared to the tweets in the EU Referendum dataset. On the other hand, the EU Referendum dataset is domain-specific which, due to the controversial views of users concerned with this political event, the dataset is considered to contain different themes that reflect the users' intentions behind their decisions to either leave or remain in the EU. These themes are apparent in the naturally occurring clusters generated by the SBCA algorithm, such as the NHS, drop in the British pound (cause and effect), trade deals with the USA, terrorist attacks, etc. Each of the generated clusters may have controversial views which encourages either the 'stronger in' campaign or the 'Brexit' campaign. Therefore, the EU Referendum dataset is considered a good candidate for cluster analysis as it provided insights on the intentions, argumentation mining, wider view of different communities that can be detected by posting similar tweets, and other use cases that demonstrate the usefulness of the SBCA algorithm in detecting semantic themes within microblogging posts.
2. A technical and important factor that is considered to have contributed in the difference in cluster sizes is related to the SBCA proximity measure (TREASURE). TREASURE incorporates a word embedding model from which it computes the semantic relationships between words. The pre-trained model used in Experiment (1) is different than the one used for Experiment (2). In the first experiment, TREASURE uses the Google News pre-trained model when applied on the STS.tweet_news dataset due to the considerations

discussed in Chapter 7. However, using a model trained on traditional text documents for the purpose of social networks linguistic analysis resulted in OOV words and missing terminology from the Google News pre-trained model. Thus, TREASURE tended to assign less similarity scores as a result of not recognising some of the words in a tweet (words that are not present in the pre-trained model). Consequently, new clusters are generated due to a similarity score that is less than the specified threshold causing a false negative by separating the two tweets being assessed for similarity (i.e. false separation decision). This is not the case for the EU Referendum dataset, where TREASURE uses the corresponding EU_Referendum word embedding model trained on the entire EU Referendum dataset (model training is described in Chapter 6). Therefore, TREASURE is not likely to encounter any OOV or terminology that is not recognized because the model was trained on the four million corpus of tweets collected on the EU Referendum domain (data collection and description is provided in Chapter5). Consequently, TREASURE tend to better capture the similarities between tweets (this claim is supported by the high correlation achieved by TREASURE on the EU_Referendum benchmark discussed in Chapter 7) and thus it is less likely to generate new clusters as a result of false negatives.

The external evaluation criteria for the SBCA algorithm provided adequate evidence to answer the two research questions outlined in Section 9.1, which are:

1. *Can the SBCA algorithm generate pure clusters from microblogging posts?*
   The high purity achieved by the SBCA algorithm on the challenging EU Referendum dataset shown in Table 9.15 based on a reliable multi-class benchmark (IRR test provided in Section 9.4.2.1), demonstrated that SBCA is able to generate pure clusters from Twitter posts, which is the most popular microblogging platform.
2. *Can the SBCA algorithm generate accurate clusters by undertaking correct separation and combining decisions with reference to a benchmark?*
   Accuracy is a measure that takes into consideration the correct and incorrect decisions undertaken by a machine learning algorithm. As the SBCA algorithm demonstrated a high accuracy as shown in Table 9.15 with reference to the reliable EU_Referendum multi-class benchmark, it can be concluded that the

SBCA can undertake accurate combining (TP) and separation (TN) decisions (Mihalcea et al.).

Thus, the main research question, "*Is it possible to automatically discover semantic themes in OSN microblogging posts based on an automated semantic computation method?*", can be answered with adequate evidence provided by the external evaluation criteria that the SBCA algorithm, based on TREASURE proximity measure, can automatically discover semantic themes in OSN microblogging posts.

9.5 Chapter Summary

This chapter has outlined and detailed the experimental methodology carried out to evaluate the new SBCA algorithm and illustrated the results of the external evaluation criteria in order to validate the development design proposed in Chapter 8 through conducting three experiments:

1. **Experiment (1)** aimed to figure out the optimal threshold value for the TREASURE proximity measure attribute. This experiment executed the SBCA algorithm on the STS.tweet_news dataset for different values of τ_{sim}. The evaluation results with reference to the STS.tweet_news similarity benchmark demonstrated that the threshold value of 3.0 provides the best clusters in terms of accuracy and F-measure (Section 9.2).
2. **Experiment (2)** was conducted to run the SBCA algorithm to detect semantic themes within the EU Referendum dataset using the similarity threshold, τ_{sim} = 3.0, derived from Experiment (1) (Section 9.3).
3. **Experiment (3)** is divided into two parts. In the first part, an experiment is conducted to gather human classifications of tweets subset from the EU Referendum dataset (Section 9.4). The second part uses the generated multi-class benchmark to evaluate the generated clusters by the SBCA algorithm from the EU Referendum dataset conducted in Experiment (2). The evaluation with reference to the multi-class benchmark was carried out using the external evaluation criteria designed in Section 9.4.3.1.

The performance of the SBCA algorithm was evaluated with reference the EU_Referendum multi-class benchmark in order to answer the following research questions:

1. *Can the SBCA algorithm generate pure clusters from microblogging posts?*

2. *Can the SBCA algorithm generate accurate clusters by undertaking correct separation and combining decisions with reference to a benchmark?*

The results from the experiments, using the external evaluation criteria with reference to the EU_Referendum multi-class benchmark, show adequate evidence to positively answer the research questions.

The main novel contributions in this chapter are:

- A new reliable benchmark of microblogging posts (tweets) assigned to their best match class, which is denoted by the clustroid of the corresponding cluster, labelled with class judgments by human experts with a good level of inter-rater agreement in the domain of Politics.
- A novel experimental methodology to produce a benchmark with human classifications derived from clusters, which are generated from a large dataset of raw microblogging posts.
- Evidence that the similarity threshold $\tau_{sim} = 3.0$, which corresponds to $\tau_{dis} = 0.4$ (applying Equations 8.1 and 8.2 respectively as in Chapter 8) provides the optimal value for the SBCA proximity measure generating the best set of clusters in terms of accuracy and F-measure compared to different threshold values.
- Evidence that the SBCA algorithm produces pure clusters from microblogging posts, particularly tweets.
- An evidence that the SBCA algorithm demonstrates a high level of accuracy in performing separation and combining decisions, which maximises true positives and true negatives.

Chapter 10 – Thesis Conclusions and Future Work

10.1 Overview

The research presented in this thesis aimed to answer two research questions:

1. Is it possible to intelligently measure the degree of semantic equivalence between OSN microblogging posts using an automated semantic computation method?
2. Is it possible to automatically discover semantic themes in OSN microblogging posts based on an automated semantic computation method?

Towards answering these questions, in the first phase of this thesis, a microblogging-based Short Text Semantic Similarity (STSS) measure, namely TREASURE (**T**weet simila**R**ity m**EASURE**), was researched, designed and developed. The second phase involved researching, designing, and developing a Semantic-Based Cluster Analysis (SBCA) algorithm aiming to detect semantic themes in microblogging posts. The SBCA algorithm incorporates TREASURE as the proximity measure from which tweets are assigned to clusters. The research involved investigation into several key areas such as, Natural Language Processing (NLP) for Semantic Textual Analysis (STA), Social Network Analysis (SNA), Language Modelling (LM), and Machine Learning (ML).

Undertaking this research required a large dataset of microblogging posts for evaluating the fundamental components of the proposed semantic-based framework, which are the TREASURE STSS measure and the SBCA algorithm. Therefore, a corpus of four million tweets was streamed using the twitter streaming Application Programming Interface (API) on the European Referendum political domain, which is considered a rich domain of controversial views. The raw tweets were pre-processed using a new heuristic-driven pre-processing methodology designed for the STSS measure (data collection and pre-processing are described in Chapter 5). Twitter Online Social Network (OSN) was the focus for this research as it is considered the most popular microblogging platform. Nevertheless, the new integrated components developed in this research could be extended to different microblogging platforms such as Tumbler[12] and Plurk[13].

[12] https://www.tumblr.com/
[13] https://www.plurk.com/portal/

The rest of the chapter is organized as follows: Sections 10.2 and 10.3 summarise the key components of the developed framework, which are TREASURE (development and evaluation were described in Chapters 6 and 7) and the SBCA algorithm (development and evaluation were described in Chapters 8 and 9) respectively. Section 10.4 lists the novel contributions of the research undertaken in this thesis. Finally, Section 10.5 discusses several considerations for future research.

10.2 The TREASURE STSS Measure

The proposed microblogging STSS measure (TREASURE) consists of two fundamental components that generate the overall similarity score for a given pair of tweets. The first is the semantic component, which is composed of semantic modules that handle the semantic computations based on deriving a semantic feature vector that represents each tweet. These are the *word analogy* and *weighting* modules. The *word analogy* module is accountable for computing the semantic relationships between words based on statistical word co-occurrence probabilities derived from a pre-trained word embedding model. In this model, each word is represented by a vector of real-valued numbers where each point captures a dimension of the word's meaning, such that semantically similar words have similar vectors. Two word embedding models were used in this research. The first, the Google News pre-trained model, was trained to learn word co-occurrences from traditional text documents. However, due to the limitations of this model in capturing social media terminology, a large proportion of out-of-vocabulary (OOV) and missing words was observed. Thus, a word embedding model was trained to learn distributed word representations from the entire corpus of EU Referendum tweets collected in this research. The *weighting* module assigns a weight to every word in a tweet, which demonstrates the word's significance in the overall meaning of a tweet based on its frequency of occurrence in a large text corpus. That is, frequently occurring words, such as function words (e.g. 'is', 'the', 'on', etc.) tend to have less information content compared to infrequently occurring words. The semantic component generates a semantic vector for each tweet that represents the semantic information contained within a tweet. The second fundamental component of TREASURE is the syntactic component, which consists of multiple syntactic modules that capture the morphological structure of words making up a tweet, as well as the textual conventions commonly used in Twitter. These are the *part-of-speech*

(Gómez-Adorno et al.) *tracking* and the *lexical analysis* modules. The *POS tracker* splits a tweet into tokens and analyses the context words in order to determine the POS of the word (e.g. *verb*, *noun*, *adjective*, or *adverb*). Whereas the lexical analyser analyses raw tweets and captures Twitter-based conventions (e.g. '#tags' and '@mentions') contained within tweets, as well as other expressive punctuations such as interrogation and exclamation marks. The output of the syntactic component is a representation of each tweet by a syntactic vector. Based on empirical experiments (described in Chapter 7), the overall similarity score produced by TREASURE is a combination of the semantic and syntactic similarities, with the semantic weighted 0.8, whereas the syntactic weighted 0.2.

The intrinsic evaluation of TREASURE involved undertaking an experiment to gather human similarity ratings on tweet pairs sampled from the EU_Referendum dataset to produce a reliable similarity benchmark. This benchmark was used to evaluate the linear association between TREASURE and the mean of human ratings. Furthermore, the generalizability of TREASURE was evaluated using a general-domain benchmark, which is the STS.tweet_news published for SemEval-2014 semantic similarity shared task. TREASURE achieved a mean correlation coefficient of $r = 0.8$, significant at (p-value < 0.01) and recorded the highest correlation among existing semantic similarity measures. Using inferential statistical analysis, the experiment results provided adequate evidence to test the hypotheses and concludes that TREASURE is a high-correlation STSS measure for microblogging posts that can be generalizable to different domains.

10.3 The SBCA Algorithm

The SBCA is a new partition-based hard clustering algorithm that generates non-overlapping clusters. Unlike other partitioning algorithms that require the number of clusters to be determined beforehand (such as k-means), SBCA is a fully unsupervised algorithm designed to detect semantic themes within microblogging posts without requiring the number of clusters to be predetermined. The SBCA algorithm incorporates TREASURE as the proximity measure such that tweets are assigned into clusters if and only if TREASURE determined that the similarity between a tweet and a clustroid is greater than a certain threshold, τ_{sim}. In order to determine the optimal parameter value, an empirical experiment was conducted with different values of τ_{sim},

and for each value, the SBCA generated clusters were evaluated and the threshold value that resulted in the clusters set with the highest accuracy was determined to be the optimal value of τ_{sim}. The empirical experimental results (described in Chapter 9) demonstrated that τ_{sim} = 3.0 generates the most accurate clusters and thus, it was determined to be the optimal value for τ_{sim}. The SBCA algorithm assigns the tweet that minimises the sum of TREASURE distances to other instances in the same cluster to be the representative of that cluster (i.e. clustroid). SBCA has an average time complexity O(*I**K*f*n*), where *K* is the number of clusters, *f* is the number of features (described in Chapter 5), *n* is the number of instances in the dataset, and *I* is the number of iterations required to update the sum of pairwise distances in each cluster. SBCA runs in less time than hierarchical approaches, which has a complexity O(n^3) for agglomerative and O(2^n) for divisive algorithms, which means that SBCA algorithm scales better for larger datasets of microblogging posts.

The SBCA algorithm was used to detect semantic themes within the EU Referendum dataset. The SBCA generated eleven themes using the threshold predetermined by the empirical experiment, which is τ_{sim} = 3.0. Towards evaluating the clusters generated by the SBCA, an experiment was conducted to gather humans classifications of EU Referendum tweets to their best match cluster in order to produce a multi-class evaluation benchmark. Subjective evaluation criteria were applied with reference to the produced EU_Referendum multi-class benchmark in order to evaluate the clusters generated by the SBCA algorithm. The evaluation results demonstrated that the SBCA algorithm has a high level of accuracy in performing the separation and combining decisions (i.e. maximising true positives and true negatives) and thus can generate pure clusters from microblogging posts.

Based on the results observed from the experimental evaluations, the evidence supports the conclusion that TREASURE can intelligently (semantically in a technical term) measure the degree of equivalence between OSN microblogging posts. In addition, the SBCA algorithm can automatically discover semantic themes within OSN microblogging posts based on an automated semantic computation method, which is TREASURE. Further work in the field of computational linguistics in OSN and ML can build on top of this work which is discussed in Section 10.5.

10.4 Research Contributions

This research has produced some significant contributions in the field of NLP for microblogging OSN. The primary aim of this research was to design and develop an integrated semantic-based framework for microblogging cluster analysis (SBCA) that detects semantic themes within microblogging posts through incorporating a novel STSS measure, which was named TREASURE. TREASURE employs word embedding models to derive hybrid semantic and syntactic features from a pair of tweets and assign an overall similarity score, which is a weighted combination of semantic and syntactic similarities. The SBCA algorithm incorporates TREASURE as the proximity measure to assign tweets to clusters according to a certain threshold that was determined using empirical experiments. The outcome of this research project is the development of a semantic integrated framework of a microblogging cluster analysis and a novel STSS measure that captures the semantic similarities between microblogging posts. TREASURE, although embedded within the SBCA algorithm, was developed in such a way that it can be used independently and adapted by the wider research community for applications related to semantic similarity computations for different microblogs. Similarly, the SBCA algorithm can incorporate different proximity measures, which can be a similarity or a distance based measure depending on the context for which it is applied.

The prominent contributions derived from this research are as follows:

10.4.1 A Heuristic-driven Pre-processing Methodology for Microblogging STSS

The research into microblogging textual challenges and existing pre-processing methodologies and computational linguistics has led to the development of a pre-processing methodology consisting of heuristic rules. This pre-processing methodology takes into account the special lexical characteristics of microblogging posts in order to transform raw tweets into a less noisy form, while preserving important features for STSS measures, such as OOV and hashtags. These heuristic rules have been evaluated and published for the benefit of the wider NLP research community (Chapter 5).

10.4.2 A Method for Developing TREASURE Hybrid Components

The research has led to the development of a novel STSS architectural design based on hybrid semantic and syntactic components, known as TREASURE. This new STSS

measure is composed of integrated modules that analyses the morphological structure of the words contained in a tweet and combines it with the semantic relationships between these words based on statistical analysis of their co-occurrences in a large text corpus. A proof of concept has been conducted using the EU Referendum political domain in Twitter. Nevertheless, evidence has been obtained through inferential statistical analysis that TREASURE can be generalized to other different domains. TREASURE can also be extended to different microblogging platforms (Chapter 6).

10.4.3 A Method for Training a Word Embedding Model from Microblogs

The research and experiments, conducted within this thesis, considering different language models and existing pre-trained word embedding models has imposed the necessity for a word embedding model trained on microblogging posts. This is due to words being used in a different manner in the context of social media than their usage in traditional text documents, which implies that their corresponding co-occurrence vectors is different. Therefore, a new word embedding model was trained to learn distributed word representations from a large corpus of microblogging posts, which was the four million tweets collected on the EU Referendum. The result is a pre-trained word embedding model that can be used for OSN-based NLP applications in the domain of politics (Chapter 6).

10.4.4 A Method for Experimentally Producing a Similarity Benchmark

The development of TREASURE has led the research to investigate existing similarity benchmarks and different methodologies for conducting a human-involved experiment to gather similarity ratings for the purpose of STSS intrinsic evaluation. An unsupervised methodology was undertaken in order to derive tweet pairs without introducing bias. The experimental methodology involved an adaptation to the semantic anchors in the Likert scale and carefully designed instructions and guidelines in order to eliminate confusion for participants and aim for a good level of inter-rater agreement (Chapter 7).

10.4.5 A Reliable Similarity Benchmark for STSS Intrinsic Evaluation

The intrinsic evaluation of TREASURE has led to the production of a reliable benchmark of human similarity ratings for tweet pairs on the EU Referendum political domain. The generated EU_Referendum similarity benchmark consists of 30 tweet pairs, each annotated with the mean of 32 human ratings sharing a good level of

agreement. This benchmark shall fill the gap of the lack of exciting microblogging-based reliable benchmark that can be utilised for different STA applications in the domain of politics (Chapter 7).

10.4.6 A Method for Developing the SBCA Algorithm

A new SBCA algorithm was designed and developed to detect semantic themes within microblogging posts. Unlike existing partition-based cluster analysis approaches, this algorithm is *fully unsupervised* and does not require the number of clusters to be pre-determined. The SBCA algorithm incorporates TREASURE as the proximity measure to generate non-overlapping clusters. Unlike clustering algorithms where instances are modelled in a Euclidean space and the centroid represents the actual centre of gravity for a cluster, the SBCA algorithm deals with unstructured textual instances. Modelling these instances using a vector space model will generate very sparse vectors and will consequently cause computational complexity and scalability issues. Thus, TREASURE is used assign tweets into clusters if and only if a tweet and a cluster centre are within a certain distance constraint with respect to a certain threshold. The SBCA algorithm was developed such that it integrates the best properties of both the partition-based and hierarchical clustering approaches. These properties are reasonable runtime complexity and the dynamic production of the number of clusters, respectively (Chapter 8).

10.4.7 A Method for Experimentally Producing a Multi-Class Benchmark

The development of a semantic based clustering algorithm required a multi-class benchmark in order to employ external evaluation criteria. A new experimental methodology was devised in order to construct a non-biased sample subset from the SBCA generated clusters. This sample was derived taking into consideration the psychology of the maximum human cognitive capacity of information processing at a single time in order to maximise the accuracy of the responses. Using a reliability statistical test, this methodology has resulted in generating a multi-class benchmark with a high level of inter-rater agreement (Chapter 9).

10.4.8 A Reliable Multi-Class Benchmark for Subjective Evaluation

The subjective evaluation of the SBCA algorithm has led to the production of a reliable benchmark of human multi-class judgments for the belongingness of tweets to

clustroids on the EU Referendum political domain. The generated EU_Referendum multi-class benchmark consists of fifteen tweet and five clustroids, each annotated with the best match clustroid. These annotations were obtained by computing the majority class of 32 human judgments sharing a good level of agreement. This benchmark shall fill the gap of the lack of existing microblogging-based reliable benchmark that can be utilised for different clustering and classification machine learning applications in the domain of politics (Chapter 9).

10.4.9 An Integrated Semantic Framework for Microblogging Cluster Analysis

The product of this research project is a semantic-based framework of integrated hybrid components developed for the aim of detecting semantic themes within microblogging posts, which is useful for different task as people are shifting from traditional media to OSN. This framework can be used collectively to generate natural semantic clusters, which has potential in the digital era of big data where the manual detection of meaningful clusters within millions of user generated records is a labour and time intensive, if not impossible, task. Thus, this research was conducted in order to automate this process and intelligently discover semantic themes in both batch and real-time modes. Nevertheless, each of the semantic-based components in the developed framework can be used independently for different research and practice objectives for various NLP and computational intelligence applications such as embedding TREASURE within a Conversational Agent.

10.5 Future Work

The research presented in this thesis has outlined a novel approach to detecting semantic themes in microblogging posts through incorporating a new STSS measure that predicts the semantic similarity between microblogging posts based on integrating semantic and syntactic components. The research at this stage, meets is aims and objectives and addresses the main research questions. However, there is room for improvement for both TREASURE and the SBCA algorithm, which can be further investigated through future research and development. Some of these suggestions are discussed in subsequent sections.

10.5.1 *280*-Character Tweet Implications

The data collection, pre-processing and feature extraction steps undertaken in this

research has taken into consideration the tweets challenges as a consequence of the 140-character limit restriction. Twitter has recently expanded this restriction to 280 characters instead of 140, which provided users for more room to express and share their thoughts. To the best of the researcher's knowledge, there is no existing research in the field of Twitter textual analytics that has investigated the effect of such increase on the textual features of tweets and its implications on NLP applications. Further research consider this expansion to assure that the semantic-based framework and its hybrid components are optimised accordingly.

10.5.2 Language Model Expansion

This research has created and trained a word embedding model on the European Referendum political domain. As the accuracy of a word embedding model is highly dependent of the size of the training corpus, data collection will carry on in order to expand the EU_Referendum pre-trained model. The expansion will include further positive examples (i.e. meaningful sentences) from political as well as other domains in order to create a larger and more generalized word embedding model. The expanded model shall provide an important lexical resource for the wider research community in the field OSN analysis.

10.5.3 Investigating Tweet assignment to Fuzzy Clusters

The SBCA implements a crisp categorization algorithm that generates non-overlapping clusters, in which a tweet belongs to one and only one cluster. Nevertheless, adding a further fuzzy layer on top of the SBCA algorithm that assigns microblogging post to different clusters with a varying degrees of belongingness shall add flexibility and provide a broader and in-depth knowledge into the fuzzy tweets and themes within a microblogging dataset (Rathore et al., 2018).

10.5.4 Multi-Lingual TRSEAURE

TREASURE can be adapted to different languages through investigating lexical resources and word embedding models that could be integrated from other languages. Following the general data collection, pre-processing, and word embedding training methodologies designed and implemented in this thesis, a word embedding model can be trained to learn from a corpus of microblogging posts in various languages. A multi-lingual TREASURE would have potential implications on NLP applications that involve translations. Furthermore, multi-lingual TREASURE shall provide wider

insights on the controversial views and arguments of microblogging users with different cultural backgrounds speaking different languages.

References

ABBASI, M.-A. & LIU, H. Measuring user credibility in social media. International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction, 2013. Springer, 441-448.

ADEDOYIN-OLOWE, M., GABER, M. M. & STAHL, F. 2013. A survey of data mining techniques for social media analysis. *arXiv preprint arXiv:1312.4617*.

AGGARWAL, C. C. & REDDY, C. K. 2013. *Data clustering: algorithms and applications*, CRC press.

AGGARWAL, C. C. & YU, P. S. 2000. *Finding generalized projected clusters in high dimensional spaces*, ACM.

AGGARWAL, C. C. & ZHAI, C. 2012. *Mining text data*, Springer Science & Business Media.

AGGARWAL, N., ASOOJA, K. & BUITELAAR, P. DERI&UPM: Pushing corpus based relatedness to similarity: Shared task system description. Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, 2012. Association for Computational Linguistics, 643-647.

AGIRRE, E., BANEA, C., CER, D., DIAB, M., GONZALEZ-AGIRRE, A., MIHALCEA, R., RIGAU, G. & WIEBE, J. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016a. 497-511.

AGIRRE, E., BANEA, C., CER, D. M., DIAB, M. T., GONZALEZ-AGIRRE, A., MIHALCEA, R., RIGAU, G. & WIEBE, J. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. SemEval@ NAACL-HLT, 2016b. 497-511.

AGIRRE, E., CER, D., DIAB, M., GONZALEZ-AGIRRE, A. & GUO, W. * SEM 2013 shared task: Semantic textual similarity. Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, 2013. 32-43.

AGIRRE, E., DIAB, M., CER, D. & GONZALEZ-AGIRRE, A. Semeval-2012 task 6: A pilot on semantic textual similarity. Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, 2012. Association for Computational Linguistics, 385-393.

AHUJA, S. & DUBEY, G. Clustering and sentiment analysis on Twitter data. 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), 2017. IEEE, 1-5.

AKKAYA, C., WIEBE, J. & MIHALCEA, R. Subjectivity word sense disambiguation. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1, 2009. Association for Computational Linguistics, 190-199.

ALBITAR, S., FOURNIER, S. & ESPINASSE, B. An effective TF/IDF-based text-to-text semantic similarity measure for text classification. International

Conference on Web Information Systems Engineering, 2014. Springer, 105-114.

ALLAN, J., WADE, C. & BOLIVAR, A. Retrieval and novelty detection at the sentence level. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003. ACM, 314-321.

ALLEN, I. E. & SEAMAN, C. A. 2007. Likert scales and data analyses. *Quality progress,* 40**,** 64-65.

ALNAJRAN, N., CROCKETT, K., MCLEAN, D. & LATHAM, A. Cluster Analysis of Twitter Data: A Review of Algorithms. Proceedings of the 9th International Conference on Agents and Artificial Intelligence, 2017. Science and Technology Publications (SCITEPRESS)/Springer Books, 239-249.

ALNAJRAN, N., CROCKETT, K., MCLEAN, D. & LATHAM, A. 2018a. An Empirical Performance Evaluation of Semantic-Based Similarity Measures in Microblogging Social Media. *In Proceedings of the Fifth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT 2018).* Zurich: IEEE/ACM.

ALNAJRAN, N., CROCKETT, K., MCLEAN, D. & LATHAM, A. A Heuristic Based Pre-processing Methodology for Short Text Similarity Measures in Microblogs. High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2018 IEEE 20th International Conference on, 2018b. IEEE.

ALNAJRAN, N., CROCKETT, K., MCLEAN, D. & LATHAM, A. 2018c. A Word Embedding Model Learned from Political Tweets. *In Computer Engineering & Systems (ICCES), 2018 13th International Conference on.* IEEE.

ANGIANI, G., FERRARI, L., FONTANINI, T., FORNACCIARI, P., IOTTI, E., MAGLIANI, F. & MANICARDI, S. A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter. KDWeb, 2016.

ANTENUCCI, D., HANDY, G., MODI, A. & TINKERHESS, M. 2011. Classification of tweets via clustering of hashtags. *EECS,* 545**,** 1-11.

ANUMOL BABU, R. V. P. 2016. Efficient Density Based Clustering of Tweets and Sentimental Analysis Based on Segmentation. *International Journal of Computer Techniques,* 3**,** 53-57.

ATOUM, I., OTOOM, A. & KULATHURAMAIYER, N. 2016. A comprehensive comparative study of word and sentence similarity measures. *arXiv preprint arXiv:1610.04533*.

BANKER, K. 2011. *MongoDB in action*, Manning Publications Co.

BÄR, D., BIEMANN, C., GUREVYCH, I. & ZESCH, T. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, 2012. Association for Computational Linguistics, 435-440.

BARALIS, E., CERQUITELLI, T., CHIUSANO, S., GRIMAUDO, L. & XIAO, X. Analysis of twitter data using a multiple-level clustering strategy. International Conference on Model and Data Engineering, 2013. Springer, 13-24.

BARRY, C., MEADOW, C. T., KRAFT, D. H. & BOYCE, B. R. 2007. *Text Information Retrieval Systems*, Academic Press.

BATES, A. 2015. *Using Term Statistics to Aid in Clustering Twitter Posts.* University of Colorado Colorado Springs.

BEAM, A. L., KOMPA, B., FRIED, I., PALMER, N. P., SHI, X., CAI, T. & KOHANE, I. S. 2018. Clinical Concept Embeddings Learned from Massive Sources of Medical Data. *arXiv preprint arXiv:1804.01486*.

BEZDEK, J. C., EHRLICH, R. & FULL, W. 1984. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences,* 10**,** 191-203.

BIRD, J. 2014. *Basic engineering mathematics,* Routledge.

BLAXTER, L., HUGHES, C. & TIGHT, M. 2006. How to Research 3rd. Open University Press.

BLEI, D. M., NG, A. Y. & JORDAN, M. I. 2003. Latent dirichlet allocation. *Journal of machine Learning research,* 3**,** 993-1022.

BOFFA, S., DE MAIO, C., GERLA, B. & PARENTE, M. Context-aware Advertisment Recommendation on Twitter through Rough sets. 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2018. IEEE, 1-8.

BOICEA, A., RADULESCU, F. & AGAPIN, L. I. MongoDB vs Oracle--database comparison. 2012 third international conference on emerging intelligent data and web technologies, 2012. IEEE, 330-335.

BOJANOWSKI, P., GRAVE, E., JOULIN, A. & MIKOLOV, T. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

BORA, D. J., GUPTA, D. & KUMAR, A. 2014. A comparative study between fuzzy clustering algorithm and hard clustering algorithm. *arXiv preprint arXiv:1404.6059*.

BORIAH, S., CHANDOLA, V. & KUMAR, V. 2008. Similarity measures for categorical data: A comparative evaluation. *red,* 30**,** 3.

BOTSCH, M., STEINBERG, S., BISCHOFF, S. & KOBBELT, L. 2002. Openmesh-a generic and efficient polygon mesh data structure.

BREIMAN, L. 2001. Random forests. *Machine learning,* 45**,** 5-32.

BROWN, P. F., DESOUZA, P. V., MERCER, R. L., PIETRA, V. J. D. & LAI, J. C. 1992. Class-based n-gram models of natural language. *Computational linguistics,* 18**,** 467-479.

BUHRMESTER, M., KWANG, T. & GOSLING, S. D. 2011. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science,* 6**,** 3-5.

BURGESS, C., LIVESAY, K. & LUND, K. 1998. Explorations in context space: Words, sentences, discourse. *Discourse Processes,* 25**,** 211-257.

CASTILLO, C., MENDOZA, M. & POBLETE, B. Information credibility on twitter. Proceedings of the 20th international conference on World wide web, 2011. ACM, 675-684.

CHARLES, W. G. 2000. Contextual correlates of meaning. *Applied Psycholinguistics,* 21**,** 505-524.

CHEN, X., LI, L., XU, G., YANG, Z. & KITSUREGAWA, M. Recommending Related Microblogs: A Comparison Between Topic and WordNet based Approaches. AAAI, 2012.

CHRISTOPH, L. 2016. Measuring Semantic Similarity and Relatedness with Distributional and Knowledge-based Approaches.

CISZAK, L. Application of clustering and association methods in data cleaning. Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on, 2008. IEEE, 97-103.

COHEN, J. 1988. Statistical power analysis for the behavioral sciences 2nd edn. Erlbaum Associates, Hillsdale.

COLLOBERT, R. & WESTON, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. Proceedings of the 25th international conference on Machine learning, 2008. ACM, 160-167.

CONSTANTINE, L. L. & LOCKWOOD, L. A. 1999. *Software for use: a practical guide to the models and methods of usage-centered design*, Pearson Education.

CORDOBÉS, H., ANTA, A. F., CHIROQUE, L. F., GARCÍA, F. P., REDONDO, T. & SANTOS, A. 2014. Graph-based Techniques for Topic Classification of Tweets in Spanish. *IJIMAI*, 2**,** 32-38.

CRESWELL, J. W. & CRESWELL, J. D. 2017. *Research design: Qualitative, quantitative, and mixed methods approaches*, Sage publications.

CROSSNO, P. J., WILSON, A. T., SHEAD, T. M. & DUNLAVY, D. M. Topicview: Visually comparing topic models of text collections. Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on, 2011. IEEE, 936-943.

DAI, X., BIKDASH, M. & MEYER, B. From social media to public health surveillance: Word embedding based clustering method for twitter classification. SoutheastCon, 2017, 2017. IEEE, 1-7.

DAS, D. & BANDYOPADHYAY, S. Developing Bengali WordNet affect for analyzing emotion. International Conference on the Computer Processing of Oriental Languages, 2010. 35-40.

DAS, D. & SMITH, N. A. Paraphrase identification as probabilistic quasi-synchronous recognition. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1, 2009. Association for Computational Linguistics, 468-476.

DAVIDSON, S. 2013. Wordnik. *The Charleston Advisor*, 15**,** 54-58.

DE BOOM, C., VAN CANNEYT, S., BOHEZ, S., DEMEESTER, T. & DHOEDT, B. Learning semantic similarity for very short texts. Data Mining Workshop (ICDMW), 2015 IEEE International Conference on, 2015a. IEEE, 1229-1234.

DE BOOM, C., VAN CANNEYT, S. & DHOEDT, B. Semantics-driven event clustering in twitter feeds. Making Sense of Microposts, 2015b. CEUR, 2-9.

DE SWERT, K. 2012. Calculating inter-coder reliability in media content analysis using Krippendorff's Alpha. *Center for Politics and Communication***,** 1-15.

DEERWESTER, S., DUMAIS, S. T., FURNAS, G. W., LANDAUER, T. K. & HARSHMAN, R. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41**,** 391-407.

DEMIRSOZ, O. & OZCAN, R. 2016. Classification of news-related tweets. *Journal of Information Science***,** 0165551516653082.

DENNIS, S., LANDAUER, T., KINTSCH, W. & QUESADA, J. Introduction to latent semantic analysis. Slides from the tutorial given at the 25th Annual Meeting of the Cognitive Science Society, Boston, 2003.

DERCZYNSKI, L., BONTCHEVA, K., LIAKATA, M., PROCTER, R., HOI, G. W. S. & ZUBIAGA, A. 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. *arXiv preprint arXiv:1704.05972*.

DEY, K., SHRIVASTAVA, R., KAUSHIK, S. & SUBRAMANIAM, L. V. 2017. Emtagger: a word embedding based novel method for hashtag recommendation on twitter. *arXiv preprint arXiv:1712.01562*.

DOLNICAR, S. 2002. A review of unquestioned standards in using cluster analysis for data-driven market segmentation.

DUONG, P. H., NGUYEN, H. T. & HUYNH, N.-T. Measuring Similarity for Short Texts on Social Media. International Conference on Computational Social Networks, 2016. Springer, 249-259.

DURLAK, J. A. 2009. How to select, calculate, and interpret effect sizes. *Journal of pediatric psychology,* 34**,** 917-928.

DUTTA, S., GHATAK, S., ROY, M., GHOSH, S. & DAS, A. K. A graph based clustering technique for tweet summarization. Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2015 4th International Conference on, 2015. IEEE, 1-6.

ELIO, R., HOOVER, J., NIKOLAIDIS, I., SALAVATIPOUR, M., STEWART, L. & WONG, K. 2011. About computing science research methodology.

ERKAN, G. & RADEV, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research,* 22**,** 457-479.

ESTER, M., KRIEGEL, H.-P., SANDER, J. & XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd, 1996. 226-231.

FANG, A., MACDONALD, C., OUNIS, I. & HABEL, P. Topics in tweets: A user study of topic coherence metrics for Twitter data. European Conference on Information Retrieval, 2016. Springer, 492-504.

FARZINDAR, A. & INKPEN, D. 2017. Natural language processing for social media. *Synthesis Lectures on Human Language Technologies,* 10**,** 1-195.

FAUL, F., ERDFELDER, E., LANG, A.-G. & BUCHNER, A. 2007. G* Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior research methods,* 39**,** 175-191.

FELLBAUM, C. 1998. *WordNet*, Wiley Online Library.

FIELD, A. 2012. Exploring data: The beast of bias. *Discovering Statistics*.

FÓCIL-ARIAS, C., ZÚÑIGA, J., SIDOROV, G., BATYRSHIN, I. & GELBUKH, A. A tweets classifier based on cosine similarity.

FOLTZ, P. W., KINTSCH, W. & LANDAUER, T. K. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse processes,* 25**,** 285-307.

FORMANN, A. K. 1984. *Die latent-class-analyse: Einführung in Theorie und Anwendung*, Beltz.

FORNEY, G. D. 1973. The viterbi algorithm. *Proceedings of the IEEE,* 61**,** 268-278.

FRANCIS, W. N. & KUCERA, H. 1964. Brown corpus. *Department of Linguistics, Brown University, Providence, Rhode Island,* 1.

FRIEDEMANN, V. 2015. Clustering a Customer Base Using Twitter Data.

GANITKEVITCH, J., VAN DURME, B. & CALLISON-BURCH, C. PPDB: The paraphrase database. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013. 758-764.

GARG, N. & RANI, R. Analysis and visualization of Twitter data using k-means clustering. Intelligent Computing and Control Systems (ICICCS), 2017 International Conference on, 2017. IEEE, 670-675.

GHASEMI, A. & ZAHEDIASL, S. 2012. Normality tests for statistical analysis: a guide for non-statisticians. *International journal of endocrinology and metabolism,* 10**,** 486.

GO, A., BHAYANI, R. & HUANG, L. 2009a. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford,* 1**,** 12.

GO, A., BHAYANI, R. & HUANG, L. 2009b. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford,* 1.

GODFREY, D., JOHNS, C., MEYER, C., RACE, S. & SADEK, C. 2014. A case study in text mining: Interpreting twitter data from world cup tweets. *arXiv preprint arXiv:1408.5427*.

GÓMEZ-ADORNO, H., MARKOV, I., SIDOROV, G., POSADAS-DURÁN, J.-P., SANCHEZ-PEREZ, M. A. & CHANONA-HERNANDEZ, L. 2016. Improving feature representation based on a neural network for author profiling in social media texts. *Computational intelligence and neuroscience,* 2016**,** 2.

GRAVETTER, F. J. & WALLNAU, L. B. 2016. *Statistics for the behavioral sciences*, Cengage Learning.

GUNDECHA, P. & LIU, H. 2012. Mining social media: a brief introduction. *New Directions in Informatics, Optimization, Logistics, and Production.* Informs.

GUO, W. & DIAB, M. A simple unsupervised latent semantics based approach for sentence similarity. Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, 2012. Association for Computational Linguistics, 586-590.

GUO, W., LI, H., JI, H. & DIAB, M. Linking tweets to news: A framework to enrich short text data in social media. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2013. 239-249.

GWET, K. L. 2014. *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*, Advanced Analytics, LLC.

HAJJEM, M. & LATIRI, C. 2016. Features extraction to improve comparable tweet corpora building. *JADT Acte, Nice, France*.

HAN, J., PEI, J. & KAMBER, M. 2011. *Data mining: concepts and techniques*, Elsevier.

HARRIS, Z. S. 1968. Mathematical structures of language.

HAYES, A. F. 2009. *Statistical methods for communication science*, Routledge.

HAYES, A. F. & KRIPPENDORFF, K. 2007. Answering the call for a standard reliability measure for coding data. *Communication methods and measures,* 1**,** 77-89.

HEREDIA, B., PRUSA, J. D. & KHOSHGOFTAAR, T. M. Location-Based Twitter Sentiment Analysis for Predicting the US 2016 Presidential Election. The Thirty-First International Flairs Conference, 2018.

HOLSTEIN, J. 1994. Phenomenology, Ethonomethodology, and Interpretive Practice. *Handbook of qualitative research***,** 105-117.

HONG, L. & DAVISON, B. D. Empirical study of topic modeling in twitter. Proceedings of the first workshop on social media analytics, 2010. ACM, 80-88.

HOTHO, A., MAEDCHE, A. & STAAB, S. 2002. Ontology-based text document clustering. *KI,* 16**,** 48-54.

HUANG, Y. & MITCHELL, T. M. Text clustering with extended user feedback. Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, 2006. ACM, 413-420.

IFRIM, G., SHI, B. & BRIGADIR, I. Event detection in twitter using aggressive filtering and hierarchical tweet clustering. Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014, 2014. ACM.

INOUYE, D. & KALITA, J. K. Comparing twitter summarization algorithms for multiple post summaries. Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on, 2011. IEEE, 298-306.

ISLAM, A. & INKPEN, D. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD),* 2**,** 10.

JAIN, A. K. & DUBES, R. C. 1988. *Algorithms for clustering data*, Prentice-Hall, Inc.

JAROMCZYK, J. W. & TOUSSAINT, G. T. 1992. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE,* 80**,** 1502-1517.

JEHL, L., HIEBER, F. & RIEZLER, S. Twitter translation using translation-based cross-lingual retrieval. Proceedings of the seventh workshop on statistical machine translation, 2012. Association for Computational Linguistics, 410-421.

JIANG, J. J. & CONRATH, D. W. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

JIANQIANG, Z. & XIAOLIN, G. 2017. Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access,* 5**,** 2870-2879.

JUNGHERR, A. 2016. Twitter use in election campaigns: A systematic literature review. *Journal of information technology & politics,* 13**,** 72-91.

JURAFSKY, D. 2000. Speech and language processing: An introduction to natural language processing. *Computational linguistics, and speech recognition*.

KANNAN, S. & GURUSAMY, V. 2014. Preprocessing Techniques for Text Mining.

KASHYAP, A., HAN, L., YUS, R., SLEEMAN, J., SATYAPANICH, T., GANDHI, S. & FININ, T. 2016. Robust semantic text similarity using LSA, machine learning, and linguistic resources. *Language Resources and Evaluation,* 50**,** 125-161.

KAUR, M. & KAUR, U. 2013. Comparison between k-means and hierarchical algorithm using query redirection. *International Journal of Advanced Research in Computer Science and Software Engineering,* 3.

KAUR, N. 2015. *A combinatorial tweet clustering methodology utilizing inter and intra cosine similarity.* Faculty of Graduate Studies and Research, University of Regina.

KHAN, K., SAHAI, A. & CAMPUS, A. 2012. A fuzzy c-means bi-sonar-based metaheuristic optomization algorithm. *IJIMAI,* 1**,** 26-32.

KIM, J., TABIBIAN, B., OH, A., SCHÖLKOPF, B. & GOMEZ-RODRIGUEZ, M. Leveraging the crowd to detect and reduce the spread of fake news and misinformation. Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018. ACM, 324-332.

KIM, S., JEON, S., KIM, J., PARK, Y.-H. & YU, H. Finding core topics: Topic extraction with clustering on tweet. Cloud and Green Computing (CGC), 2012 Second International Conference on, 2012. IEEE, 777-782.

KINNEAR, P. R. & GRAY, C. D. 1999. *SPSS for Windows made simple*, Taylor & Francis.

KLAUS, K. 1980. Content analysis: An introduction to its methodology. Sage Publications.

KUMAR, S., MORSTATTER, F. & LIU, H. 2014. *Twitter data analytics*, Springer.

LACHLAN, K. A., SPENCE, P. R. & LIN, X. 2014. Expressions of risk awareness and concern through Twitter: on the utility of using the medium as an indication of audience needs. *Computers in Human Behavior,* 35**,** 554-559.

LANDAUER, T. K. & DUMAIS, S. T. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review,* 104**,** 211.

LANIADO, D. & MIKA, P. Making sense of twitter. International Semantic Web Conference, 2010. Springer, 470-485.

LASTRA-DÍAZ, J. J., GARCÍA-SERRANO, A., BATET, M., FERNÁNDEZ, M. & CHIRIGATI, F. 2017. HESML: A scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset. *Information Systems,* 66**,** 97-118.

LEACOCK, C. & CHODOROW, M. 1998. Combining local context and WordNet similarity for word sense identification. *WordNet: An electronic lexical database,* 49**,** 265-283.

LEE, K., PALSETIA, D., NARAYANAN, R., PATWARY, M. M. A., AGRAWAL, A. & CHOUDHARY, A. Twitter trending topic classification. Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, 2011. IEEE, 251-258.

LESK, M. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. Proceedings of the 5th annual international conference on Systems documentation, 1986. ACM, 24-26.

LESKOVEC, J., RAJARAMAN, A. & ULLMAN, J. D. 2014. *Mining of massive datasets*, Cambridge university press.

LI, C., SUN, A., WENG, J. & HE, Q. 2015. Tweet segmentation and its application to named entity recognition. *IEEE Transactions on Knowledge and Data Engineering,* 27**,** 558-570.

LI, Q., SHAH, S., LIU, X. & NOURBAKHSH, A. 2017. Data sets: Word embeddings learned from tweets and general data. *arXiv preprint arXiv:1708.03994*.

LI, Y., MCLEAN, D., BANDAR, Z. A., O'SHEA, J. D. & CROCKETT, K. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering,* 18**,** 1138-1150.

LIN, D. An information-theoretic definition of similarity. Icml, 1998. Citeseer, 296-304.

LIN, Y.-S., JIANG, J.-Y. & LEE, S.-J. 2014. A similarity measure for text classification and clustering. *IEEE transactions on knowledge and data engineering,* 26**,** 1575-1590.

LIPPI, M. & TORRONI, P. 2016. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT),* 16**,** 10.

LIU, A. & KIRCHHOFF, K. 2018. Context Models for OOV Word Translation in Low-Resource Languages. *arXiv preprint arXiv:1801.08660*.

LIU, H. & WANG, P. 2014. Assessing Text Semantic Similarity Using Ontology. *JSW,* 9**,** 490-497.

LIU, X., LI, Y., WEI, F. & ZHOU, M. Graph-Based Multi-Tweet Summarization using Social Signals. COLING, 2012. 1699-1714.

MANNING, C., SURDEANU, M., BAUER, J., FINKEL, J., BETHARD, S. & MCCLOSKY, D. The Stanford CoreNLP natural language processing toolkit. Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, 2014. 55-60.

MARTINETZ, T. & SCHULTEN, K. 1991. A" neural-gas" network learns topologies.

MEHROTRA, R., SANNER, S., BUNTINE, W. & XIE, L. Improving lda topic models for microblogs via tweet pooling and automatic labeling. Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, 2013. ACM, 889-892.

MIHALCEA, R., CORLEY, C. & STRAPPARAVA, C. Corpus-based and knowledge-based measures of text semantic similarity. AAAI, 2006. 775-780.

MIKOLOV, T., CHEN, K., CORRADO, G. & DEAN, J. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G. S. & DEAN, J. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 2013b. 3111-3119.

MILLER, G. A. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review,* 63**,** 81.

MILLER, G. A., BECKWITH, R., FELLBAUM, C., GROSS, D. & MILLER, K. J. 1990. Introduction to WordNet: An on-line lexical database. *International journal of lexicography,* 3**,** 235-244.

MILLER, G. A. & CHARLES, W. G. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes,* 6**,** 1-28.

MIYAMOTO, S., SUZUKI, S. & TAKUMI, S. Clustering in tweets using a fuzzy neighborhood model. Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on, 2012. IEEE, 1-6.

MNIH, A. & HINTON, G. E. A scalable hierarchical distributed language model. Advances in neural information processing systems, 2009. 1081-1088.

MOHAMMADI, E., THELWALL, M., KWASNY, M. & HOLMES, K. L. 2018. Academic information on Twitter: A user survey. *PloS one,* 13**,** e0197265.

MONDAL, J. & DESHPANDE, A. 2014. Stream querying and reasoning on social data. *Encyclopedia of Social Network Analysis and Mining.* Springer.

MOZETIČ, I., TORGO, L., CERQUEIRA, V. & SMAILOVIĆ, J. 2018. How to evaluate sentiment classifiers for Twitter time-ordered data? *PloS one,* 13**,** e0194317.

MUKHERJEE, S. & BALA, P. K. 2017. Sarcasm detection in microblogs using Naïve Bayes and fuzzy clustering. *Technology in Society,* 48**,** 19-27.

NAILI, M., CHAIBI, A. H. & GHEZALA, H. H. B. 2017. Comparative study of word embedding methods in topic segmentation. *Procedia Computer Science,* 112**,** 340-349.

NOLAN, S. A. & HEINZEN, T. 2011. *Statistics for the behavioral sciences,* Macmillan.

O'SHEA, J., BANDAR, Z. & CROCKETT, K. 2013. A new benchmark dataset with production methodology for short text semantic similarity algorithms. *ACM Transactions on Speech and Language Processing (TSLP),* 10**,** 19.

O'SHEA, J., BANDAR, Z., CROCKETT, K. & MCLEAN, D. 2010. Benchmarking short text semantic similarity. *International Journal of Intelligent Information and Database Systems,* 4**,** 103-120.

O'SHEA, J., BANDAR, Z., CROCKETT, K. & MCLEAN, D. A comparative study of two short text semantic similarity measures. KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, 2008a. Springer, 172-181.

O'SHEA, J., BANDAR, Z., CROCKETT, K. & MCLEAN, D. 2008b. A comparative study of two short text semantic similarity measures. *Agent and Multi-Agent Systems: Technologies and Applications***,** 172-181.

O'SHEA, K., BANDAR, Z. & CROCKETT, K. 2010. A conversational agent framework using semantic analysis. *International Journal of Intelligent Computing Research (IJICR),* 1.

OKAZAKI, N., MATSUO, Y., MATSUMURA, N. & ISHIZUKA, M. 2003. Sentence extraction by spreading activation through sentence similarity. *IEICE TRANSACTIONS on Information and Systems,* 86**,** 1686-1694.

PALLANT, J. 2013. *SPSS survival manual,* McGraw-Hill Education (UK).

PAWAR, A. & MAGO, V. 2018. Calculating the similarity between words and sentences using a lexical database and corpus statistics. *arXiv preprint arXiv:1802.05667*.

PENNINGTON, J., SOCHER, R. & MANNING, C. Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014. 1532-1543.

PRAMANIK, S., WANG, Q., DANISCH, M., GUILLAUME, J.-L. & MITRA, B. 2017. Modeling cascade formation in Twitter amidst mentions and retweets. *Social Network Analysis and Mining,* 7**,** 41.

PURWITASARI, D., FATICHAH, C., ARIESHANTI, I. & HAYATIN, N. K-medoids algorithm on Indonesian Twitter feeds for clustering trending issue as important terms in news summarization. Information & Communication Technology and Systems (ICTS), 2015 International Conference on, 2015. IEEE, 95-98.

RADA, R., MILI, H., BICKNELL, E. & BLETTNER, M. 1989. Development and application of a metric on semantic nets. *IEEE transactions on systems, man, and cybernetics,* 19**,** 17-30.

RAMASWAMY, S. no date. Comparing the Efficiency of Two Clustering Techniques.

RATHORE, P., BEZDEK, J. C., ERFANI, S. M., RAJASEGARAR, S. & PALANISWAMI, M. 2018. Ensemble fuzzy clustering using cumulative aggregation on random projections. *IEEE Transactions on Fuzzy Systems,* 26**,** 1510-1524.

REIMERS, N., BEYER, P. & GUREVYCH, I. Task-oriented intrinsic evaluation of semantic textual similarity. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016. 87-96.

RESNIK, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

ROCCHIO, J. J. 1971. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing***,** 313-323.

ROSA, K. D., SHAH, R., LIN, B., GERSHMAN, A. & FREDERKING, R. 2011. Topical clustering of tweets. *Proceedings of the ACM SIGIR: SWSM*.

ROSS, C., TERRAS, M., WARWICK, C. & WELSH, A. 2011. Enabled backchannel: Conference Twitter use by digital humanists. *Journal of Documentation,* 67**,** 214-237.

ROUSSEEUW, P. J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics,* 20**,** 53-65.

RUBENSTEIN, H. & GOODENOUGH, J. B. 1965. Contextual correlates of synonymy. *Communications of the ACM,* 8**,** 627-633.

RUDRAPAL, D., DAS, A. & BHATTACHARYA, B. Measuring Semantic Similarity for Bengali Tweets Using WordNet. Proceedings of the International Conference Recent Advances in Natural Language Processing, 2015. 537-544.

SALEM, S. B., NAOUALI, S. & SALLAMI, M. 2017. Clustering Categorical Data Using the K-Means Algorithm and the Attribute's Relative Frequency. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering,* 11**,** 709-714.

SALTON, G. & BUCKLEY, C. 1987. Term weighting approaches in automatic text retrieval. Cornell University.

SALTON, G. & BUCKLEY, C. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management,* 24**,** 513-523.

SANNER, M. F. 1999. Python: a programming language for software integration and development. *J Mol Graph Model,* 17**,** 57-61.

SATYAPANICH, T., GAO, H. & FININ, T. Ebiquity: Paraphrase and semantic similarity in Twitter using skipgrams. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015. 51-55.

SCHÜTZE, H., MANNING, C. D. & RAGHAVAN, P. 2008. *Introduction to information retrieval*, Cambridge University Press.

SEIFZADEH, S., FARAHAT, A. K., KAMEL, M. S. & KARRAY, F. Short-Text Clustering using Statistical Semantics. Proceedings of the 24th International Conference on World Wide Web, 2015. ACM, 805-810.

SEVERINO, R. 2006. Getting Your Random Sample in Proc SQL.

SHAH, C. 2008. INLS 490-154W: Information Retrieval Systems Design and Implementation. Fall 2009.

SHARMA, A., LÓPEZ, Y. & TSUNODA, T. 2017. Divisive hierarchical maximum likelihood clustering. *BMC bioinformatics,* 18**,** 546.

SHEELA, L. J. 2016. A review of sentiment analysis in twitter data using Hadoop. *International Journal of Database Theory and Application,* 9**,** 77-86.

SINGH, T. & KUMARI, M. 2016. Role of text pre-processing in twitter sentiment analysis. *Procedia Computer Science,* 89**,** 549-554.

SINNOTT, R. O. & WANG, W. 2017. Estimating micro-populations through social media analytics. *Social Network Analysis and Mining,* 7**,** 13.

SOĞANCıOĞLU, G., ÖZTÜRK, H. & ÖZGÜR, A. 2017. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics,* 33**,** i49-i58.

SONI, R. & MATHAI, K. J. 2015. Improved Twitter Sentiment Prediction through Cluster-then-Predict Model. *arXiv preprint arXiv:1509.02437*.

SRIRAM, B., FUHRY, D., DEMIR, E., FERHATOSMANOGLU, H. & DEMIRBAS, M. Short text classification in twitter to improve information filtering. Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, 2010. ACM, 841-842.

STARCH, D. 1910. A demonstration of the trial and error method of learning. *Psychological Bulletin,* 7**,** 20.

STEIGER, E., WESTERHOLT, R., RESCH, B. & ZIPF, A. 2015. Twitter as an indicator for whereabouts of people? Correlating Twitter with UK census data. *Computers, Environment and Urban Systems,* 54**,** 255-265.

SULTAN, M. A. 2016. *Short-Text Semantic Similarity: Algorithms and Applications.* University of Colorado at Boulder.

SULTAN, M. A., BETHARD, S. & SUMNER, T. DLS $@ $ CU: Sentence Similarity from Word Alignment. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014. 241-246.

TENENBAUM, A. M. 1990. *Data structures using C*, Pearson Education India.

THOMPSON, M. A., MAJHAIL, N. S., WOOD, W. A., PERALES, M.-A. & CHABOISSIER, M. 2015. Social media and the practicing hematologist: Twitter 101 for the busy healthcare provider. *Current hematologic malignancy reports,* 10**,** 405-412.

TWITTER INTERNATIONAL COMPANY, T. 2018. Developer Agreement and Policy.

VANEGAS, J. & BONET, I. Clustering Algorithm Optimization Applied to Metagenomics Using Big Data. Conference on Information Technologies and Communication of Ecuador, 2018. Springer, 182-192.

VATHY-FOGARASSY, Á. & ABONYI, J. 2013. *Graph-based clustering and data visualization algorithms*, Springer.

VICENTE, M., BATISTA, F. & CARVALHO, J. P. Twitter gender classification using user unstructured information. Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on, 2015. IEEE, 1-7.

WANG, Z., BAI, G., CHOWDHURY, S., XU, Q. & SEOW, Z. L. 2017. TwiInsight: Discovering Topics and Sentiments from Social Media Datasets. *arXiv preprint arXiv:1705.08094*.

WIEMER-HASTINGS, P. Adding syntactic information to LSA. Proceedings of the Annual Meeting of the Cognitive Science Society, 2000.

WU, Z. & PALMER, M. Verbs semantics and lexical selection. Proceedings of the 32nd annual meeting on Association for Computational Linguistics, 1994. Association for Computational Linguistics, 133-138.

XIONG, H., STEINBACH, M., TAN, P.-N. & KUMAR, V. HICAP: Hierarchical clustering with pattern preservation. Proceedings of the 2004 SIAM International Conference on Data Mining, 2004. SIAM, 279-290.

XU, W., RITTER, A. & GRISHMAN, R. Gathering and generating paraphrases from twitter with application to normalization. Proceedings of the sixth workshop on building and using comparable corpora, 2013. 121-128.

YANG, Z., ALGESHEIMER, R. & TESSONE, C. J. 2016. A Comparative Analysis of Community Detection Algorithms on Artificial Networks. *Scientific Reports,* 6.

YOON, S., ELHADAD, N. & BAKKEN, S. 2013. A practical approach for content mining of tweets. *American journal of preventive medicine,* 45**,** 122-129.

ZADEH, L. A., ABBASOV, A. M. & SHAHBAZOVA, S. N. Analysis of Twitter hashtags: Fuzzy clustering approach. Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC), 2015 Annual Conference of the North American, 2015. IEEE, 1-6.

ZANZOTTO, F. M., PENNACCHIOTTI, M. & TSIOUTSIOULIKLIS, K. Linguistic redundancy in twitter. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011. Association for Computational Linguistics, 659-669.

ZESCH, T. 2010. *Study of semantic relatedness of words using collaboratively constructed semantic resources.* Technische Universität.

ZHANG, Z. & LAN, M. Estimating Semantic Similarity between Expanded Query and Tweet Content for Microblog Retrieval. TREC, 2014.

ZHAO, W. X., JIANG, J., WENG, J., HE, J., LIM, E.-P., YAN, H. & LI, X. Comparing twitter and traditional media using topic models. European Conference on Information Retrieval, 2011. Springer, 338-349.
ZHAO, Y. 2011. R and Data Mining: Examples and Case Studies.
ZHAO, Y. 2012. *R and data mining: Examples and case studies*, Academic Press.
ZHU, G. & IGLESIAS, C. A. 2017. Computing semantic similarity of concepts in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering,* 29**,** 72-85.

Appendices

Appendix A – The Metadata Associated with a Tweet

Attribute	Type	Description
created_at	String	UTC time when this Tweet was created. **Example**: "created_at":"Wed Aug 27 13:08:45 +0000 2008"
id	Int64	The integer representation of the unique identifier for this Tweet. This number is greater than 53 bits and some programming languages may have difficulty/silent defects in interpreting it. Using a signed 64 bit integer for storing this identifier is safe. Use id_str for fetching the identifier to stay on the safe side. **Example**: "id":114749583439036416
id_str	String	The string representation of the unique identifier for this Tweet. Implementations should use this rather than the large integer in id. **Example**: "id_str":"114749583439036416"
text	String	The actual UTF-8 text of the status update. **Example**: "text":"Tweet Button, Follow Button, and Web Intents"
source	String	Utility used to post the Tweet, as an HTML-formatted string. Tweets from the Twitter website have a source value of web. **Example**: "source":"Twitter for Mac"
truncated	Boolean	Indicates whether the value of the text parameter was truncated, for example, as a result of a retweet exceeding the original Tweet text length limit of 140 characters. Truncated text will end in ellipsis, like this ... Since Twitter now rejects long Tweets vs truncating them, the large majority of Tweets will have this set to false . Note that while native retweets may have their toplevel text property shortened, the original text will be available under the retweeted_status object and the truncated parameter will be set to the value of the original status (in most cases, false). **Example**: "truncated":true
in_reply_to_status_id	Int64	*Nullable.* If the represented Tweet is a reply, this field will contain the integer representation of the original Tweet's ID. **Example**: "in_reply_to_status_id":114749583439036416
in_reply_to_status_id_str	String	*Nullable.* If the represented Tweet is a reply, this field will contain the string representation of the original Tweet's ID. **Example**: "in_reply_to_status_id_str":"114749583439036416"
in_reply_to_user_id	Int64	*Nullable.* If the represented Tweet is a reply, this field will contain the integer representation of the original Tweet's author ID. This will not necessarily always be the user directly mentioned in the Tweet. **Example**: "in_reply_to_user_id":819797
in_reply_to_user_id_str	String	*Nullable.* If the represented Tweet is a reply, this field will contain the string representation of the original Tweet's author ID. This will not necessarily always be the user directly mentioned in the Tweet. **Example**: "in_reply_to_user_id_str":"819797"

in_reply_to_screen_name	String	*Nullable.* If the represented Tweet is a reply, this field will contain the screen name of the original Tweet's author. **Example**: "in_reply_to_screen_name":"twitterapi"

user	User object	The user who posted this Tweet. **Example** highlighting select attributes:
		
		{
		"user": {
		"id": 2244994945,
		"id_str": "2244994945",
		"name": "TwitterDev",
		"screen_name": "TwitterDev",
		"location": "Internet",
		"url": "https://dev.twitter.com/",
		"description": "Your source for Twitter news",
		"verified": true,
		"followers_count": 477684,
		"friends_count": 1524,
		"listed_count": 1184,
		"favourites_count": 2151,
		"statuses_count": 3121,
		"created_at": "Sat Dec 14 04:35:55 +0000 2013",
		"utc_offset": null,
		"time_zone": null,
		"geo_enabled": true,
		"lang": "en",
		"profile_image_url_https": "https://pbs.twimg.com/"
		}
		}

coordinates	Coordinates	*Nullable.* Represents the geographic location of this Tweet as reported by the user or client application. The inner coordinates array is formatted as geoJSON (longitude first, then latitude). **Example**:
		"coordinates":
		{
		"coordinates":
		[
		-75.14310264,
		40.05701649
		],
		"type":"Point"
		}

Place	Places	*Nullable* When present, indicates that the tweet is associated (but not necessarily originating from) a Place. **Example**: "place": { "attributes":{}, "bounding_box": { "coordinates": [[[-77.119759,38.791645], [-76.909393,38.791645], [-76.909393,38.995548], [-77.119759,38.995548]]],

		"type":"Polygon" }, "country":"United States", "country_code":"US", "full_name":"Washington, DC", "id":"01fbe706f872cb32", "name":"Washington", "place_type":"city", "url":"http://api.twitter.com/1/geo/id/0172cb32.json" }
quoted_status_id	Int64	This field only surfaces when the Tweet is a quote Tweet. This field contains the integer value Tweet ID of the quoted Tweet. **Example**: "quoted_status_id":114749583439036416
quoted_status_id_str	String	This field only surfaces when the Tweet is a quote Tweet. This is the string representation Tweet ID of the quoted Tweet. **Example**: "quoted_status_id_str":"114749583439036416"
is_quote_status	Boolean	Indicates whether this is a Quoted Tweet. **Example**: "is_quote_status":false
quoted_status	Tweet	This field only surfaces when the Tweet is a quote Tweet. This attribute contains the Tweet object of the original Tweet that was quoted.
retweeted_status	Tweet	Users can amplify the broadcast of Tweets authored by other users by retweeting. Retweets can be distinguished from typical Tweets by the existence of a retweeted_status attribute. This attribute contains a representation of the *original* Tweet that was retweeted. Note that retweets of retweets do not show representations of the intermediary retweet, but only the original Tweet.
quote_count	Integer	*Nullable.* Indicates approximately how many times this Tweet has been quoted by Twitter users. **Example**: "quote_count":1138 Note: This object is only available with the Premium and Enterprise tier products.
reply_count	Int	Number of times this Tweet has been replied to. **Example**: "reply_count":1585 Note: This object is only available with the Premium and Enterprise tier products.
retweet_count	Int	Number of times this Tweet has been retweeted. **Example**: "retweet_count":1585
favorite_count	Integer	*Nullable.* Indicates approximately how many times this Tweet has been liked by Twitter users. **Example**: "favorite_count":1138

entities	Entities	Entities which have been parsed out of the text of the Tweet. **Example**:
		"entities":
		{
		"hashtags":[],
		"urls":[],
		"user_mentions":[],
		"media":[],
		"symbols":[]
		"polls":[]
		}

extended_entities	Extended	When between one and four native photos or one video or

	Entities	one animated GIF are in Tweet, contains an array 'media' metadata. **Example**:
		"entities":
		{
		"media":[]
		}

favorited	Boolean	*Nullable.* Indicates whether this Tweet has been liked by the authenticating user. **Example**: "favorited":true
retweeted	Boolean	Indicates whether this Tweet has been Retweeted by the authenticating user. **Example**: "retweeted":false
possibly_sensitive	Boolean	*Nullable.* This field only surfaces when a Tweet contains a link. The meaning of the field doesn't pertain to the Tweet content itself, but instead it is an indicator that the URL contained in the Tweet may contain content or media identified as sensitive content. **Example**: "possibly_sensitive":true
filter_level	String	Indicates the maximum value of the filter_level parameter which may be used and still stream this Tweet. So a value of medium will be streamed on none, low, and medium streams. **Example**: "filter_level": "medium"
lang	String	*Nullable.* When present, indicates a BCP_47 language identifier corresponding to the machine-detected language of the Tweet text, or und if no language could be detected. **Example**: "lang": "en"

matching_rules	Array of Rule Objects	Present in *filtered* products such as Twitter Search and PowerTrack. Provides the *id* and *tag* associated with the rule that matched the Tweet. With PowerTrack, more than one rule can match a Tweet. **Example**:
		"matching_rules": " [{
		"tag": "rain Tweets",
		"id": 831566737246023680,
		"id_str": "831566737246023680"
		}, {
		"tag": "snow Tweet",
		"id": 831567402366218240,
		"id_str": "831567402366218240"
		}]"

Table A.1: The tweet metadata[14]

[14] https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object

Appendix B – Sample of the European Referendum Corpus

Text: RT @KGeorgievaEU: Following situation in Brussels. EU institutions working together to ensure security of staff& premises.Please stay home
Text: Belgian Terror Attacks: Only Brexit Can Save Britain From This Scourge Of Political Islam Waging War InEurope
Text: RT @goddersbloom: If not today, exactly when ?
Text: RT @nickymstevenson: What are the facts around Brexit? Check out @propacad speaker overview from economist Roger Martin-Fagg
Text: Brexit risks range from 'small' to 'severe': In three months time the UK will vote in a referendum on whether #dw
Text: RT @_DAGOSPIA_: FACCI: 'IL REFERENDUM ABROGATIVO SULLE TRIVELLE NON SERVE A NIENTE E CI COSTA 300 MILIONI'
Text: Iain Duncan Smith will do anything for Brexit even tell the truth @pollytoynbee
Text: Agreed. The primary focus should be on the victims of such heinous acts and their friends and families.
Text: Guardian: Can Glastonbury swing the #EUreferendum? Festival urges visitors to set up postal votes
Text: RT @LisaVikingstad: Classy #Brussels #PrayForTheWorld
Text: RT @realbritainros: This by @pollytoynbee on Iain Duncan Smith - 100%. "How can this Nosferatu say he never had a taste for blood?" - https
Text: RT @chrisem61: BOOMB IN BRUSSELS. So are you sure that you still want to stay in the EU... TAKE BACK CONTROL BREXIT THE EU
Text: #StrongerIn
Text: @SkyNewsBreak Should #molenbeck be torn down? Attacks have almost guaranteed that Britain will now leave the #EU. #Brexit #ISIS #Merkel
Text: Interesting read by @HuffingtonPost on how #Brexit could effect the #construction industry:
Text: RT @PrisonPlanet: Some people are more outrage over Farage's comments than the actual jihadist massacre itself. #Brussels
Text: E invece noi il 17 aprile votiamo s al referendum, contro l'ennesimo regalo di Renzi ai suoi. @dp_parisi @AlessiaMorani @micheleemiliano
Text: Belgian Terror Attacks: Only Brexit Can Save Britain From This Scourge Of Political Islam Waging War InEurope
Text: RT @OwenJones84: This is a sick attempt to politically exploit a horrendous atrocity.
Text: @astroehlein @allisonpearson Brexit, dick head Merkel, has let in floods of refugees with no account for who they are OUT
Text: RT @m_donato_91: "I trivellati" L'Appunto di @FilippoFacci1 su Libero Un #referendum cretino. #nostopitaly
Text: Unless your #BREXIT campaign involves stopping wars and bombs, terrorists will still exist in your brave new world.
Text: RT @katelallyx: As if people are using what's happened in #Brussels to score referendum points. Unbelievable.
Text: RT @QuentinMunroe: @CllrBSilvester It's not a matter of "if" but a matter of "when". America needs #Trump UK needs #Brexit
Text: We want YOU to share your views on the #EU Referendum. Are you In or out?
Text: RT @SJ_Powell: The economic case against Brexit is collapsing @CBItweets #LeaveEU #VoteLeave #GO via @CityAM
Text: RT @DavidHeadViews: Read this and feel justifiable revulsion: the truly ugly face of #Brexit fanaticism.
Text: @TheDirtyPurple @CountRollo @KTHopkins No it wasn't. It was a vote for a referendum. & Tories didn't tell every Muhammed to come. #Brexit
Text: BACK OFF BARRY: 100 MPs Tell Obama to Stay Out of EU Referendum Intervention via @regisgiles
Text: RT @chrisem61: BOOMB IN BRUSSELS. So are you sure that you still want to stay in the EU... TAKE BACK CONTROL BREXIT THE EU

Appendix C – Participant Consent Form

Noufa Alnajran
PhD in Computing
John Dalton Building
Faculty of Science and Engineering
Manchester Metropolitan University
noufa.alnajran@stu.mmu.ac.uk

Title of Project: *A Study of Twitter-Based Cluster Analysis*

Name of Researcher: Noufa Alnajran

Participant Identification Code for this project: Please initial box

1.	I confirm that I have read and understood the participant information sheet dated 23/07/2018 for the above project and have had the opportunity to ask questions about the experiment procedure.	☐
2.	I understand that my participation is voluntary and that I am free to withdraw at any time without giving any reason to the named researcher.	☐
3.	I understand that my similarity judgements of a selection of tweets will be used for evaluation purposes for this research project.	☐
4.	I understand that my input data will remain anonymous.	☐
5.	I agree to take part in the above research project.	☐
6.	I understand that at my request a copy of my judgements on tweets similarity can be made available to me.	☐

Participant's comments (optional)

_____	_____	_____
Name of Participant	Date	Signature
_____	_____	_____
Researcher	Date	Signature

Appendix D – Participant Information Sheet (PIS)

Researcher: Mrs. Noufa Alnajran **Supervisors:** Dr Keeley Crockett – Dr David McLean – Dr Annabel Latham
Address: E113
John Dalton Building
School of computing, Mathematics and Digital Technology
Chester Street
Manchester, M1 5GD

Phone: +(44)7481737292 **Email:** noufa.alnajran@stu.mmu.ac.uk

Study Title: A Study of Twitter-Based Semantic Similarity and Cluster Analysis

This **Participant Information Sheet** describes an experiment on evaluating the performance of a clustering algorithm on Twitter short text messages (*i.e.* tweets[15]) at Manchester Metropolitan University as part of a PhD research study.

Invitation to participate

I would like to invite you to take part in a research study about cluster analysis in the context of social media. Before you decide you need to understand why the research is being done and what it will involve you to do. Please take time to read the following information carefully. Ask questions if anything you read is not clear or would like more information. Take time to decide whether or not to take part.
The overall objective of this research study is:

- To develop an automated process for fining semantically similar groups of tweets in a Twitter-based dataset.
- In order to subjectively evaluate the accuracy of this process, this experiment aims at collecting human judgements on the belongingness of data points (i.e. tweets) to the most relevant category (i.e. cluster).
- The collected data from this experiment will be compared to the outcome of the developed clustering algorithm for performance evaluation.

What is the purpose of the study?

This study is undertaken as a part of validating a new algorithm that has been developed as part of a PhD research project. It aims at assessing how a computer algorithm can understand the meaning of tweets and accurately group the tweets into clusters which have similar meanings. Therefore, the purpose of the study is to acquire human judgements on Twitter-based cluster belongingness. The opinions of a human is then compared with that of the computer based semantic clustering algorithm.

Do I have to take part?

It is up to you to decide. I will describe the study and go through the information sheet, which I will give to you. I will then ask you to sign a consent form to show you agreed to take part. You are free to withdraw at any time, without giving a reason.

What will happen to me if I take part?

If you agree, you will be given a sheet containing a number of tweets representing a number of categories (representative tweets) along with a random selection of tweets

[15] A tweet is a post consisting of 140 characters or less on Twitter, which is a very popular social network and microblogging service.

text. During the exercise, you will be required to perform two tasks:

1. **Tweet categorisation** – for each tweet, please assign it to the most similar category based on your interpretation of the meaning of the text.
2. **Similarity assessment**– for each category, please assign a score to each tweet based on its similarity in meaning to the representative tweet for that category. Please assign a score between 0.0 (minimum similarity) and 5.0 (maximum similarity) according to the following scale.

    0.0 The overall meaning of the sentences is unrelated (on different topics).
    1.0 The overall meaning of the sentences is vaguely similar (on the same topic).
    2.0 The overall meaning of the sentences is clearly similar (share some details).
    3.0 The overall meaning of the sentences is very much alike (missing / different important information).
    4.0 The overall meaning of the sentences is strongly related (unimportant details differ).
    5.0 The overall meaning of the sentences is identical (equivalent).

To show finer degrees of similarity, you can use the first decimal place, for example if you think the similarity is half way between 3.0 and 4.0 you can use a value like 3.5.

Nature of the data

The data under consideration are political tweets in the context of the EU Referendum, as it has been an active trend in Online Social Networks and a rich source of controversy views. The United Kingdom European Union Membership (known as EU Referendum) took place on the 23rd of June 2016 in the UK. Based on a voting criteria, the campaign is supported to either remain in the European Union (EU) or leave. This data has been collected three months prior to the day of the referendum.

Why you were invited to take part?

You were invited because of your perceived expertise and interest in the political domain. Neither ethnicity, gender, nor mother language matter in this experiment.

What if I change my mind?

If you wish to withdraw at any time, please indicate through email stating that you no longer want to take part and destroy the experiment sheet. We will keep a copy of your consent form for the purposes of auditing the research study.

Do I receive financial compensation?

There is no financial compensation for taking part.

How long will it take?

This experiment should take no more than one hour to complete.

What are the possible disadvantages and risks of taking part?

There is no risk involved in taking part as no personal nor sensitive data will be asked. This experiment is similar to browsing Twitter during the EU Referendum campaign but with a judgement task.

What are the possible benefits of taking part?

We cannot promise the study will help you but the information we get from the study will help to increase the intelligence of computer algorithms in understanding the modern language used in social media, which will have implications on research and practice.

Will my taking part in the study be kept confidential?

This research experiment does not require collecting any personal information from any participant and therefore no sensitive personal data will be held. The Informed consent form containing your personalised data will be kept in a locked cupboard within Manchester Metropolitan University and be destroyed within 6 months after the end of the project. The anonymised judgements will be kept for research purposes.

What if I have concerns about the study?

If you have a concern about any aspect of this study, you should ask to speak to the researcher who will do their best to answer your questions:

What if I have a complaint about the study?

If you wish to make a complaint about this study, then please contact:
The Research Ethics and Governance Team at Manchester Metropolitan University (ethics@mmu.ac.uk, 0161 247 2853)

Investigator (researcher):

Noufa Alnajran (noufa.alnajran@stu.mmu.ac.uk)

Appendix E – The Experiment Questionnaire

This research is interested in analysing human generated short text messages in online social networks (OSN), particularly Twitter. Twitter is an active public OSN where users connect with each other through posting tweets. These tweets are short texts used for sharing insights and sending out updates and reports on current events. Tweets are limited to 140 characters, which might seem too little to express yourself clearly. However, tweeters have come up with a variety of ways to turn their tweets into unique content formats. This has imposed lots of noise such as misspellings, abbreviations, and out of vocabulary words, which makes it difficult for computers to capture the meaning of tweets and find similar ones.

This experiment is set out to collect human perceptions on the similarity of tweets in order to evaluate the performance of machine learning algorithms. Due to the nature of the language used in OSN, you may find some of the words that are used in tweets offensive. If you do, please withdraw from the experiment.

Section (a) Tweet Categorisation

For each tweet in Table 2, please assign it to the most similar category in Table 1 based on your interpretation of the meaning of the tweet (if you don't know, please put the best match).

Category	Representative tweets
A	Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today
B	EU Referendum Briefing on Living and Working in the EU #ProtectJobs #Expats
C	Sterling slides on renewed Brexit worries
D	#Brexit Emerges As Threat To TTIP[16] Deal
E	It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union

Table E.1 Representative tweets for each category

Id	Tweet	Best Match
1	Has anyone from the Brexit Brigade addressed the issue of what will happen to existing EU citizens living and working in the UK	
2	Sterling has dipped cause markets believe Brexit will happen-GOOD-spivs in the city will adjust after playing their gambling games	
3	How can we save NHS inside EU	
4	I'm very sad for the families of the Brussels victims, but not at all surprised it happened! Wake up Europe #Brexit	
5	On one hand, there are decent human beings that send their sympathies to the Brussels victims and their families. And then there's Brexit.	
6	#Brussels attacks: Terrorism could break the EU and lead to Brexit	

[16] Transatlantic Trade and Investment Partnership (TTIP)

7	Sterling slides on renewed Brexit worries - A decline of 1% marks the 25th day this year the pound has moved	
8	@caddenlimos connecting low paid workers doing non skilled jobs from Poland with terrorism in Belgium	
9	Brussels bombing rose Brexit risk. How to trade Pound in case of Brexit: GBP $USD #FX	
10	I did worry about threat to NHS from TTIP - but EU and @EU_TTIP_team have listened to our concerns @HealthierIn	
11	UK's NHS will NOT survive staying in the EU	
12	#Brexit, a new threat to TTIP transatlantic trade talks	
13	We must stay in #EU to protect jobs	
14	Jobs rely on trade NOT a political union. Security relies on sharing information NOT a political union. #Brexit #StrongerIn #VoteLeave	
15	Naive to think Brexit would solve the #ttip problem...only way to protect #NHS is for govt to exclude it from TTIP	

Table E.2 Tweet categorization

Section (b) Similarity Assessment

For each category in Table 3, please assign a score to each tweet by writing a number between 0.0 (minimum similarity) and 5.0 (maximum similarity) based on its similarity in meaning to the representative tweet for that category using the following similarity scale.

0.0 The overall meaning of the sentences is unrelated (on different topics).
1.0 The overall meaning of the sentences is vaguely similar (on the same topic).
2.0 The overall meaning of the sentences is clearly similar (share some details).
3.0 The overall meaning of the sentences is very much alike (missing/different important information).
4.0 The overall meaning of the sentences is strongly related (unimportant details differ).
5.0 The overall meaning of the sentences is identical (equivalent).

You can use the first decimal place, for example if you think the similarity is half way between 3.0 and 4.0 you can use a value like 3.5 to show finer degrees of similarity.

Category representative tweet	**Tweets**	**Similarity Score**
Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today	Brussels attacks may sway Brexit vote: Strategists	
	On one hand, there are decent human beings that send their sympathies to the Brussels victims and their families. And then there's Brexit.	
	#Brussels attacks: Terrorism could break the EU and lead to Brexit	
	Terrorism is the scariest think. And it's ways more scarier if it's in the EU, in your home. Stay strong Brussels! #prayersforBrussels	
	Brussels Attacks Spur Brexit Campaign: Anti-Immigration Parties Link Terror To EU Open Borders	
	The world is seriously fucked up right now.	
EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats	@caddenlimos connecting low paid workers doing non skilled jobs from Poland with terrorism in Belgium	
	Has anyone from the Brexit Brigade addressed the issue of what will happen to existing EU citizens	

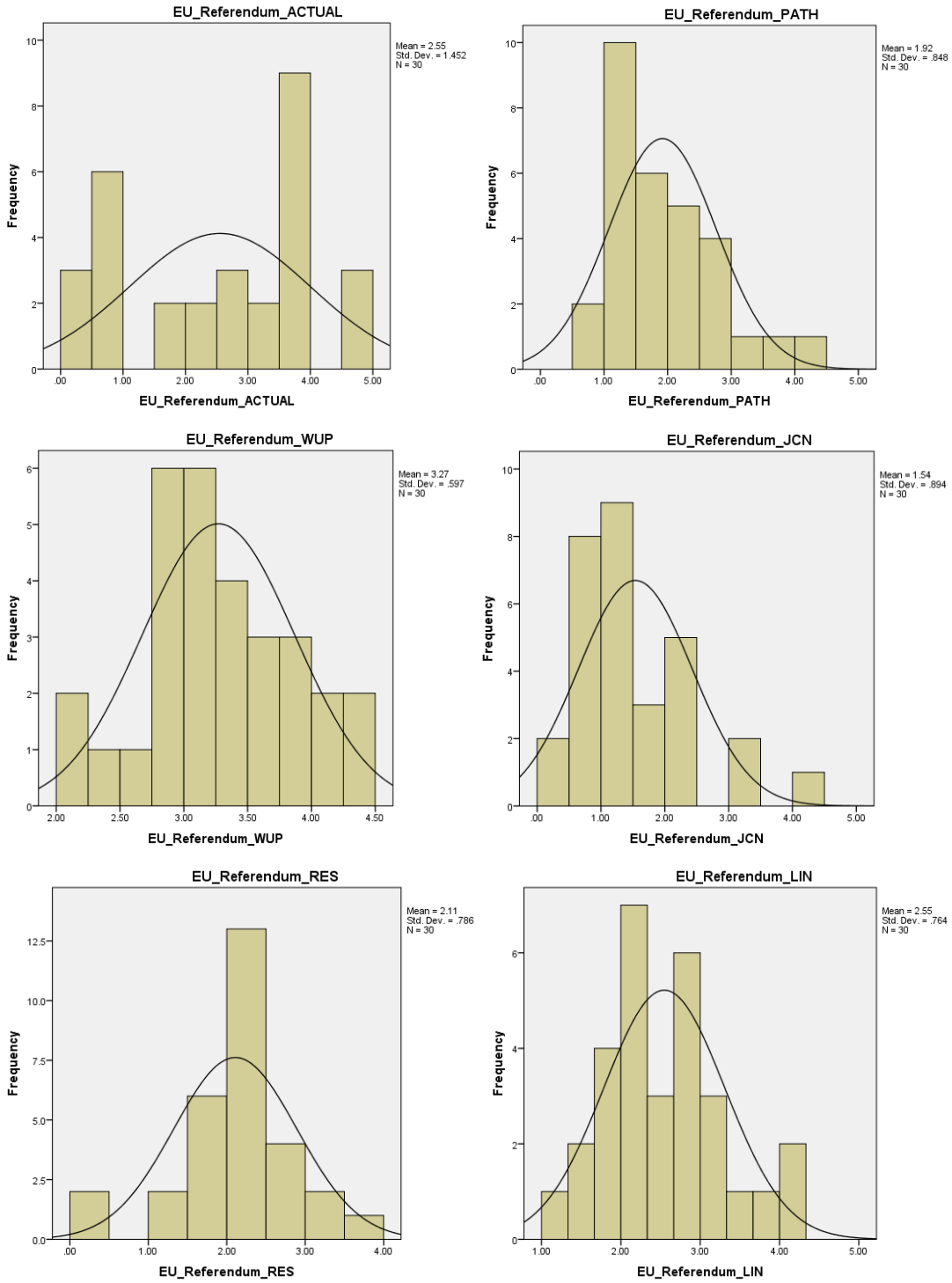
	living and working in the UK	
	@thebobevans Today's atrocity foreseable under EU policy. Trust UK security services to protect UK citizens. Brexit	
	#Brexit supporters claim EU needs UK more than we need it. 45% of UK exports go to EU, 10% of EU exports come here	
	Could 2m+ 18-34 Year Old Workers Emigrating After a Brexit Cause a Recruitment Nightmare?	
	We must stay in #EU to protect jobs	
Sterling slides on renewed Brexit worries	Sterling slides on renewed Brexit worries - A decline of 1% marks the 25th day this year the pound has moved	
	London-based crowdfunding platform Seedrs poll on the EU referendum finds 47% of investors and 43% of entrepreneurs	
	Brussels bombing rose Brexit risk. How to trade Pound in case of Brexit: GBP $USD #FX	
	Sterling has dipped cause markets believe Brexit will happen-GOOD-spivs in the city will adjust after playing their gambling games	
	In most scenarios #Brexit will impose a significant long-term cost on the UK economy #OEBrexit	
	it's not just an economic argument	
#Brexit Emerges As Threat To TTIP Deal	Jobs rely on trade NOT a political union. Security relies on sharing information NOT a political union. #Brexit #StrongerIn #VoteLeave	
	#Brexit, a new threat to TTIP transatlantic trade talks	
	Naive to think Brexit would solve the #ttip problem...only way to protect #NHS is for govt to exclude it from TTIP	
	Benign Brexit would require accepting high levels of immigration and deep trade agreement with EU	
	Brexit Risks Rising	
	Negotiating trade agreements after #Brexit would be complicated for UK as there's no @wto for #services: @angusarmstrong8 at @FedTrust event	
It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union | via @Telegraph	UK's NHS will NOT survive staying in the EU	
	What would #Brexit mean for the #pharma industry?	
	To the "expats" in spain who are moaning about immigration can i just say this to you? Jog the fuck on you UTTER hypocrites	
	How can we save NHS inside EU	
	We send £350 million to Brussels every week - enough to build a new NHS hospital every week. Let's #VoteLeave and #TakeControl	
	The EU referendum is a vote for the EU or the NHS, we can't have both	

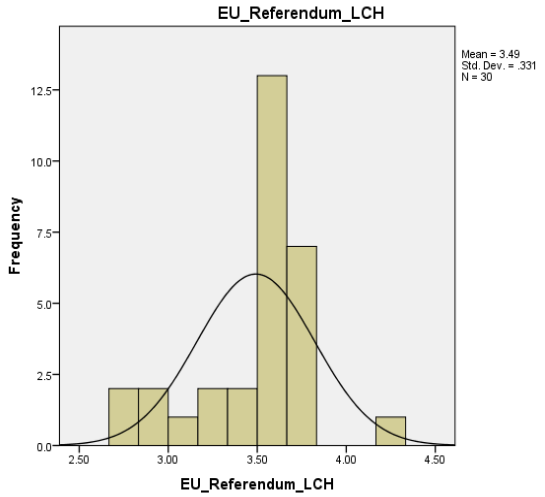
Table E.3 Tweets similarity assessment

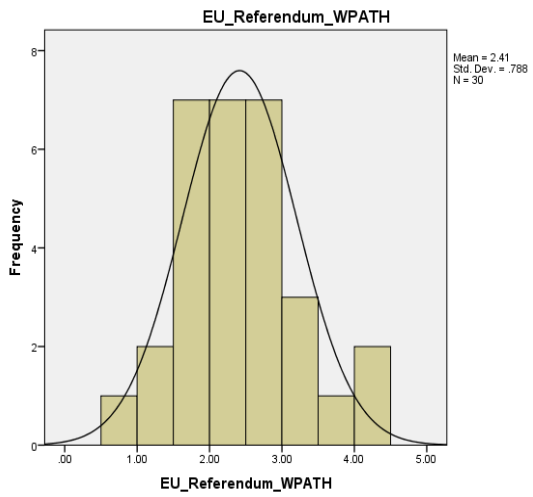
Thank you for taking part in this research experiment.

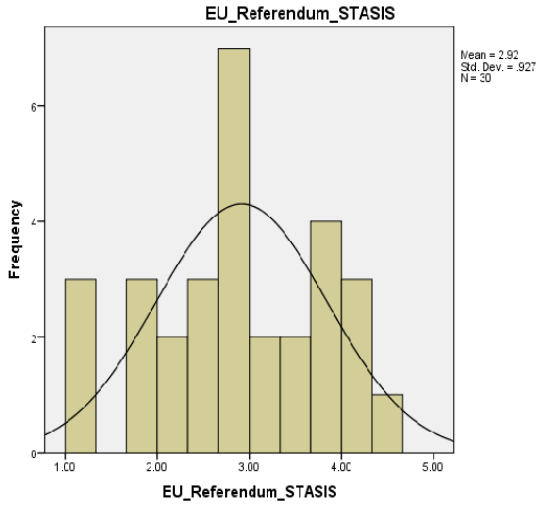
Appendix F – Normality histograms of the Human Similarity (Actual) and STSS (Estimated) Values

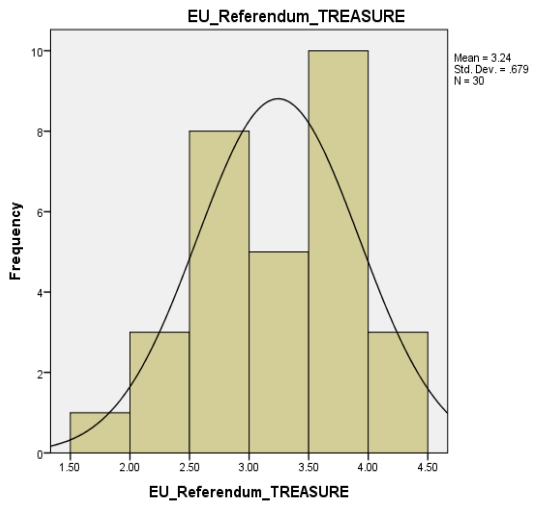
F.1 The EU_Referendum Dataset



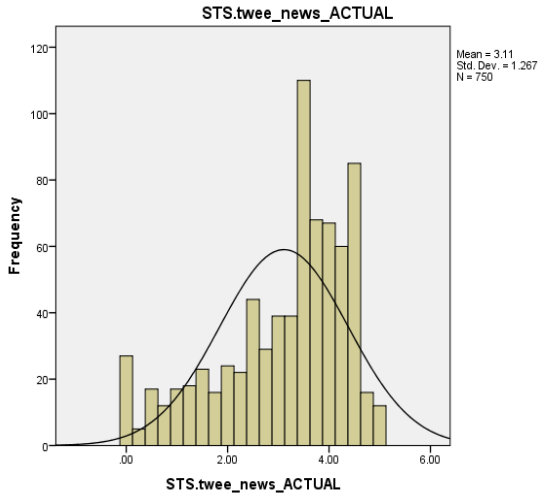


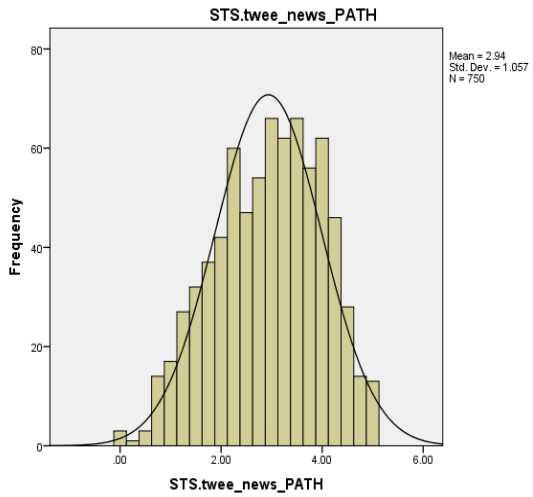


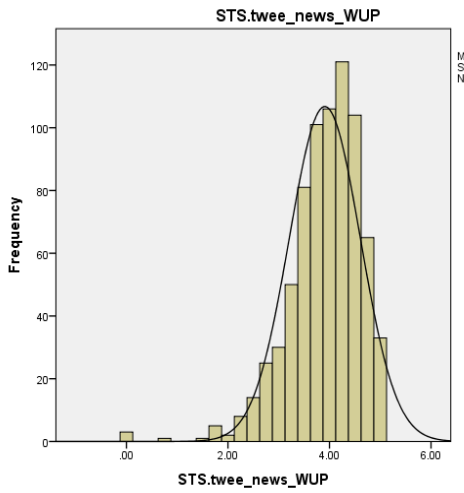


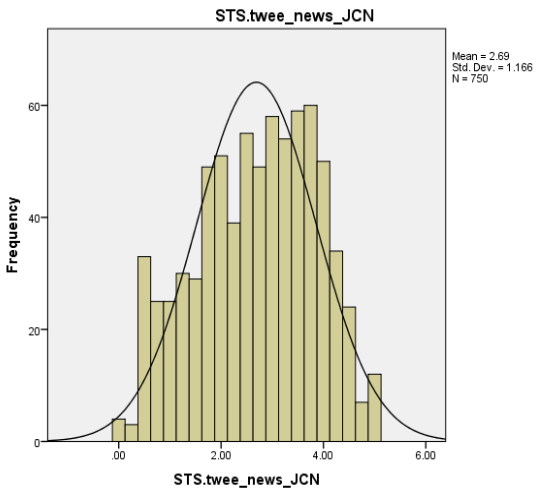


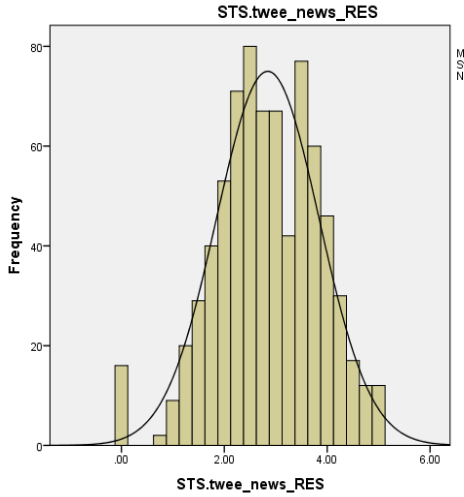
F.2 The STS.tweet_news Dataset

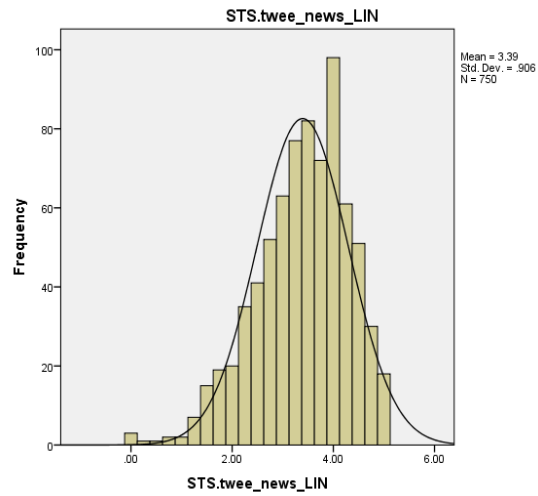


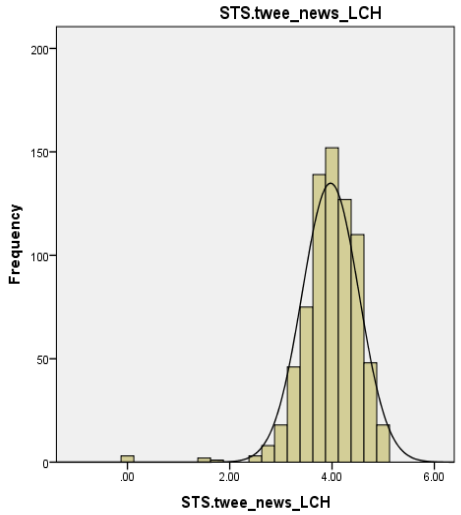


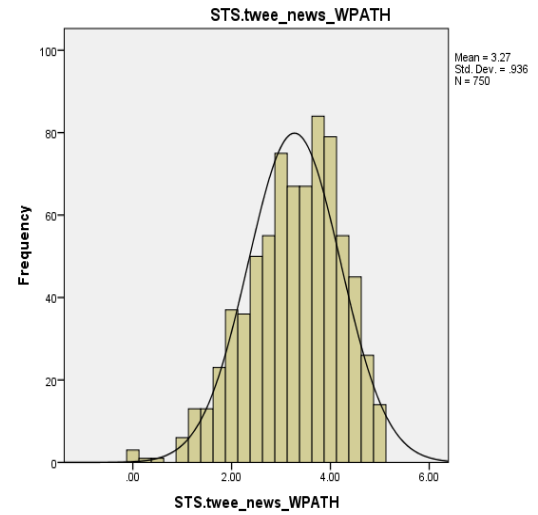

STS.twee_news_WUP
Mean = 3.91
Std. Dev. = .70
N = 750
Frequency
STS.twee_news_WUP

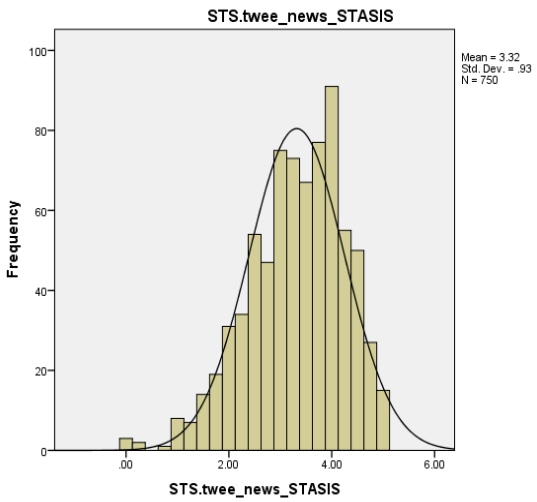

STS.twee_news_JCN
Mean = 2.69
Std. Dev. = 1.166
N = 750
Frequency
STS.twee_news_JCN

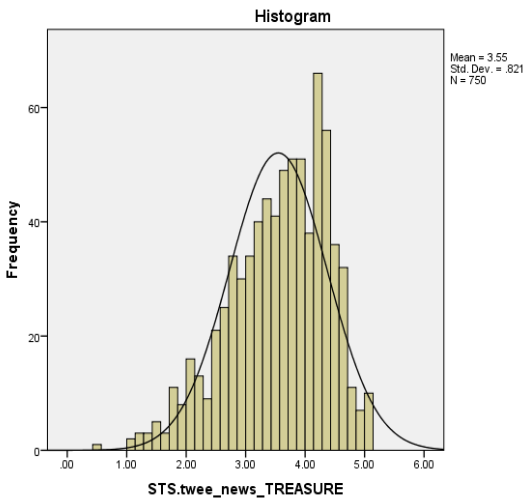

STS.twee_news_RES
Mean = 2.84
Std. Dev. = .997
N = 750
Frequency
STS.twee_news_RES


STS.twee_news_LIN
Mean = 3.39
Std. Dev. = .906
N = 750
Frequency
STS.twee_news_LIN


STS.twee_news_LCH
Mean = 3.97
Std. Dev. = .555
N = 750
Frequency
STS.twee_news_LCH

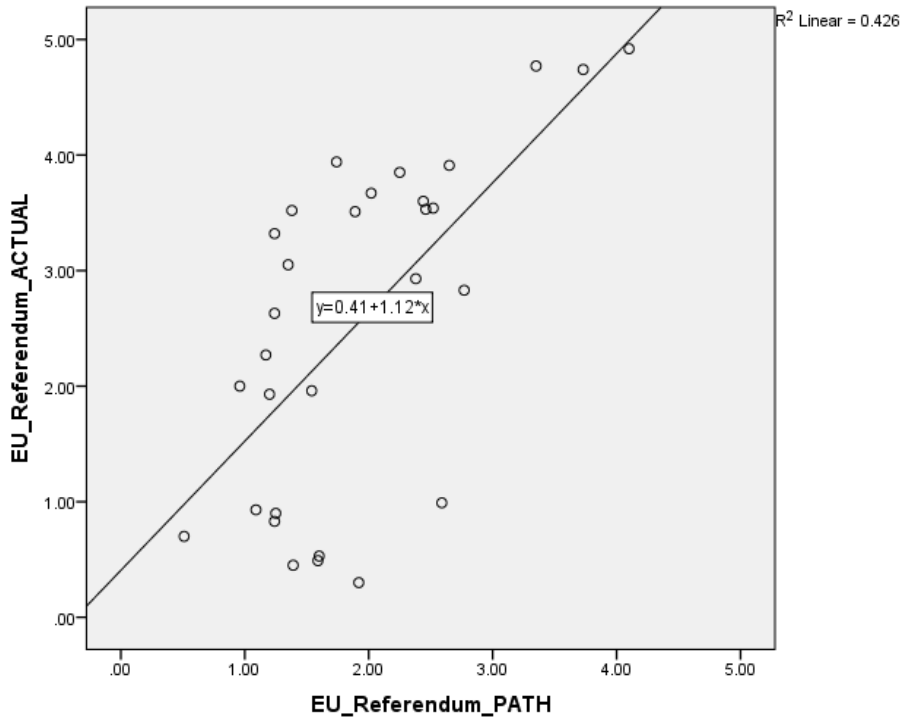

STS.twee_news_WPATH
Mean = 3.27
Std. Dev. = .936
N = 750
Frequency
STS.twee_news_WPATH

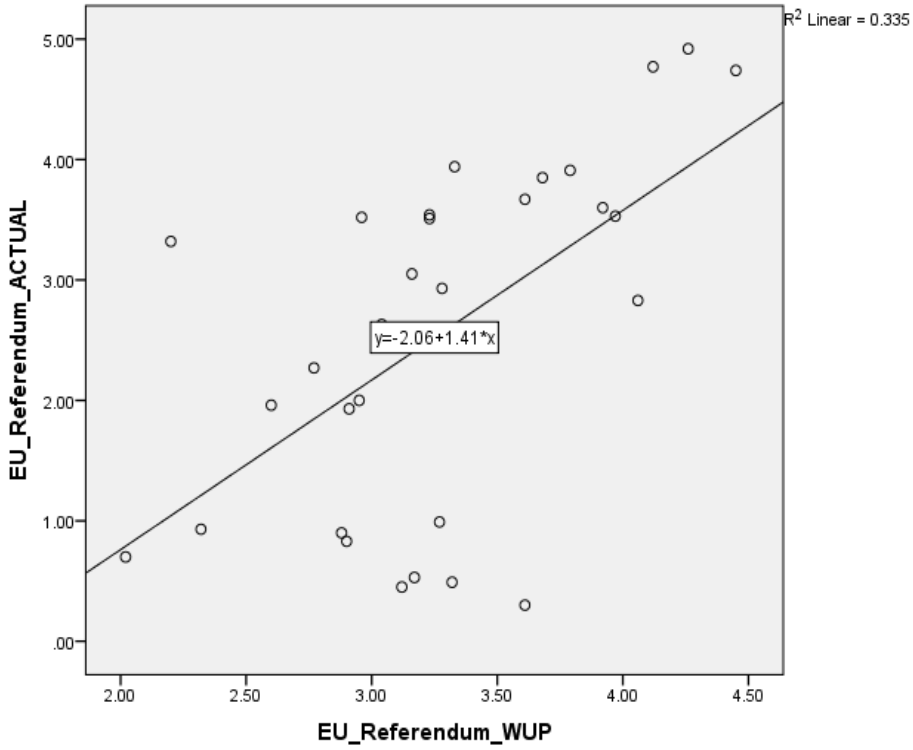

STS.twee_news_STASIS
Mean = 3.32
Std. Dev. = .93
N = 750
Frequency
STS.twee_news_STASIS

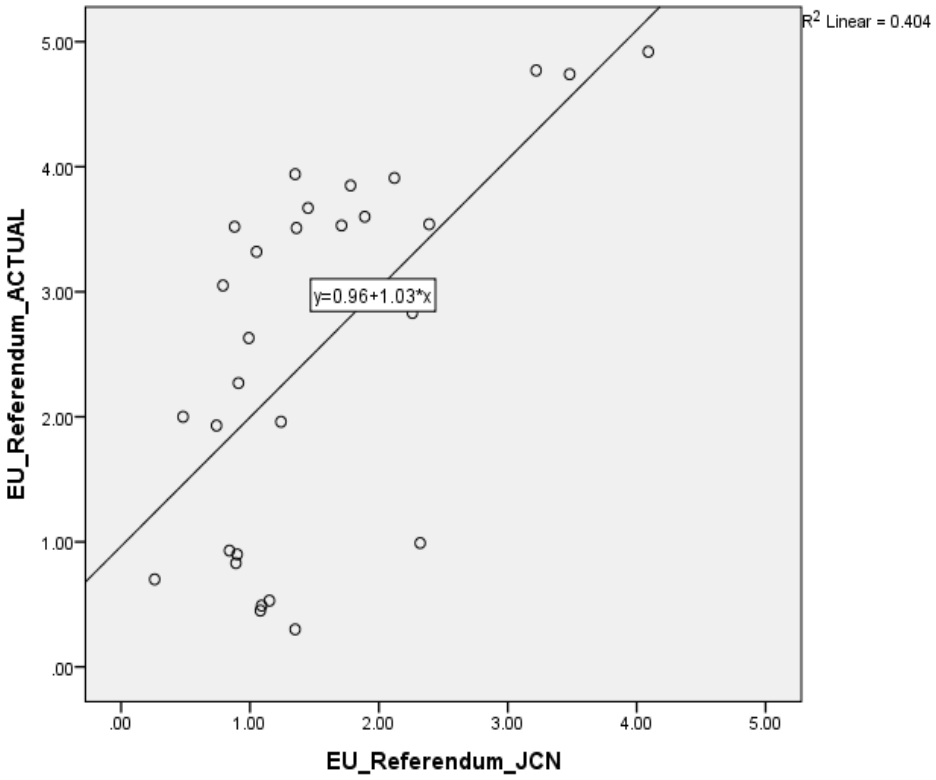

Histogram
Mean = 3.55
Std. Dev. = .821
N = 750
Frequency
STS.twee_news_TREASURE

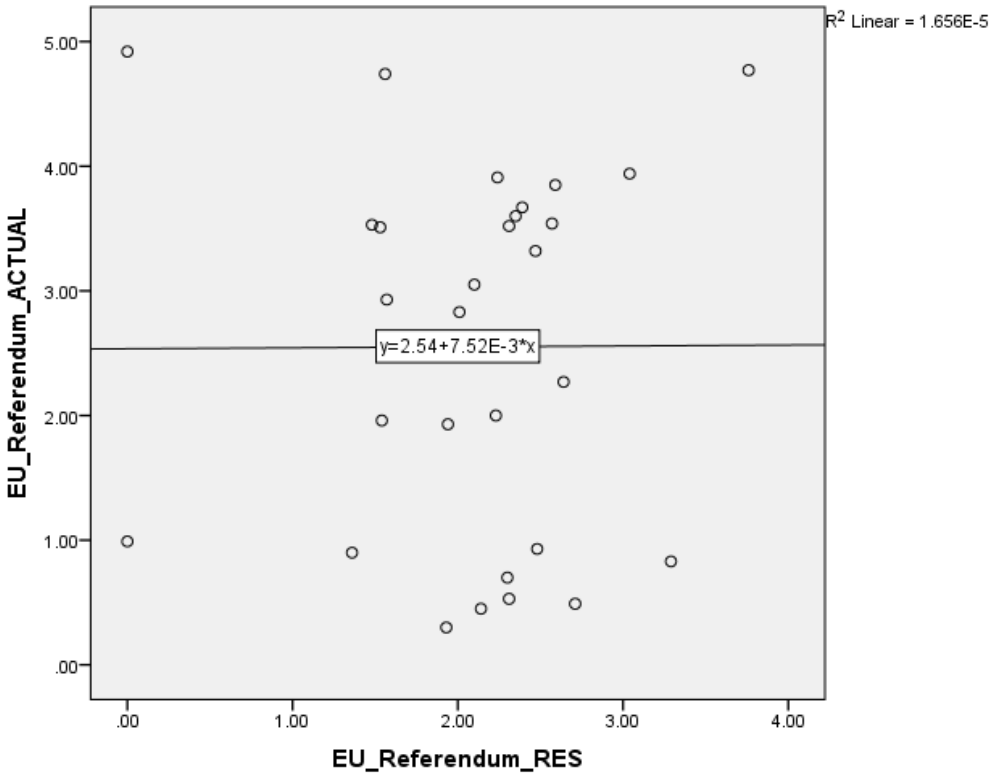
Appendix G – Correlation Scatterplots of the Human Similarity (Actual) and STSS (Estimated) Values

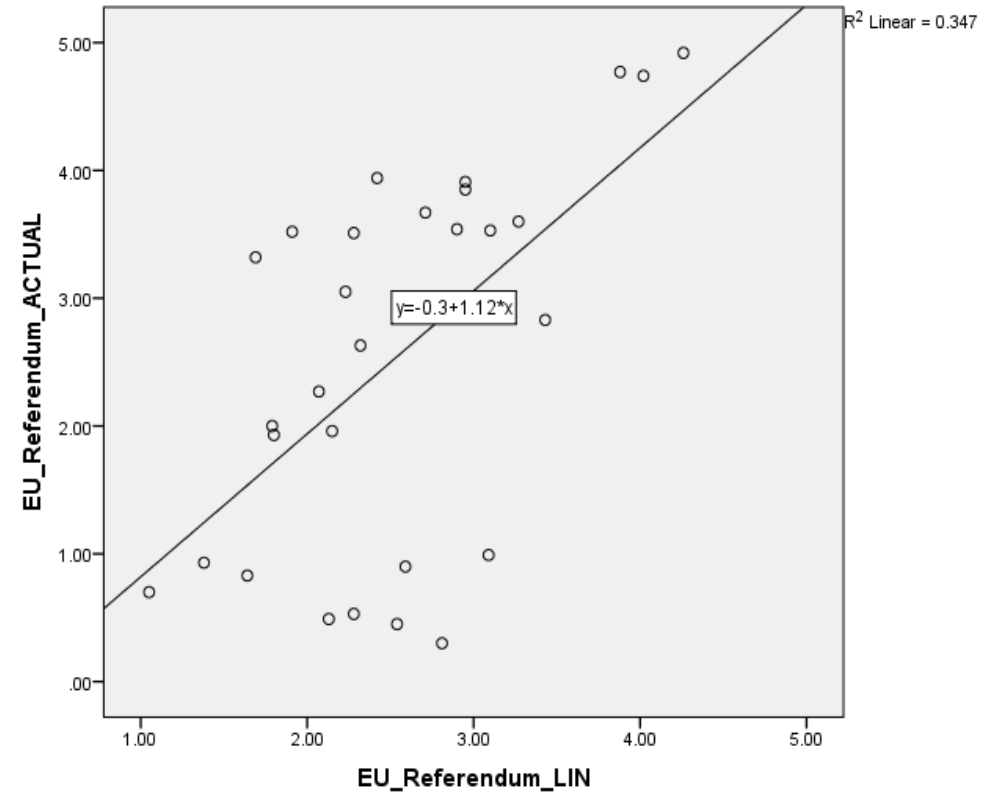
G.1 The EU_Referendum Dataset

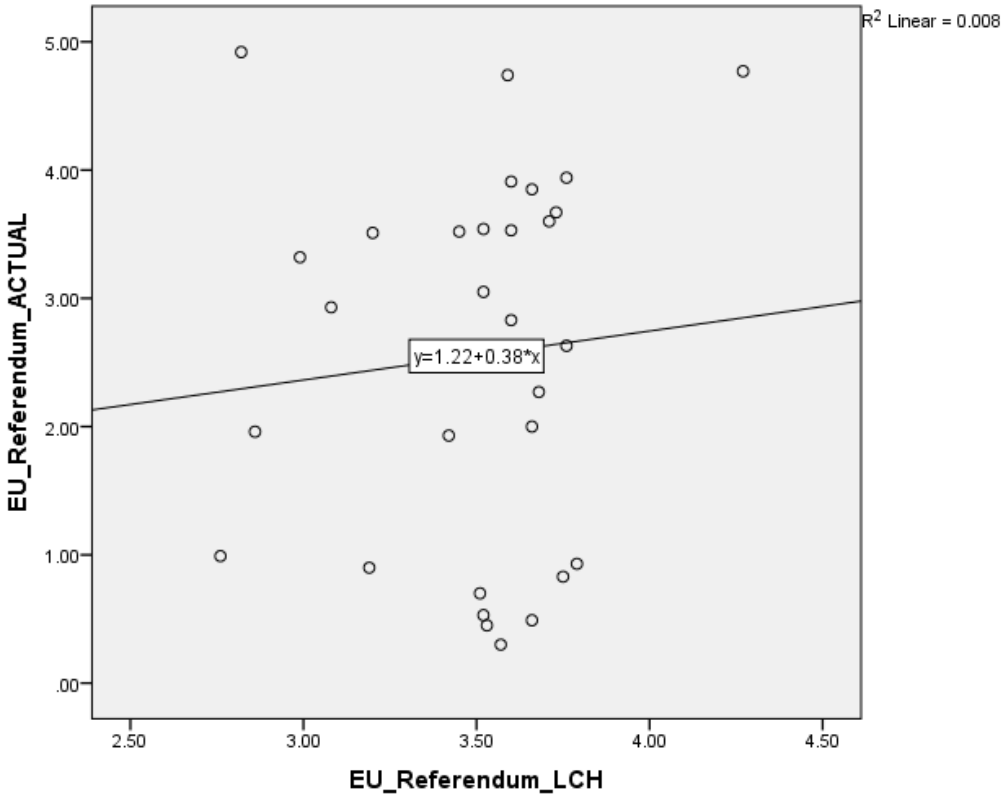


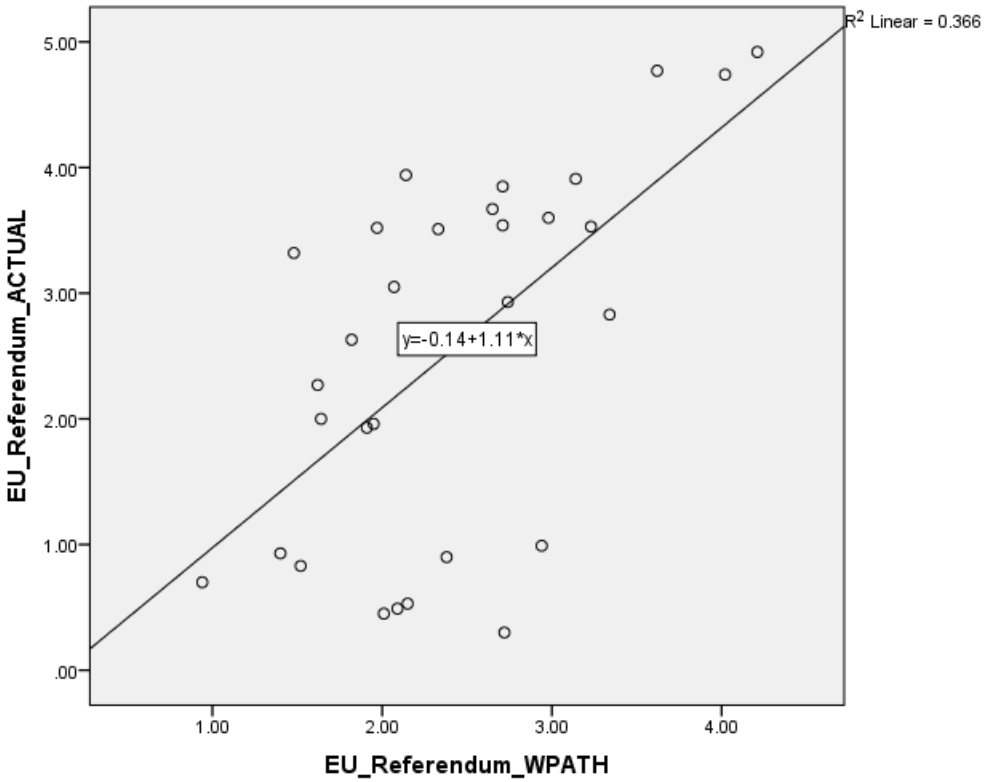


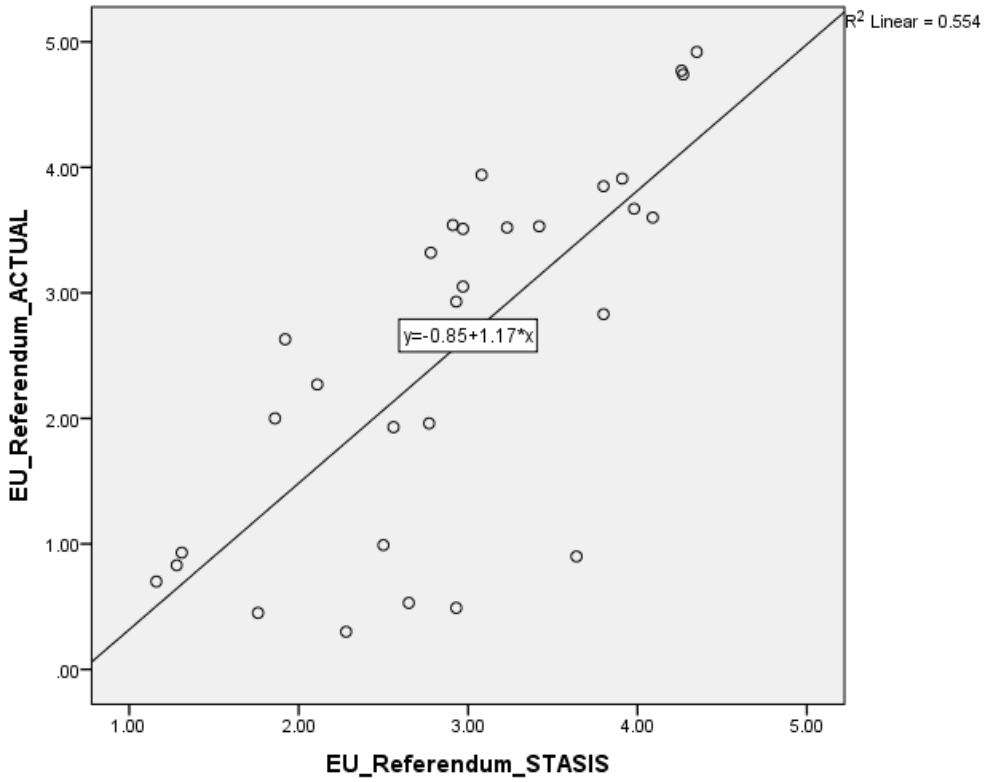

R2 Linear = 0.404
y=0.96+1.03*x
EU_Referendum_ACTUAL
EU_Referendum_JCN
5.00
4.00
3.00
2.00
1.00
.00

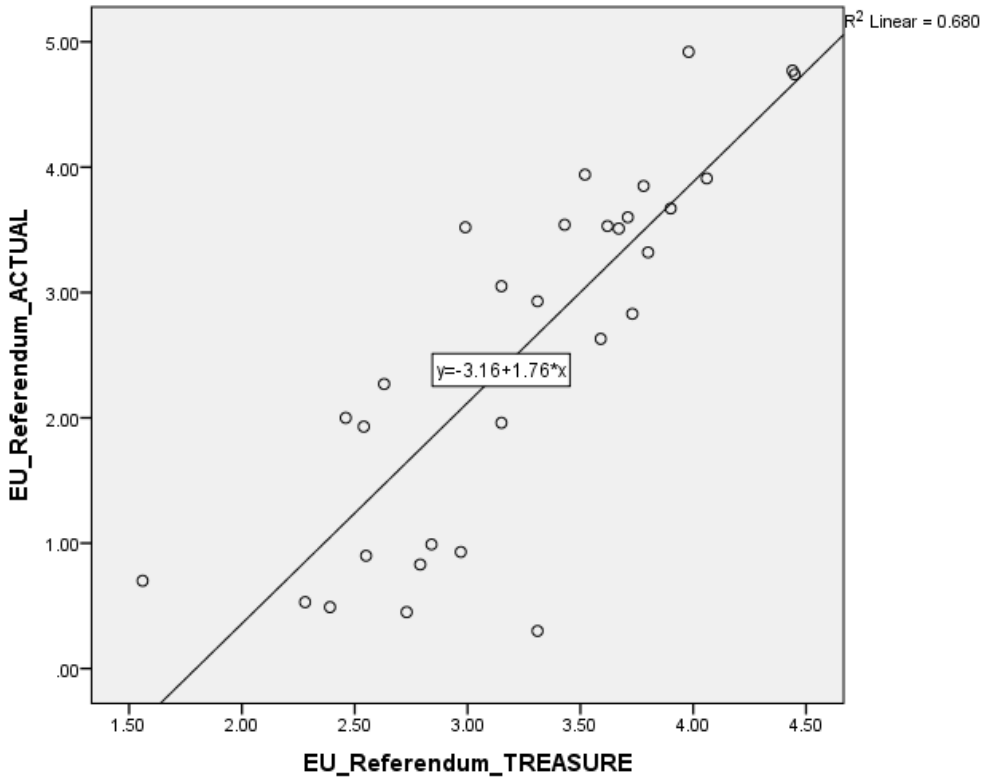

R2 Linear = 1.656E-5
y=2.54+7.52E-3*x
EU_Referendum_ACTUAL
EU_Referendum_RES
5.00
4.00
3.00
2.00
1.00
.00


R² Linear = 0.347
y=-0.3+1.12*x
EU_Referendum_ACTUAL
EU_Referendum_LIN
5.00
4.00
3.00
2.00
1.00
.00
1.00
2.00
3.00
4.00
5.00

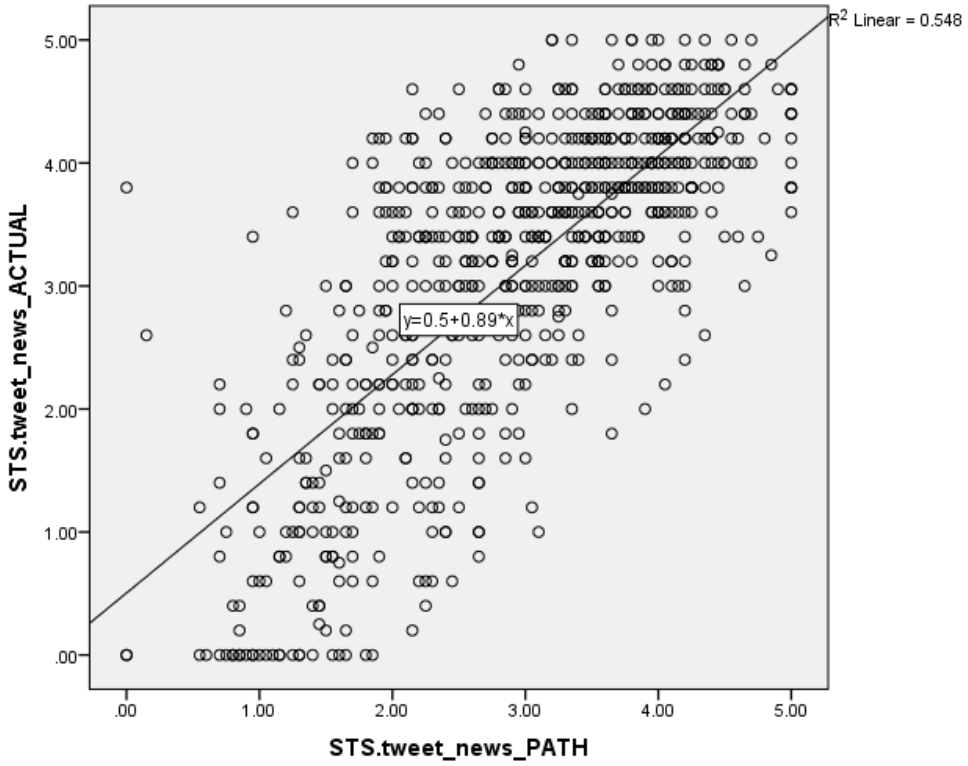

R² Linear = 0.008
y=1.22+0.38*x
EU_Referendum_ACTUAL
EU_Referendum_LCH
5.00
4.00
3.00
2.00
1.00
.00
2.50
3.00
3.50
4.00
4.50

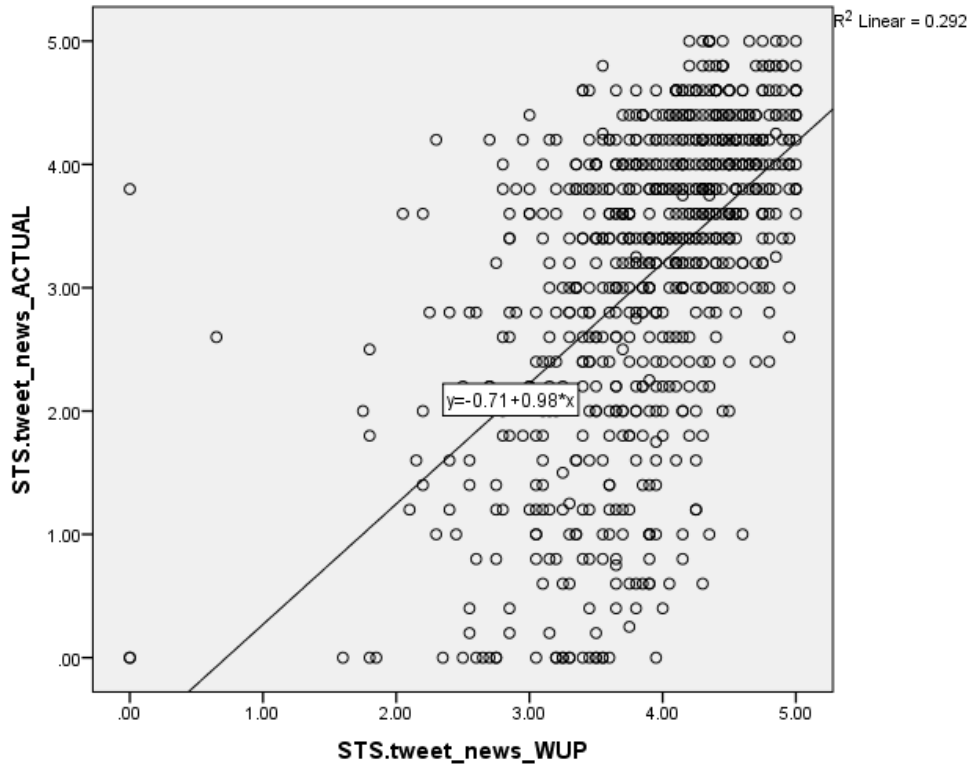

R² Linear = 0.366
y=-0.14+1.11*x
EU_Referendum_ACTUAL
EU_Referendum_WPATH
5.00
4.00
3.00
2.00
1.00
.00
1.00
2.00
3.00
4.00

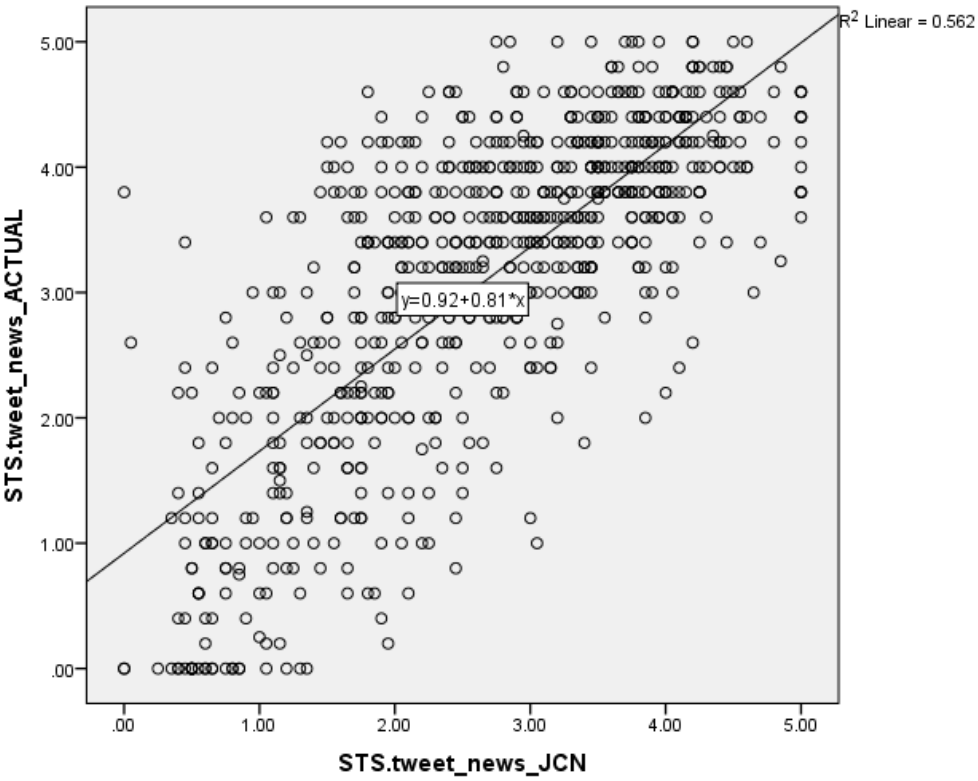

R² Linear = 0.554
y=-0.85+1.17*x
EU_Referendum_ACTUAL
EU_Referendum_STASIS
5.00
4.00
3.00
2.00
1.00
.00
1.00
2.00
3.00
4.00
5.00

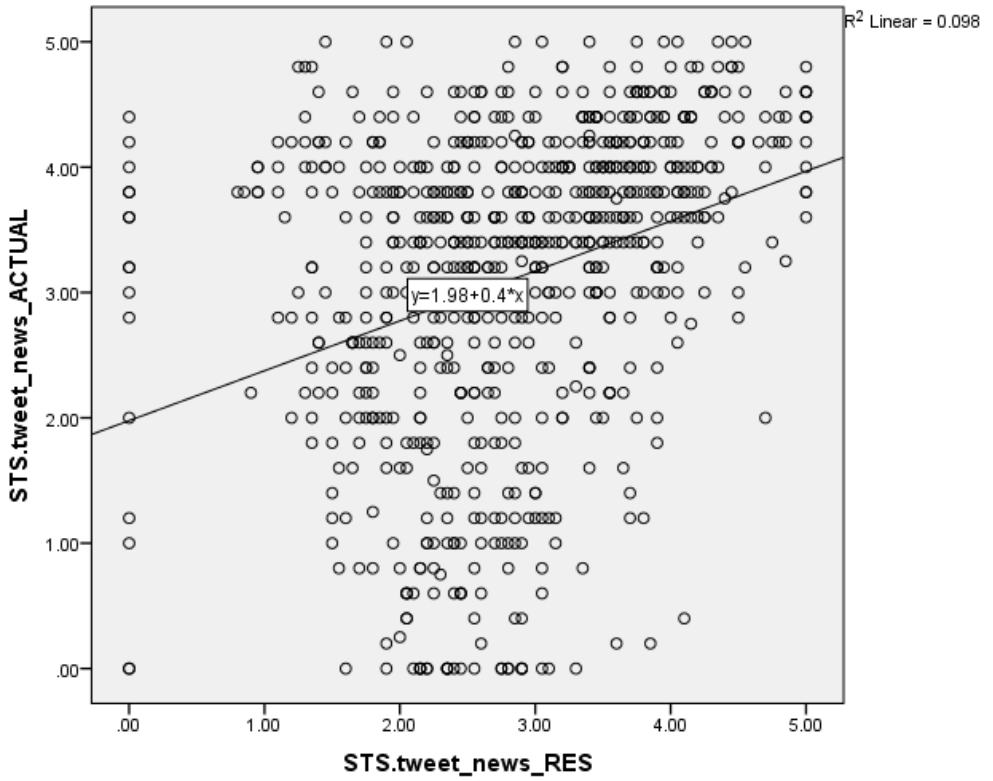


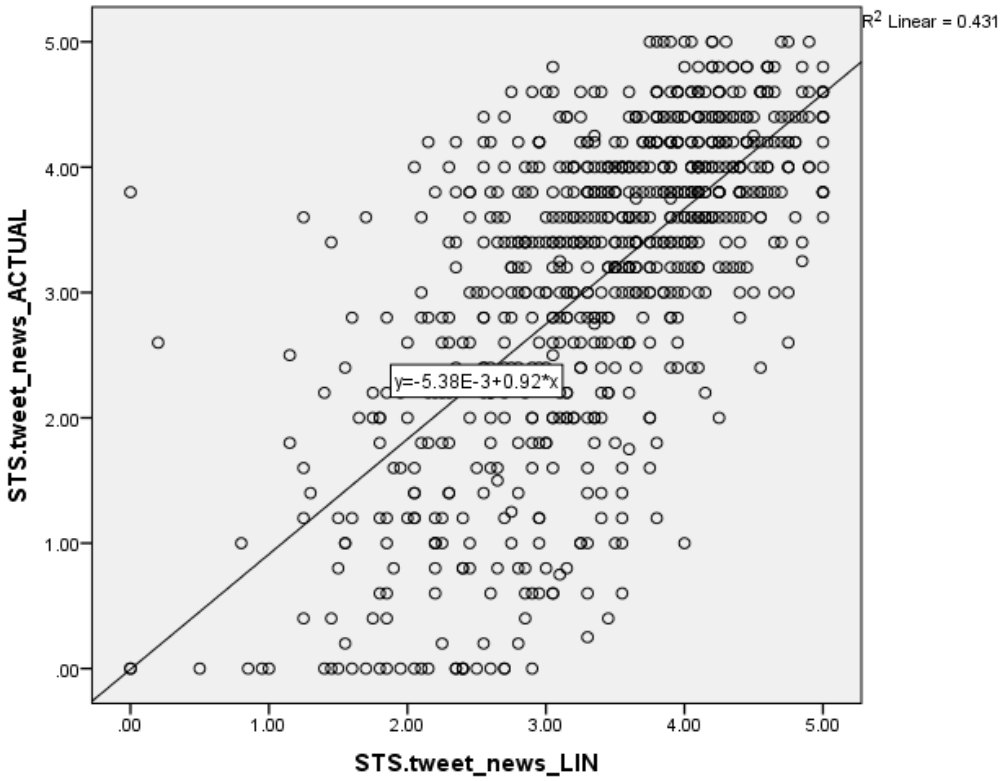
G.2 The STS.tweet_news Dataset

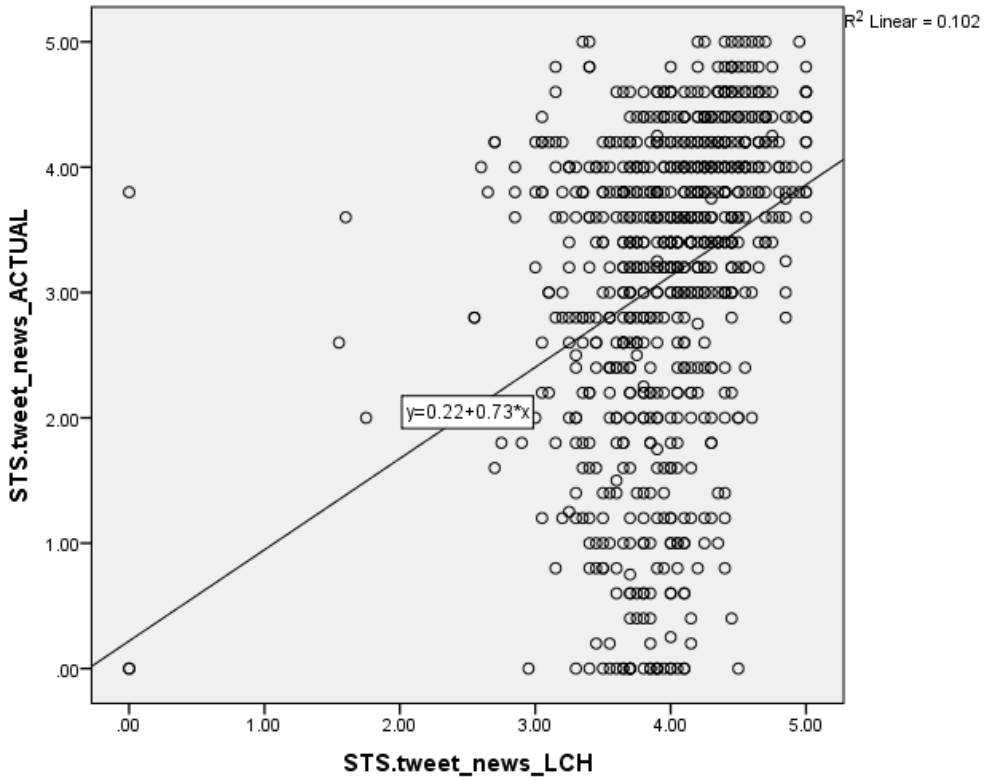


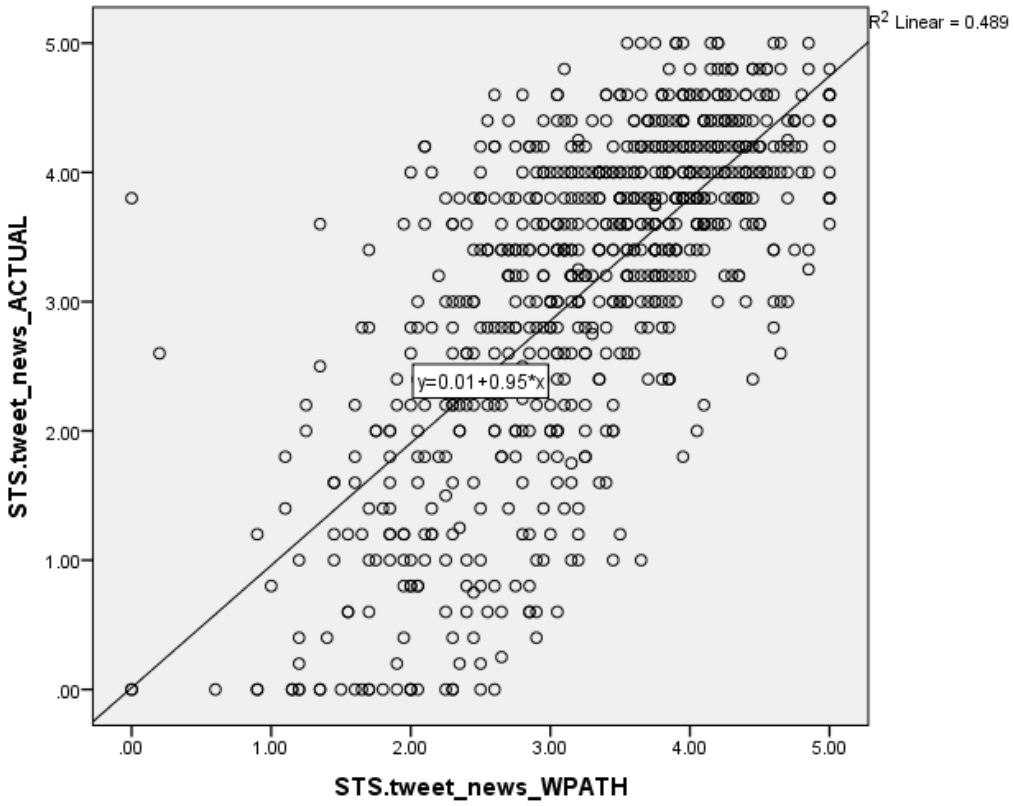

R2 Linear = 0.292
y=-0.71+0.98*x
STS.tweet_news_ACTUAL
STS.tweet_news_WUP
5.00
4.00
3.00
2.00
1.00
.00
.00
1.00
2.00
3.00
4.00
5.00

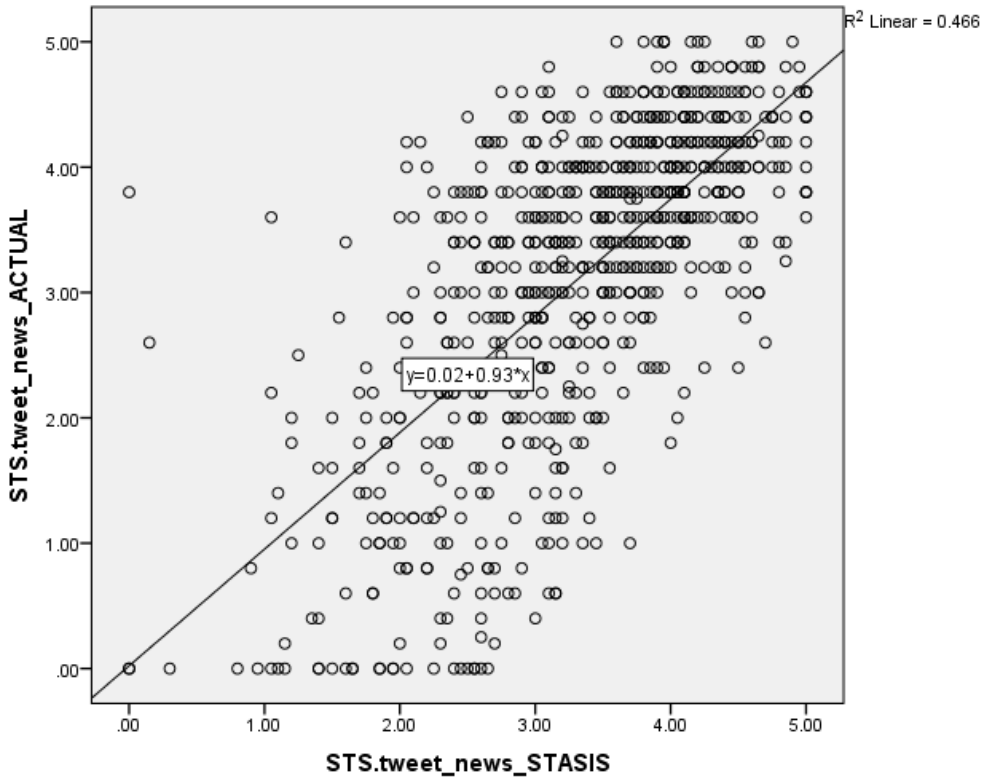

R2 Linear = 0.562
y=0.92+0.81*x
STS.tweet_news_ACTUAL
STS.tweet_news_JCN
5.00
4.00
3.00
2.00
1.00
.00
.00
1.00
2.00
3.00
4.00
5.00

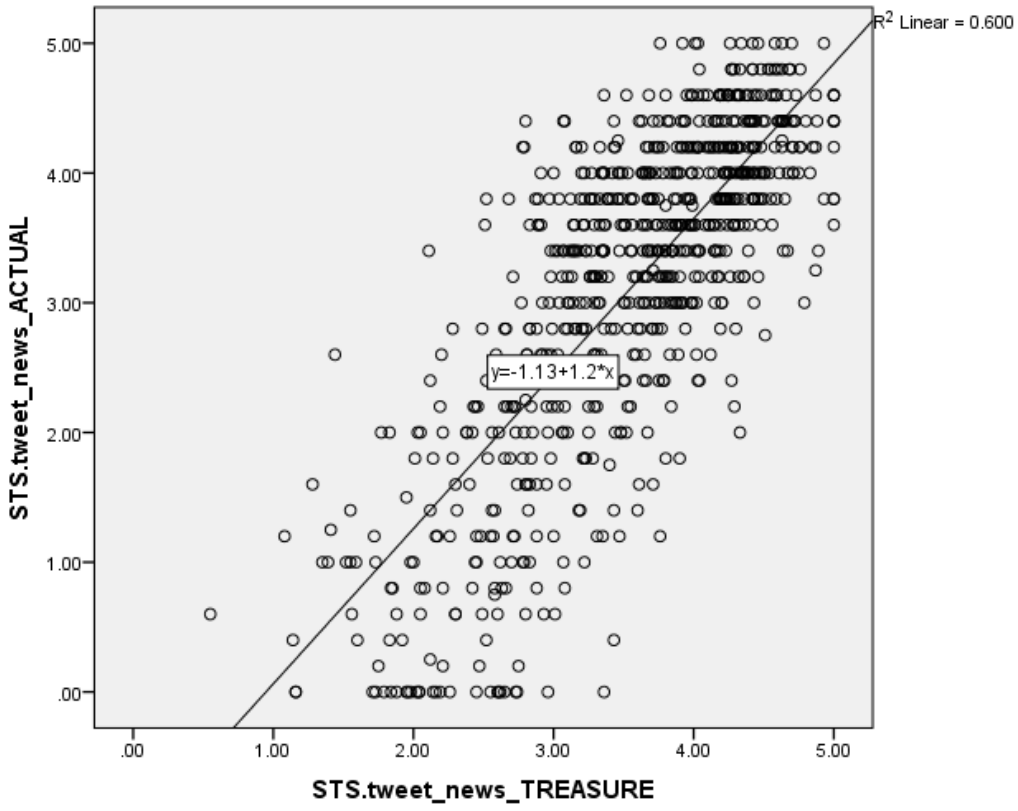

R2 Linear = 0.098
y=1.98+0.4*x
STS.tweet_news_ACTUAL
STS.tweet_news_RES
5.00
4.00
3.00
2.00
1.00
.00
.00
1.00
2.00
3.00
4.00
5.00


R2 Linear = 0.431
y=-5.38E-3+0.92*x
STS.tweet_news_ACTUAL
STS.tweet_news_LIN
5.00
4.00
3.00
2.00
1.00
.00
.00
1.00
2.00
3.00
4.00
5.00


R² Linear = 0.102
y=0.22+0.73*x
STS.tweet_news_ACTUAL
STS.tweet_news_LCH
5.00
4.00
3.00
2.00
1.00
.00
.00
1.00
2.00
3.00
4.00
5.00


R² Linear = 0.489
y=0.01+0.95*x
STS.tweet_news_ACTUAL
STS.tweet_news_WPATH
5.00
4.00
3.00
2.00
1.00
.00
.00
1.00
2.00
3.00
4.00
5.00


R² Linear = 0.466
y=0.02+0.93*x
STS.tweet_news_ACTUAL
STS.tweet_news_STASIS
5.00
4.00
3.00
2.00
1.00
.00
.00
1.00
2.00
3.00
4.00
5.00


R² Linear = 0.600
y=-1.13+1.2*x
STS.tweet_news_ACTUAL
STS.tweet_news_TREASURE
5.00
4.00
3.00
2.00
1.00
.00
.00
1.00
2.00
3.00
4.00
5.00

Appendix H – Author Publications

Cluster Analysis of Twitter Data: A Review of Algorithms

Noufa Alnajran, Keeley Crockett, David McLean and Annabel Latham
School of Computing, Math and Digital Technology, Manchester Metropolitan University
John Dalton Building, All Saints, Manchester, M1 5GD, U.K.
Noufa.alnajran@stu.mmu.ac.uk, (k.crockett, d.mclean, a.latham)@mmu.ac.uk

Keywords: Clustering, Social Network Analysis, Twitter, Data Mining, Machine Learning.

Abstract: Twitter, a microblogging online social network (OSN), has quickly gained prominence as it provides people with the opportunity to communicate and share posts and topics. Tremendous value lies in automated analysing and reasoning about such data in order to derive meaningful insights, which carries potential opportunities for businesses, users, and consumers. However, the sheer volume, noise, and dynamism of Twitter, imposes challenges that hinder the efficacy of observing clusters with high intra-cluster (i.e. minimum variance) and low inter-cluster similarities. This review focuses on research that has used various clustering algorithms to analyse Twitter data streams and identify hidden patterns in tweets where text is highly unstructured. This paper performs a comparative analysis on approaches of unsupervised learning in order to determine whether empirical findings support the enhancement of decision support and pattern recognition applications. A review of the literature identified 13 studies that implemented different clustering methods. A comparison including clustering methods, algorithms, number of clusters, dataset(s) size, distance measure, clustering features, evaluation methods, and results was conducted. The conclusion reports that the use of unsupervised learning in mining social media data has several weaknesses. Success criteria and future directions for research and practice to the research community are discussed.

1 INTRODUCTION

The rapid evolution of web 2.0 technologies such as OSN applications, has led to the continuous generation of an enormous volume of digital heterogeneous data being published at an unprecedented rate. These technologies have significantly changed the way people communicate and share information among each other in various domains. Millions of people have shifted from the traditional media channels such as newspapers, to online social media. In this context, Twitter has gained massive popularity as it provides an informal platform where people can easily publish and broadcast messages on different areas across the world. It had a prominent role in spreading awareness of natural disasters such as Hurricane Sandy and socio-political events such as the Arab Spring (Kumar et al., 2014). This has made Twitter an important source of information for synthesizing evidence in argumentation, and a goldmine of potential cross-domain opportunities for both businesses and decision makers. However, the exponential amount of user generated content on this site is too vast for manual analysis. More than 500 million short-text messages, referred to as "tweets", are published every day (Krestel et al., 2015). This requires an automated and scalable mining process to discover patterns in the unstructured data.

Cluster analysis is the unsupervised process of grouping data instances into relatively similar categories, without prior understanding of the groups structure or class labels (Han et al., 2011). It is a prominent component of exploratory data analysis. A subfield of clustering includes text mining, where large volumes of text are analysed to find patterns between documents (Godfrey et al., 2014). The growth of these unstructured data collections, advances in technology and computer power, and enhanced software capabilities, has made text mining an independent academic field. Moreover, the emergence of OSNs has yielded new frontiers for academic research, where researchers in the broad area of Natural Language Processing consider text analysis one of the most important research areas. Recent studies in various disciplines have shown increasing interest in micro-blogging services, particularly Twitter (Sheela, 2016). The

applications of text mining tools for studying features of content and semantics in tweets propagating through the network has been widely studied (Kumar et al., 2014). Several studies have aimed at analysing social data from Twitter through performing data mining techniques such as classification (Castillo et al., 2011). However, these techniques could be considered to have limited capabilities due to the unpredictable nature of the dataset. Cluster analysis of tweets has been reported to be particularly suitable for this kind of data for two reasons (Go et al., 2009): (1) the amount of data for training is too vast for manual labelling. (2) The nature of the data implies the existence of unforeseen groups that may carry important nuggets of information which can only be revealed by unsupervised learning.

Among the research conducted around clustering tweets' short-text and other text mining applications on Twitter, researchers aim to find relevant information such as inferring users' interests and identifying emergent topics. However, several natural challenges of the data prevent standard clustering algorithms being applied with their full potentials:

- Sparseness –unlike traditional clustering of documents which are rich in context, tweets are restricted to 140 characters.
- Non-standardization –people invented many ways to expand the semantics that are carried out by the tweet. This implies the usage of slangs, misspelled, and connected words. Users also use self-defined hashtags to identify topics or events.
- Volume –the rapid generation of tweets results in high volumes of data.

Therefore, due to the textual length restriction of the text, the content in tweets is limited, however it still may contain rich meanings. Therefore, tweets require intelligent techniques, such as incorporating semantic technologies that can analyse datasets with such complex characteristics and convey meanings and correlations.

The main purpose of this paper is to:

- Review various clustering algorithms that are implemented on different features of Twitter datasets.
- Review various domains of applications and success criteria that are used for measuring and evaluating the accuracy of the algorithm.
- Compare relevant approaches in terms of clustering methods, algorithms, number of clusters, dataset(s) size, distance measure, clustering features, evaluation methods, and results.
- Recommend future directions for research and practice to the research community.

To the best of our knowledge, there does not exist research that reviews the prominent clustering algorithms available to use on challenging, large, and unstructured data such as Twitter. Thus, this shall provide a thorough literature review and a valuable source of information on the state of the art for relevant research in this field.

The rest of this paper is organized as follows: section 2 describes the methods that are used in this review. Section 3 includes the techniques of mining Twitter datasets that use four clustering methods: (1) partition-based, (2) hierarchical-based, (3) hybrid-based, and (4) density-based. Section 4 contains the discussion and section 5 has the conclusion and future work. A table providing a summary of the studies featured in this review is located at the end of the paper.

2 METHODS

2.1 Literature Search Procedures

In this review, multiple research databases were investigated, such as Google Scholar and DeepDyve, to conduct online searches. This process includes searching for the following terms: 'mining Twitter short-text', 'clustering tweets', 'unsupervised learning on Twitter', and 'categorization of tweets'.

2.2 Inclusion Criteria

The inclusion criteria for this paper includes research that involve:

- An implementation of one of the following clustering methods: partition, hierarchical, hybrid, and density, on Twitter short-text messages. The reason for choosing these methods is that these generally cover the major clustering algorithms and have not been reviewed previously in the context of Twitter data.
- An approach to find hidden patterns and similar groups of information in tweets using models of unsupervised learning.

A total of 13 articles from 2011 to present have met the inclusion criteria as Twitter text mining applications using unsupervised learning.

3 CLUSTER-BASED TWITTER MINING

Many clustering methods exist in the literature, and it is difficult to provide a crisp categorization of these methods as they may overlap and share features. Nevertheless, the major clustering methods are included in this review (Han et al., 2011).

Clustering has been widely studied in the context of Twitter mining. It has been applied to analyse social behaviours in a variety of domains to achieve different tasks, such as tailoring advertisements for groups with similar interests (Friedemann, 2015), event detection (De Boom et al., 2015), and trending issues extraction (Purwitasari et al., 2015). This review focuses on the major clustering methods: partition, hierarchical, hybrid, and density, which have been used in the context of Twitter data.

3.1 Partition-Based Clustering

Partitioning algorithms attempt to organize the data objects into k partitions ($k \leq n$), each representing a cluster, where n is the number of objects in a dataset. Based on a distance function, clusters are formed such that objects within the cluster are similar (intra-similarity), whereas dissimilar objects lie in different clusters (inter-similarity). Partitioning algorithms can be further divided into hard and fuzzy (soft) clustering. In this section, six articles are summarized in which partitioning-based clustering algorithms has been applied in the exploratory analysis of Twitter.

3.1.1 Hard Clustering

Methods of hard partitioning of data assign discrete value label 0, 1, in order to describe the belonging relationship of objects to clusters. These conventional clustering methods provide crisp membership assignments of the data to clusters. K-means and k-medoids are the most popular hard clustering algorithms (Preeti Arora, 2016).

K-means is a centroid-based iterative technique which takes the number of representative instances, around which the clusters are built. Data instances are assigned to these clusters based on a dissimilarity function (i.e. distance measure). In each iteration, the mean of the assigned points to the cluster is calculated and used to replace the centroid of the last iteration until some criteria of convergence is met.

K-means has been adapted in numerous ways to suit different datasets including numerical, binary, and categorical features. In the context of Twitter mining applications, k-means approach for clustering customers of a company using social media data from Twitter was proposed (Friedemann, 2015). The technique constructs features from a massive Twitter dataset and clusters them using a similarity measure to produce groupings of users. The study performed k-means clustering and produced satisfactory experimental results. It is considered to be relatively computational efficient. In (Soni and Mathai, 2015), a 'cluster-then-predict' model was proposed to improve the accuracy of predicting Twitter sentiment through a composition of both supervised and unsupervised learning. After building the dataset, k-Means was performed such that tweets with similar words are clustered together. This unsupervised phase was performed after a feature extraction process. After the clustering phase, classification was done on the same data. The data was divided into training and testing sets, with 70% and 30% of the data respectively. Finally, the Random Forest learning algorithm was used for building the learning model, which was applied to each of the training datasets individually (Breiman, 2001). This algorithm has been chosen as it provides satisfactory trade-off between accuracy, interpretability, and execution time. Empirical evaluation shows that combining both supervised and unsupervised learning (k-Means then Random Forest) performed better than various stand-alone learning algorithms.

K-medoids is an object-based representative technique that deals with discrete data. It is an improvement to k-means in relation to its sensitivity to outliers. Instead of referring to the mean value of cluster objects, k-medoids picks the nearest point to the center of data points as the representative of the corresponding cluster. Thus, minimizing the sum of distances between each object, o, and its corresponding center point. That is, the sum of the error for all objects in each cluster is calculated as (Han et al., 2011),

$$E = \sum_{j=1}^{k} \sum_{p \in C_j} |p - o_j| \quad (1)$$

Where k is the number of clusters, p is an object in the cluster C_j, while o_j is the representative objects of C_j. The lower the value of E, the higher clustering quality.

A recent study focused on the usage of *k*-medoids algorithm for tweets clustering due to its simplicity and low computational time (Purwitasari et al., 2015). In this study, the author applied this algorithm to extract issues related to news that is posted on Twitter such as "flight passengers asking for refund" in Indonesia. Their proposed methodology for Twitter trending issues extraction consists of clustering tweets with *k*-medoids, in which they divided the tweets dataset into groups and used a representative tweet as the cluster center. Issue terms are then selected from the clusters result and assigned higher weight values. The terms that weigh over a certain threshold are extracted as trending issues. Weight score is calculated as the frequency of word occurrences in the dataset. Average Silhouette Width (Rousseeuw, 1987), a method for validating clusters' consistency, was used to measure and evaluate the clustering performance (Ramaswamy, [no date]). In the work, the experiments demonstrated good results of using *k*-medoids for this purpose, however, re-tweets (i.e. duplicates) had influenced the clustering results. Another study used *k*-means and *k*-medoids respectively to cluster a single Twitter dataset and compare the results of each algorithm (Zhao, 2011). Initially, *k*-means was applied, which took the values in the matrix as numeric, and set the number of clusters, *k*, to eight. After that, the term-document matrix was transformed to a document-term one and the clustering was performed. Then, the frequent words in each cluster and the cluster centers were computed in order to find what they are about. The first experiment showed that the clusters were of different topics. The second experiment was conducted using *k*-medoids, which used representative objects instead of means to represent the cluster center.

K-medoids has the advantage of robustness over *k*-means as it is less influenced by noise and outliers. However, this comes at the cost of efficiency. This is due to the high processing time that is required by *k*-medoids compared to *k*-means. Both methods require the number of clusters, *k*, to be fixed. In terms of clustering sparse data such as tweets, *k*-medoids may not be the best choice as these do not have many words in common and the similarities between them are small and noisy (Aggarwal and Zhai, 2012). Thus, a representative sentence does not often contain the required concepts in order to effectively build a cluster around it.

3.1.2 Fuzzy Clustering

This partition-based method is particularly suitable in the case of no clear groupings in the data set. Unlike hard clustering, fuzzy algorithms assign a continuous value [0, 1] to provide reasonable clustering. Multiple fuzzy clustering algorithms exist in the literature, however fuzzy *c*-means (FCM) is the most prominent.

FCM provides a criteria on grouping data points into different clusters to varying degrees that are specified by a membership grade. It incorporates a membership function that represents the fuzziness of its behaviour. The data are bound to each cluster by means of this function.

In the context of Twitter analysis, a recent study presented a simple approach using fuzzy clustering for pre-processing and analysis of hashtags (Zadeh et al., 2015). The resulting fuzzy clusters are used to gain insights related to patterns of hashtags popularity and temporal trends. To analyse hashtags' dynamics, the authors identified groups of hashtags that have similar temporal patterns and looked at their linguistic characteristics. They recognised the most and least representative hashtags of these groups. The adopted methodology is fuzzy clustering based and multiple conclusions were drawn on the resulting clusters with regards to variations of hashtags throughout a period of time. Their clustering was based on the fact that categorization of hashtags is not crisp, rather, most data points belong to several clusters according to certain degrees of membership. Another study compared the performance of supervised learning against unsupervised learning in discriminating the gender of a Twitter user (Vicente et al., 2015). Given only the unstructured information available for each tweet in the user's profile, the aim is to predict the gender of the user. The unsupervised learning involved the usage of soft in conjunction with hard clustering algorithms. *K*-means and FCM were applied on a 242K Twitter users' dataset. The unsupervised approach based on FCM proved to be highly suitable for detecting the user's gender, achieving a performance of about 96%. It also has the privilege of not requiring a labelled training set and the possibility of scaling up to large datasets with improved accuracy.

Experiments have shown that fuzzy-based clustering is more complex than clustering with crisp boundaries. This is because the former requires more computation time for the involved kernel (Bora et al., 2014). Fuzzy methods provide relatively high clustering accuracy and more realistic probability of belonging. Therefore, they can be considered an effective method that excludes the need of a labelled

dataset. This is particularly useful for sheer volumes of tweets, where human annotations can be highly expensive. However, these methods generally have low scalability and results can be sensitive to the initial parameter values. In terms of optimization, fuzzy clustering methods can be easily drawn into local optimal.

3.2 Hierarchical-Based Clustering

In hierarchical clustering algorithms, data objects are grouped into a tree like (i.e. hierarchy) of clusters. These algorithms can be further classified depending on whether their composition is formed in a top-down (divisive) or bottom-up (agglomerative) manner. This section reviews three studies that performed hierarchical-based clustering algorithms in applications of Twitter mining.

Hierarchical clustering was used for topic detection in Twitter streams, based on aggressive tweets/terms filtering (Ifrim et al., 2014). The clustering process was performed in two phases, first the tweets and second the resulting headlines from the first clustering step. Their methodology is composed of initially computing tweets pair-wise distances using the cosine metric. Then computing a hierarchical clustering so that tweets belonging to the same topic shall cluster together, and thus each cluster is considered as a detected topic. Afterwards, they controlled the tightness of clusters by cutting the resulting dendrogram at 0.5 distance threshold. In this way they will not have to provide the number of required clusters a-priori as in *k*-Means. The threshold was set to 0.5 in order to avoid having loose or tight clusters, rather, a value of 0.5 worked well for their method. Each resulting cluster is then assigned a score and ranked according to that score. The top-20 clusters are then assigned headlines, which are the first tweet in each of them (with respect to publication time). The final step involved re-clustering the headlines to avoid topic fragmentation, also using hierarchical clustering, the resulting headlines are then ranked by the one with the highest score inside a cluster. The headlines with the earliest publication time are selected and their tweet text is presented as a final topic headline. Another research implemented a hierarchical approach for the purpose of helping users parse tweets results better by grouping them into clusters (Ramaswamy, [no date]). The aim was for fewer clusters that are tightly packed, rather than too many large clusters. The work involved using a dataset of tweets to see how the choice of the distance function affects the behaviour of hierarchical clustering algorithms. Ramaswamy conducted a survey of two clustering algorithms that are both hierarchical in nature but different in their core implementation of the distance function has been conducted. A total of 925 tweets comprising of various topics with common keyword have been used in the experiments. In the first algorithm, the author considered each of the given objects to be in different clusters. Then determining if the object o is close enough to cluster c, and if so, add o to c. This process continues until the maximum size of the desired clusters is reached or no more new clusters can be formed. In this first algorithm, the notion of the distance between an object and a cluster has been defined using concepts from association rule problems –support and confidence. The second algorithm maintained the average distance of an object from each element in the cluster as the similarity measure. If the average is small enough, the object is added to the cluster. Both clustering algorithms were implemented using C# and involve reading the tweets, tokenizing them, clustering them and returning the clustered output. Although the overall behaviour was found to be similar for both algorithms, the second one seemed to fare better for each of the confidence and support level value. An integrated hierarchical approach of *agglomerative* and *divisive* clustering was proposed to dynamically create broad categories of similar tweets based on the appearance of nouns (Kuar, 2015). The bottom-up approach merges similar clusters together to reduce their redundancy. The technique adopted a recursive and incremental process of dividing and combining clusters in order to produce more meaningful sorted clusters. It has shown an increase in clustering effectiveness and quality compared to standard hierarchical algorithms.

In this context, empirical evaluations provided that hierarchical methods performed slower than hard partition-based clustering, particularly *k*-means (Manpreet Kaur, 2013). Therefore, for massive social media datasets, hard partitioning methods are considered to be relatively computationally efficient as well as producing acceptable experimental results.

3.3 Hybrid-Based Clustering

Because hierarchical clustering algorithms tend to compare all pairs of data, their robustness is relatively high. However, this makes them not very efficient due to their tendency to require at least $O(n^2)$ computation time. On the other hand, partitioning algorithms may not be the optimal choice despite being more efficient than hierarchical

algorithms. This is because the former may not be very effective as they tend to rely on small number of initial cluster representatives. This trade-off has led researchers to propose several clustering algorithms that combined the features of hierarchical and partitioning methods in order to improve their performance and efficiency. These hybrid algorithms include any aggregations between clustering algorithms. In general, they initially partition the input dataset into sub clusters and then construct a new hierarchical cluster based on these sub clusters.

There is not much research conducted using a hybrid clustering approach in the area of Twitter mining. Nevertheless, one approach implemented clustering of keywords that are presented in the tweets using agglomerative hierarchical clustering and crisp *c*-means (Miyamoto et al., 2012). The clustering features was based on a series of tweets as one long sequence of keywords. The approach involved building two datasets, each composed of 50 tweets in different timeframes. Several observations of agglomerative clusters obtained by cutting the dendrogram and *c*-means clusters, with and without pair-wise constrains were analysed. Better clustering results are provided using pair-wise constrains, however, the size of datasets is relatively small for a generalization.

3.4 Density-Based Clustering

This method groups data located in the region with high density of the data space to belong to the same cluster. Therefore, it is capable of discovering clusters with arbitrary shape. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is the prominent density-based algorithm. It grows regions with sufficiently high density into clusters (Ester et al., 1996). In this section, three articles are summarized in which density-based algorithms has been applied in the exploratory analysis of Twitter.

A density-based clustering has been adopted in the context of Twitter textual data analysis to discover cohesively the information posted by users about an event as well as the user's perception about it (Baralis et al., 2013). The provided framework adopts a multiple-level clustering strategy, which focuses on disjoint dataset portions iteratively and identifies clusters locally. DBSCAN has been exploited for the cluster analysis as it allows discovering arbitrarily shaped clusters, and increases cluster homogeneity by filtering out noise and outliers. Additionally, it does not require prior specification of the number of expected clusters in the data. In this approach, DBSCAN has been applied iteratively on disjoint dataset portions and all the original dataset is clustered at the first level. Then, tweets labelled as outliers in the previous level are re-clustered at each subsequent level. To discover representative clusters for their Twitter dataset, they attempt to avoid clusters containing few tweets. They also attempt to limit the number of tweets labelled as outliers and thus un-clustered, in order to consider all different posted information. Through addressing these issues, DBSCAN parameters were properly set at each level. A recent study employed DBSCAN as part of its novel method for creating an event detection ground truth through utilizing tweets hashtags (De Boom et al., 2015). The authors clustered co-occurring hashtags using DBSCAN. The method required setting two thresholds: the minimum number of hashtags per cluster and a minimum similarity measure between two hashtags, above which the two hashtags belong to the same neighbourhood. A collection of clusters of sufficiently co-occurring hashtags on the same day were obtained by running DBSCAN for every day in the dataset. A recent study has introduced the application of DBSCAN for representing meaningful segments of tweets in batch mode (Anumol Babu, 2016). The segmentation was done based on calculations of the stickiness score. This score considers the probability of a segment being a phrase within the batch of tweets (i.e. local context) and the probability of it being a phrase in English (i.e. global context) (Weng et al., 2015). Sentimental variations in tweets were then analysed based on these segments. Each word in the text is assigned a sentiment score according to a predetermined sentiment lexicon. The sentiment of a tweet is then denoted as the summation of the most positive score and the most negative score among individual words in the tweet. In this approach, the core of the clustering consisted of integrating DBSCAN with Jaccard Coefficient similarity function. Empirical evaluations indicated an enhancement of the existing system as a result of using DBSCAN for clustering,

It can be observed from the literature surrounding Density-based algorithms in Twitter mining, that they are highly efficient and can be particularly suitable for clustering unstructured data, such as tweets, as it allows the identification of clusters with arbitrary shape. Moreover, it is less prone to outliers and noise, and does not require initial identification of the required number of clusters. However, clustering high data volumes requires big memory size.

4 DISCUSSION

Several approaches of unsupervised learning applications for mining unstructured social media data have been reviewed and presented in table 1. The table provides a comparison on the features that are used in the studies including: research approach, clustering method, algorithm, number of clusters, dataset size, distance measure, clustering features, evaluation methods, and results. The review comprises 13 studies spanning from 2011 to the present. These studies have different approaches, in which the clustering of Twitter data was performed in various settings and domains to achieve different business values or satisfy certain requirements. These approaches range from pure clustering perspectives, such as determining the impact of a distance function choice on clustering behaviour, to a more general pattern recognition application, such as targeting advertisements and events detection. The majority of the studies performed clustering in order to detect news, topics, events, and facts and to predict sentiments. Different clustering methods and algorithms were implemented in these studies, each of different dataset and number of clusters. From the 13 reviewed datasets, it can be observed that the average dataset size is 162,550 for tweets textual data, ranging from 50 to 1,084,200 and average of 126,329 for Twitter user accounts, ranging from 10,000 to 242,658 distinct user accounts. The majority of the dataset sizes observed in the surveys are relatively small, which means that the high volume challenge of Twitter data has not been taken into consideration. Therefore, in order for these algorithms to be effective, they should be able to scale well to the massive amounts of Twitter data. In this matter, the scalability (in terms of clustering performance) of most of the algorithms implemented in the surveys is questionable as these algorithms have not been tested on considerably large datasets.

As partitioning algorithms require the number of clusters, *c*, to be pre-set, *c* has been included in the review to provide an indication on the number of clusters that might be appropriate for similar tasks. From the provided comparisons, the average number of clusters maintained can be derived, which is 7, with 2 as the minimum clusters and 10 as the maximum. The table additionally compared the different distance measures used. It can be observed that Euclidean distance is the prominent for partitioning algorithms, whereas hierarchical algorithms commonly implemented the cosine similarity measure. In terms of clustering features, different sets were used depending on the implemented approach. The features observed from the review include some or all of the following:

- Hashtags –31% of the reviewed surveys included hashtags in the features set and considered their impact, 23% treated hashtags as normal words in the text, and 31% removed hashtags before analysis (excluding the 15% studies that are clustering upon user accounts).
- Account metadata –username, date, status, latitude, longitude, followers, and account followings.
- Tweet metadata –tweet id, published date, and language.
- Maintaining a BOW of the unique words contained in each textual data of a tweet and their frequencies as the feature vector. Some included hashtags in the BOW while others ignored them.

None of the surveys studied the impact of retweets nor "@mentions". Rather, some datasets did not remove the retweeted tweets which affected the resulting clustering credibility. Because tweets commonly get large number of retweets, keeping them in the dataset will produce large clusters containing redundant tweets rather than tweets with similar features. This will consequently reinforce false patterns and increase run time.

Evaluation methods vary from robust measures, such as ASW to manual observations, such as manually comparing an algorithm's detected topics with Google news headlines. ASW has been utilised by most of the studies to measure the clustering performance. Some of the evaluation methods are derived from other data mining techniques such as association rules and classification. These methods include clustering based on confidence and support levels, and calculating precision, recall and the F measure from a confusion matrix.

5 CONCLUSION

The review contributes to the literature in several significant ways. First, it provides a comparative analysis on applications that utilized and tuned text mining methods, particularly clustering, to the characteristics of Twitter unstructured data. Second, the review concentrated on algorithms of the general clustering methods: (1) partition-based, (2) hierarchical-based, (3) hybrid-based, and (4) density-based, in Twitter mining. Third, unlike existing reviews which provides high level and

abstract specification of surveys, this review was comprehensive in that it provided comparative information and discussion across the dataset size, approach, clustering methods, algorithm, number of clusters, distance measure, clustering feature, evaluation methods, and results.

Thirteen articles were reviewed in this paper, and the results indicated that there is a sufficient improvement in the exploratory analysis of social media data. However, many of the existing methodologies have limited capabilities in their performance and thus limited potential abilities in recognising patterns in the data:

- Most of the dataset sizes are relatively small which is not indicative of the patterns in social behaviours and therefore generalised conclusions cannot be drawn. Because of the sparsity of Twitter textual data, it is difficult to discover representative information in small datasets. Therefore, future studies should aim to increase the size of the dataset.
- Some of the algorithms implemented may have provided effective results in terms of efficiency and accuracy. However, this may be attributed to the small size of dataset as the scalability has not been evaluated.
- Some of the reviewed datasets included redundant tweets (i.e. retweets) which yields inaccurate clustering. Therefore future studies should perform a comprehensive pre-processing phase in which retweets and other noise, such as URLs, are removed from the dataset prior to clustering.
- Most of the studies implemented keyword-based techniques, such as term frequencies and BOW which ignores the respective order of appearance of the words and does not account for correlations between text segments. Therefore, future research should incorporate and measure the underlying semantic similarities in the dataset.

In conclusion, after conducting this review it can be clearly noticed that clustering is an important element of exploratory text analysis in which unstructured data can be useful for pattern recognition as well as identification of user potentials and interests. However, future research must demonstrate the effectiveness of such approaches through acquiring larger datasets in order for the algorithms to be useful in discovering knowledge and applicable in several contexts and domains. A meta-analysis review is recommended as a future work, which will provide a quantitative estimate for the impact and usefulness of clustering methods in providing insights from social media data.

Table 1: Summary of the studies featured in this review

Author & Year	Approach	Method		Algorithm & Number of Clusters (C)	Dataset Size	Distance Measure	Clustering Features	Evaluation Methods	Results
(Friedemann, 2015)	Targeting advertise-ments	Partitioning-Based Clustering	Hard Partitioning	*k*-Means C: 5	10,000 Twitter user account	Euclidean distance	posted status, number of followers and account followings, latitude, longitude, whether a popular Twitter account (*influencer*) is followed	Computing a metric of clustering quality *q*. The lower the value of *q*, the better clustering performance	Achieved clustering is midway between ideal and randomized data. Experiments emphasized the credibility of Twitter data for market analysis
(Soni and Mathai, 2015)	Sentiment prediction			*k*-Means C: 2	1200 "Apple" tweets	Squared Euclidean distance	Bag-of-Words (BOW) from twitter corpus (frequency of word occurrences)	Confusion matrix and ROC (Receiver Operator Characteristic) graph	Model integration of supervised and unsupervised *k*-Means learning improved twitter sentiment prediction
(Purwitasari et al., 2015)	News summary			*k*-Medoids C: 10	200 tweets (geo-location: Indonesia)	Cosine similarity	Term frequencies and weight in tweet text. Hashtags omitted	The larger ASW value, the more homogeneous the cluster result	Inclusion of retweets affected cluster result quality
(Zhao, 2011)	R Data Mining			*k*-Means C: 8	1st 200 tweets from @rdatamining account	Euclidean distance	Term frequencies in tweet text (document-term matrix). Hashtags omitted	Checked the top 3 terms in every cluster	Clusters of different topics
				k-Medoids C: 9		Manhattan distance		ASW	Clusters overlap and not well separated
(Vicente et al., 2015)	Gender detection		Fuzzy Partitioning	*k*-Means C: 2	242,658 unique Twitter users	Euclidean distance	Screen name and user name	Two experiments: 1st used labelled data for building clusters and evaluating performance. 2nd used unlabelled data for clustering and labelled for evaluation	*C*-Means provided better clustering performance than *k*-Means. More usage of unlabelled data significantly enhanced *c*-means but got *k*-Means worse
				c-Means C: 2					
(Zadeh et al., 2015)	Events and facts detection			FANNY (Kaufman and Rousseeuw, 2009) C: 6	40 distinct hashtag	Manhattan Distance	Temporal aspects of hashtags	Defined a *misfit* measure to identify elements' degree of "not fitting" into a cluster. Clustering performance measured using ASW	Insights into patterns associated with each cluster for hashtags changing popularities over time

Table 1: (Continued)

Author and Year	Approach	Method	Algorithm and Number of Clusters (C)	Dataset Size	Distance Measure	Clustering Features	Evaluation Methods	Results
(Ifrim et al., 2014)	Topic detection	Hierarchical-Based Clustering	Agglomerative (dendrogram cut at 0.5)	1st dataset: 1,084,200 tweets. 2nd: 943,175 JSON format English tweets	Cosine similarity	Date, tweet id, text, user mentions, hashtags, URLs, media URLs, and retweet or not	(1) a subset of ground truth topics, (2) google for the automatically detected topic headline, in the manual assessment of how many detected topics are actually published news in traditional media	Application of agglomerative clustering can detect topics with 80% accuracy. However, not efficient for real-time data analysis.
(Ramaswamy, [no date])	Impact of distance function choice on clustering behaviour		Two Ward (Jr., 1963) algorithms C: 5	925 tweets	Ratio of tweets appearing in different clusters / Avg. distance between tokens and clusters	Tokenization of tweets' texts	Several experiments conducted to determine appropriate values of confidence and support levels which determine further clustering	Generally similar behaviour of the 2 algorithms. In terms of fewer, tightly packed clusters, 2nd algorithm fared better for confidence and support values
(Kaur, 2015)	Noun-based tweet categorization		Agglomerative / Divisive	15062 "Stem Cell" tweets	Inter-cosine similarity / Intra-cosine similarity	Frequency of occurrences for nouns in tweets. Hashtags omitted	Experimental comparisons of clustering quality against: k-means, Ward, and DBSCAN clustering.	Combinatorial approach provided higher accuracy compared to existing methodologies, however, at the cost of performance. Clustering runtime: 1hour
(Miyamoto et al., 2012)	Keyword clustering	Hybrid-based Clustering	Hard *c*-Means (partitioning) C: 2 / Agglomerative (hierarchical) C: 2	1st dataset: 50 tweets (35 terms occur > 8 times) 2nd: 50 tweets (38 terms occur > 5 times)	Squared Euclidean distance	Sequence of word occurrences in a set of tweets	Several observations of: clusters with and without pair-wise constraints clusters obtained by cutting the dendrogram with and without pair-wise constraints	Application of pair-wise constraints improved clustering quality. However, dataset size is arguably small
(Baralis et al., 2013)	Cohesive information discovery	Density-Based Clustering	DBSCAN	"Paralympics" dataset: 1969 tweets / "Concert" dataset: 2960 tweets	Cosine similarity	BOW of tweets including hashtags	ASW	Effective in discovering knowledge. Performance relatively low for not very large dataset. Clustering runtime: 2min 9sec May not scale well to massive datasets
(De Boom et al., 2015)	Event detection		DBSCAN	63,067 tweets (geolocation: Belgium)	Sum of avg. occurrences of both hashtags per day/2	Hashtags co-occurrence matrix	Precision, recall, and F measures	Improvement in event detection and clustering through high-level semantic information
(Anumol Babu, 2016)	Sentiment Analysis		DBSCAN	100 synthetic tweets	Jaccard similarity	Tweet text and publication time. Hashtags omitted	Evaluating tweets segmentation and its accuracy through an experiment	Enhancement of the present system as DBSCAN was integrated

REFERENCES

AGGARWAL, C. C. & ZHAI, C. 2012. *Mining text data*, Springer Science & Business Media.

ANUMOL BABU, R. V. P. 2016. Efficient Density Based Clustering of Tweets and Sentimental Analysis Based on Segmentation. *International Journal of Computer Techniques,* 3, 53-57.

BARALIS, E., CERQUITELLI, T., CHIUSANO, S., GRIMAUDO, L. & XIAO, X. Analysis of twitter data using a multiple-level clustering strategy. International Conference on Model and Data Engineering, 2013. Springer, 13-24.

BORA, D. J., GUPTA, D. & KUMAR, A. 2014. A comparative study between fuzzy clustering algorithm and hard clustering algorithm. *arXiv preprint arXiv:1404.6059*.

BREIMAN, L. 2001. Random forests. *Machine learning,* 45, 5-32.

CASTILLO, C., MENDOZA, M. & POBLETE, B. Information credibility on twitter. Proceedings of the 20th international conference on World wide web, 2011. ACM, 675-684.

DE BOOM, C., VAN CANNEYT, S. & DHOEDT, B. Semantics-driven event clustering in twitter feeds. Making Sense of Microposts, 2015. CEUR, 2-9.

ESTER, M., KRIEGEL, H.-P., SANDER, J. & XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd, 1996. 226-231.

FRIEDEMANN, V. 2015. Clustering a Customer Base Using Twitter Data.

GO, A., BHAYANI, R. & HUANG, L. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford,* 1, 12.

GODFREY, D., JOHNS, C., MEYER, C., RACE, S. & SADEK, C. 2014. A case study in text mining: Interpreting twitter data from world cup tweets. *arXiv preprint arXiv:1408.5427*.

HAN, J., PEI, J. & KAMBER, M. 2011. *Data mining: concepts and techniques*, Elsevier.

IFRIM, G., SHI, B. & BRIGADIR, I. Event detection in twitter using aggressive filtering and hierarchical tweet clustering. Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014, 2014. ACM.

JR., J. H. W. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association,* 58, 236-244.

KAUFMAN, L. & ROUSSEEUW, P. J. 2009. *Finding groups in data: an introduction to cluster analysis*, John Wiley & Sons.

KAUR, N., 2015. *A Combinatorial Tweet Clustering Methodology Utilizing Inter and Intra Cosine Similarity* (Doctoral dissertation, Faculty of Graduate Studies and Research, University of Regina).

KRESTEL, R., WERKMEISTER, T., WIRADARMA, T. P. & KASNECI, G. Tweet-Recommender: Finding Relevant Tweets for News Articles. Proceedings of the 24th International Conference on World Wide Web, 2015. ACM, 53-54.

KUMAR, S., MORSTATTER, F. & LIU, H. 2014. *Twitter data analytics*, Springer.

MANPREET KAUR, U. K. 2013. Comparison Between K-Mean and Hierarchical Algorithm using Query Redirection. *International Journal of Advanced Research in Computer Science and Software Engineering* 3, 54-59.

MIYAMOTO, S., SUZUKI, S. & TAKUMI, S. Clustering in tweets using a fuzzy neighborhood model. Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on, 2012. IEEE, 1-6.

PREETI ARORA, D. D., SHIPRA VARSHNEY. Analysis of K-Means and K-Medoids Algorithm For Big Data. 2016 India. Procedia Computer Science, 507-512.

PURWITASARI, D., FATICHAH, C., ARIESHANTI, I. & HAYATIN, N. K-medoids algorithm on Indonesian Twitter feeds for clustering trending issue as important terms in news summarization. Information & Communication Technology and Systems (ICTS), 2015 International Conference on, 2015. IEEE, 95-98.

RAMASWAMY, S. Comparing the Efficiency of Two Clustering Techniques.

ROUSSEEUW, P. J. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics,* 20, 53-65.

SHEELA, L. 2016. A Review of Sentiment Analysis in Twitter Data Using Hadoop. *International Journal of Database Theory and Application,* 9, 77-86.

SONI, R. & MATHAI, K. J. 2015. Improved Twitter Sentiment Prediction through Cluster-then-Predict Model. *arXiv preprint arXiv:1509.02437*.

VICENTE, M., BATISTA, F. & CARVALHO, J. P. Twitter gender classification using user unstructured information. Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on, 2015. IEEE, 1-7.

WENG, J., LI, C., SUN, A. AND HE, Q., 2015. Tweet Segmentation and its Application to Named Entity Recognition.

ZADEH, L. A., ABBASOV, A. M. & SHAHBAZOVA, S. N. Analysis of Twitter hashtags: Fuzzy clustering approach. Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC), 2015 Annual Conference of the North American, 2015. IEEE, 1-6.

ZHAO, Y. 2011. R and Data Mining: Examples and Case Studies.

An Empirical Performance Evaluation of Semantic-Based Similarity Measures in Microblogging Social Media

Noufa N. Alnajran, Keeley A. Crockett, David McLean, Annabel Latham
Department of Computing, Mathematics, and Digital Technology
Manchester Metropolitan University
Noufa.alnajran@stu.mmu.ac.uk,{k.crockett, d.mclean, a.latham}@mmu.ac.uk

***Abstract*—Measuring textual semantic similarity has been a subject of intense discussion in NLP and AI for many years. A new area of research has emerged that applies semantic similarity measures within Twitter. However, the development of these measures for the semantic analysis of tweets imposes fundamental challenges. The sparsity, ambiguity, and informality present in social media are hampering the performance of traditional textual similarity measures as "tweets", have special syntactic and semantic characteristics. This paper reviews and evaluates the performance of topological, statistical, and hybrid similarity measures, in the context of Twitter analysis. Furthermore, the performance of each measure is compared against a naïve keyword-based similarity computation method to assess the significance of semantic computation in capturing the meaning in tweets. An experiment is designed and conducted to evaluate the different measures through examining various metrics, including correlation, error rates, and statistical tests on a benchmark dataset. The potential weaknesses of semantic similarity measures in relation to Twitter applications of textual similarity assessment and the research contributions are discussed. This research highlights challenges and potential improvement areas for the semantic similarity of tweets, a resource for researchers and practitioners.**

Keywords*— *statistical semantics, semantic similarity, online social network analysis, text similarity, Twitter, WordNet

I. INTRODUCTION

Short Text Semantic Similarity (STSS) measures are employed for measuring the degree to which short-texts are subjectively evaluated by humans as being semantically equivalent to each other [1]. Short-texts refer to typical human utterances that are of sentence length ranging from 10 to 25 words [2]. Human generated sentences are prone to forms of text that do not conform to typical grammatical and syntactical rules of a sentence. O'Shea et al. [2] suggested that semantic similarities of these short-texts can be measured through the application of STSS measures. These measurements are gaining prominence as much research in the field of natural language processing (NLP) and artificial intelligence (AI) are emerging in multiple domains. The task of assessing the semantic similarity between short-texts has been a central problem in NLP, due to its importance in a variety of applications. Some of the earliest text similarity applications have been implemented for text classification and information retrieval [3], automatic word sense disambiguation [4], and extractive text summarization [5]. More recent applications of STSS include the incorporation of the measure in a conversational agent to reduce the time associated with the scripting process [6], measuring the similarity between documents [7], and in supervised learning and text classification [8]. Measuring semantic similarity can be performed at various levels, ranging from words, phrases and sentences, to paragraphs and documents. Each of these categories employ different methods and techniques to gauge the underlying meaning at that particular level.

A. Problem Statement

In this paper, the focus is on semantic similarity measures at the short text level. The challenges in determining the degree of semantic equivalence between sentences is attributed to the variations in natural language expressions. In natural languages, a single meaning of a sentence can be expressed in many ways, and therefore the task of measuring the semantic similarity of natural language sentences is very complex. This problem is more prevalent in Online Social Network (OSN) texts due to the informal nature and the high degree of lexical variations used. Areas of work within related fields, such as classification and clustering of tweets face similar issues when identifying similarities in natural language text presented in Twitter [9]. To illustrate some challenges present in Twitter, consider the following tweet [10]: "#qcpoli enjoyed a hearty laugh today with #plq debate audience for @jflisee #notrehome tune was that the intended reaction?" The presence of symbols, spelling mistakes, letter repetitions, e.g. "@jflisee", and abbreviations complicate the process of tokenization and Part-of-Speech [11] tagging required by text analysis tasks. Little research has been conducted in the area of semantic analysis of Twitter data especially in relation to semantically measuring the degree of equivalence between tweets. This may be attributed to the characteristics of such data that make the task significantly more difficult than analyzing general short-text. However, several studies highlighted the potential and significance of developing semantic similarity measures [12] and paraphrase identification techniques [13], [14] specifically for tweets. In the context of Twitter, semantic similarity measures are particularly useful in reducing the challenge of high redundancy and the sparsity inherent in its data. One of the possible approaches to reduce the complexity of dealing with massive data is through integration of these measures in applications of Machine Learning.

This paper addresses the problem of STSS applicability in

the context of Twitter short text messages. As these messages share special lexical and syntactical characteristics, traditional STSS measures, which analyse proper English sentences fail to capture the semantic similarities between these messages. Therefore, this paper sets out to review and empirically evaluate different approaches to STSS measures to compare their performance on a labelled dataset of tweets. This is particularly important for research aiming to adapt or develop new STSS measures that consider the different sorts of noise present in social media data.

B. Research Questions

The paper aims to answer the following research questions:

RQ1. *Which approaches exist that support the identification of semantic similarity between Twitter short text messages?*

RQ2. *What are the challenges present in the language used in Twitter that hinder an effective process of semantic similarity identification?*

RQ3. *How do different kinds of STSS measures perform in relation to human assessments for Twitter short-text Messages?*

C. Contributions and Outline

In this paper, topological-based and statistical-based STSS measures are reviewed and evaluated in terms of performance. Towards accomplishing this purpose, the research investigated in this paper has the following objectives:

1) Provide an overview of the different approaches that can be adapted for identifying sentence-based semantic similarities.
2) Highlight the challenges of the natural language used in Twitter that hamper the performance of semantic similarity measures.
3) Evaluate and compare the performance of various STSS measures in applications of Twitter short text messages.

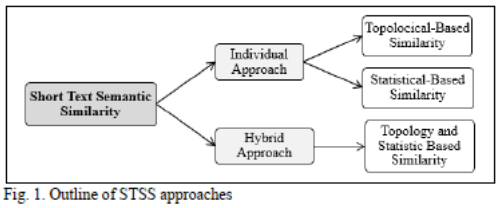


Fig. 1. Outline of STSS approaches

A hybrid semantic similarity is a more recent approach which is composed of a combination of different implementations of STSS measures. The resource of integrated information provided in this paper shall provide insights on the relevant issues and perspectives that should be considered in future proposals, and therefore facilitate the development of future works that aim to contribute to the field of Twitter NLP and social media analysis. Fig. 1 summarizes the similarity approaches studied in this paper.

The remainder of the paper is organized as follows: Section II describes the methods that are used in the review part. Section III describes the three categories of STSS measures under consideration. Section IV discusses the challenges presented in Twitter that hinder the performance of these measures and observations derived from the reviewed approaches. Section V explains the experimental methodology in terms of design, hypothesis, dataset and sample size, feature set, and experiment analysis and evaluation metrics. In section VI, the experiment results and analysis using correlations, Mean Squared Error (MSE) and inferential statistical analysis are presented and explained. Section VII discusses the experiment results and observations taking into consideration the current settings in which the experiment took place. Finally, the conclusion and further directions are provided in Section VIII.

II. Methods

A. Inclusion Criteria

The inclusion criteria for the contributions reviewed in this research are as follows:

1) Contributions to enhance the semantic textual analysis of Twitter short text messages through the development of semantic similarity measures.
2) Contributions to determine latent topics in textual data obtained from Twitter through potential semantic similarity processes for topic modelling, such as Latent Dirichlet Allocation (LDA), which is further elaborated in Section III.B.

III. Short Text Semantic Similarity Measures

STSS measures are generally divided, in terms of their core functionality and attributes, into three categories: topological, statistical, and hybrid.

A. Topology-Based STSS

The semantic similarity between short-texts can be gauged through defining a topological similarity, which is based on using knowledge bases such as ontologies. The distance between terms and concepts are determined by means of these resources. Calculating the topological similarity between ontological concepts can be done either by using the edges and their types (edge-based) or the nodes and their properties (node-based) as data sources. Liu and Wang [15] presented a topological measure for computing the semantic similarity between short texts based on the structural and semantic relationships in a predefined hierarchical concept tree (HCT), without requiring any additional corpus information. A major drawback of this approach is that it does not take into account the word's sequence in which it appears in the sentence. For instance, the sentences *the cat chased the dog* and *the dog chased the cat* would be considered identical.

Another drawback is related to the scalability and performance of the current state-of-the-art semantic measures libraries. The authors in [16] argue that these

drawbacks are due to using naïve graph representation models, which fail to capture the intrinsic structure of the represented taxonomies. Consequently, topological algorithms that are based on naïve models suffer from degraded performance due to demanding high computational cost. This complexity problem is derived from the caching strategy adopted by current semantic measures libraries. This strategy stores all nodes' ancestors and descendants within the taxonomy, which significantly increases memory usage leading to scalability problems concerning the taxonomy size. Moreover, the dynamic resizing of the caching data structures, further memory allocation, or the integration with external relational databases will raise performance issues.

Current state-of-the-art is a new representation model for taxonomies, along with a new software library based on it [16]. This model is claimed to properly encode the intrinsic structures and bridges the aforementioned gaps of scalability and performance. It is an adaptation of the half edge representation in the field of computational geometry [17] in order to represent and interrogate large taxonomies in an efficient manner.

1) Applications of topology-based STSS in Twitter Analysis: Rudrapal et al. [18] proposed a method for measuring the semantic similarity between Bengali tweets using the Bengali WordNet developed by Das and Bandyopadhyay [19]. The Bengali model computes the semantic similarity score of a pair of tweets through the use of a lexical based method. It is built on the basis of analyzing common words similarity among tweets. This approach may be used for English tweets, bearing in mind that Bengali tweets are less noisy in nature compared to English, and therefore requires less comprehensive pre-processing. This is because people tend to use fewer abbreviated words (e.g. "great" instead of "gr8"), character repetition (e.g. "heeeey" for "hey"), etc. in Bengali tweets. Another approach to applying topological STSS which is based on knowledge bases is provided in [20]. The authors utilized the English WordNet ontology [21] to estimate the semantic score between microblogs and recommended the top similar microblog records to the user. In their approach, the authors computed the similarity between sentences based on the similarity of the pairs of words contained in the corresponding sentences. Furthermore, the semantic similarity between two word senses is captured through path length, in which the taxonomy is treated as an undirected graph and the distance is calculated between them based on WordNet. The performance of this approach was compared to a statistical based approach, which will be presented and discussed in Section III.B. Findings suggested that this topological-based approach performed better than the statistical-based one in terms of precision. Further research aimed at comparing the performance of several models for determining topic coherence in relation to a Twitter dataset with human assessments has been conducted in [22]. Among the utilized models, the approach employed an individual thesaurus and corpus based measures to determine the semantic similarity between terms within extracted topics from the Twitter dataset. The topics were identified through Latent Dirichlet Allocation (LDA) (described further in Section III.B) and each topic was represented by the top ten words ranked according to their probabilities in the term distribution. Any two words from these top ten form word pairs of a topic and the topic coherence is measured by averaging the semantic similarity of all word pairs in that topic. In this approach, the semantic similarity was computed by using individual measures on WordNet and statistical measures on Wikipedia and a Twitter corpus containing 30,151,847 processed tweets. Three path length based methods were used to calculate the lexical similarity between words in WordNet, LCH [23], JCN [24], and LESK [4]. LCH finds the shortest path between concepts in WordNet. This path length is then scaled by the maximum length observed in the "is-a" hierarchy, in which the two concepts occur. JCN, on the other hand, includes the information of the least common subsumer in addition to the shortest path length. Finally, LESK incorporates information from WordNet glosses, where it finds overlaps between the glosses of the two concepts under consideration, in addition to the concepts that directly link to them. This WordNet based approach will be referred to in the subsequent section, where comparisons are made.

B. Statistical-Based STSS

Statistical approaches determine the semantic similarity between short texts through calculating words co-occurrence frequencies based on a large corpus of text. Deerwester et al.'s Latent Semantic Analysis (LSA) is the prominent statistical-based semantic similarity measure, which is provided as a method for information retrieval [25]. LSA, which is sometimes referred to as Latent Semantic Indexing (LSI), is based on the distributional hypotheses that words similar in meaning will occur in similar contexts [26]. Therefore, calculating word similarity can be derived from a statistical analysis of a large text corpus. The set of unique terms and documents (short-texts in this context) in the corpus are used to generate a high dimensional matrix of terms occurrences. This term-document matrix is commonly decomposed by the application of a matrix factorization algorithm such as Singular Value Decomposition (SVD). The incorporation of SVD into LSA reduces the dimensionality of the single frequency matrix through approximating it into three sub matrices, term-concept matrix, singular value matrix, and concept-document matrix. The SVD process in LSA preserves the important semantic information while reducing noise presented in the original space. It has been found that SVD has improved the effectiveness of word similarity measures [27].

LDA is a semantic topic extraction model that is based on probabilities [28]. LDA is a significant extension of LSA, where terms are grouped into topics, in which most of these terms exist in more than one topic [29]. Despite the commonalities between LDA and LSA, each of the

algorithms generate distinct models. While LSA uses SVD in which the maximum variance across the data is determined for a reduced number of dimensions, LDA employs a Bayesian model. This model considers each document as a mixture of underlying topics and every topic is modeled as a mixture of term probabilities from a vocabulary. Moreover, even though LDA and LSA outputs may be used in similar scenarios, the values of their outputs represent completely different quantities, with different ranges and meanings. LSA generates term by concept and document by concept correlation matrices, with values ranging between -1 and 1 with negative values denoting inverse correlations. On the other hand, LDA generates term by topic and document by topic probability matrices, in which probabilities range from 0 to 1. LDA has an advantage over LSA, which is its ability to tackle the problem of disambiguation and therefore has higher accuracy. This is done by comparing a document to two topics and determining which of them is closer to the document, across all combinations of topics that seem broadly relevant. This direct interpretation of similarities and differences between the most effective statistical semantic measures is important for the challenging process of understanding which measure may be most appropriate for a given text analysis task.

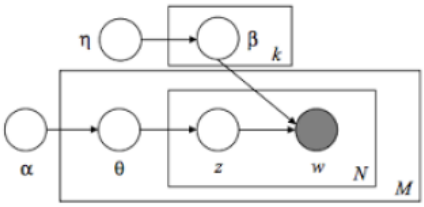


Fig. 2. LDA graphical model [28]

In recent years, there has been an increase in approaches proposing to compose word vectors by using neural language models, which have a core of trained neural networks [30]. Given a sequence of initial words, early neural models were designed to predict the next word in the sentence [31] (e.g. text input auto-completion). While these models can be trained with a variety of techniques to achieve different tasks, they share a common feature of having at their core a dense vector representation of words that can be exploited for computing similarity. This representation is commonly referred to as "neural word embedding", in which their effectiveness varies with regard to the chosen technique and corpus for similarity computation.

1) Applications of statistical-based STSS in Twitter analysis: Steiger et al. used LDA to assess the semantic similarity among tweets [32]. A corpus of 20.4 million processed tweets was created as the lexical resource for which LDA performed its semantic probabilistic model. The application of LDA reduced the semantic dimensions through clustering co-occurring words into topics. Each topic is referred to by labeling it with the highest probability associated words (>0.03). In their adopted approach of LDA, Steiger et al. assumed each tweet α contains a random number of topics, and each topic is characterized by a word distribution β (see Fig. 2). For an individual word w within each tweet, z is the corresponding associated topic. The topic distribution for the overall number of tweets M is denoted by θ, each being of length N. The main challenges encountered, were the estimation of the posterior parameter and the computation of variables such as the number of topics k. However, this study has several limitations that need to be further addressed. Some pitfalls within the bag-of-words (BOW) assumption of LDA caused words to be assigned to various topics while they should be associated with the same topic. Moreover, taking into consideration the syntactical structure (e.g. n-grams) would allow for word orders to be associated to several topics, and therefore better handle semantic complexities. Further, this study did not include the author-topic model [33] (i.e. all tweets of the same user are treated as a single document) due to missing benchmarking process.

Another study that used LDA to gauge the semantic similarity in the context of Twitter data, includes the work presented in [20], in which a corpus of 548 tweets is used. In this approach, each tweet (microblog) is represented as a topic vector, and consequently, the similarity calculation between tweets is equal to the dot product of the two corresponding topic vectors. This statistical method of assessing the semantic similarity was evaluated and compared to the performance of the topology based approach explained earlier in Section III.A. The results showed that the topological-based approach performed better than the topic-based one in terms of precision.

LSA and Pointwise Mutual Information (PMI) statistical approaches were used on Wikipedia and a background dataset of tweets as corpora. SVD was applied to reduce LSA space to 300 dimensions. The empirical evaluation showed that the PMI based measure using Twitter corpus worked better than PMI using Wikipedia, and it best matched the human ground truth ranking of topic coherence on Twitter among all semantic similarity measures used. This might be due to the generic and formal nature of Wikipedia that may prevent capturing specific terms and trends used in Twitter.

C. Hybrid-Based STSS

Some of the topological methods of estimating the semantic similarity may incorporate a statistical function of term frequency in a corpus in order to determine the value of a concept [34-38]. However, their fundamental component of determining the degree of semantic equivalence remains based on a predefined ontology. The similarity computation might also be composed of a combination of statistical and topological methods.

STASIS [35] is an effective measure that estimates the semantic similarity between short sentences based on topological information derived from WordNet ontology and

statistical information obtained through the use of the Brown corpus [39]. This measure calculates the overall semantic score of similarity between two sentences based on a function of multiple factors. These factors include the path between two synsets in the ontology, depth of the subsumer in the hierarchical semantic nets, and information content derived from the Brown corpus. STASIS forms a word order vector composed of unique words contained in both sentences. The combination of syntactic word order and semantic information determines the overall similarity. Although the proposed method does not consider word sense disambiguation for polysemous words as this would scale up the measure's complexity, it still performs well as per the experimental results.

During the last few years, many state-of-the-art STSS approaches have used linear combinations of measures. For example, six topology-based and two statistical-based measures were tested in [40], for the related task of paraphrase identification. In this work, the efficacy of applying topological-based word similarity measures was explored in comparison to texts. They reported that the two approaches are comparable to corpus-based measures such as LSA. The authors of [41] proposed a method that uses a combination of mandatory (string and semantic word) and optional (common word order) similarities. Evaluated on a dataset of 30 sentence pairs, this method outperformed the correlation obtained in [35]. Moreover, a hybrid approach was proposed in [34] where the authors combined a statistical-based semantic relatedness measure over the complete sentence in addition to a topology-based semantic similarity scores that were computed for the words that share similar syntactical role labels in both sentences. These calculated scores performed as the features that were fed to machine learning models such as BOW to predict a single similarity score given two sentences. Results of this method showed a significant improvement of a hybrid measure compared to corpus-based measures taken alone. UKP (Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures) [38], is a similarity detection system that showed reasonable correlation results. It implemented a string similarity, a semantic similarity, and text expansion mechanisms and measures related to structure and style. These multiple text similarity measures were combined through the use of a simple regression model based on training data.

1) Applications of hybrid-based STSS in Twitter Analysis: Das and Smith presented an approach for measuring the semantic similarity between pairs of tweets through identifying whether the two hold a paraphrase relationship [36]. The probabilistic model incorporates syntax and lexical semantics to compute the similarity between two sentences by using a logistic regression model, with eighteen features based on n-grams. The system builds a binary classification model for identifying paraphrase through using precision, recall, and F1-score of n-gram tokens from sentence pairs. The model is capable of determining whether there exists a semantic relationship between a pair of tweets. However, it may be improved by principled combination with more standard lexical approaches.

SemSim is a hybrid based semantic textual similarity system, composed of several modules designed to handle the automatic computation of the degree of equivalence between pieces of multilingual short-text *[37]*. The system was developed to handle general short texts segments and has been tested on a tweets dataset. The system is composed of a module for calculating the semantic similarity of words and another one for pairs of short-text. The former is the core of the system that computes the semantic similarity based on a combination of HAL and WordNet. The semantic textual similarity module uses the semantic word similarity model to calculate the similarity between pairs of short-text. Keywords similarities are calculated through the word similarity module after aligning multiple terms in one sentence to a single term in the other sentence. The words are then paired and the overall similarity score is computed through the semantic textual similarity (STS) module. Generally, SemSim demonstrated a good performance in terms of correlation, but performed poorly in the case of tweets. This is attributed to the absence of some words in the vocabulary, and the top definitions of other words are not always reliable as they may be less prominent.

This section highlighted current state-of-the-art algorithms to distinguish areas of improvement and stimulate creativity towards the development of new approaches. **RQ1** has been explored through discussing settings and features of the aforementioned algorithms in the context of Twitter text analysis. To the best of our knowledge, STSS measures have not been previously reviewed with regard to social media data. Tackling **RQ1** paves the way towards **RQ2** which investigates weaknesses of applying current STSS measures on the noisy and challenging social data and calls for improvement in research and practice. These challenges and weaknesses are further emphasized in the subsequent section.

IV. STSS CHALLENGES IN TWITTER

One of the most difficult aspects of NLP is to establish the understanding and reasoning of the underlying meaning of the text. The challenge of measuring the semantic similarity increases when there is a reduced quantity and quality of text. In terms of social media data, particularly Twitter, the task becomes much harder due to many inaccuracies that may be present in the short pieces of text. These inaccuracies include:

1) Poor grammatical and syntactical structure due to the character limit which encourage the frequent use of abbreviations and irregular expressions [9].
2) Misspellings, out-of-vocabulary words, and acronyms.
3) Lots of redundant information as people tend to repost some original messages.
4) Conventions such as hashtags and other metadata that may interrupt the potential meaning in a text.

Due to these inaccuracies, computers face difficulties in understanding the intended meaning or associating the semantic similarity between pairs of tweets. This is especially true in a tweet which expresses sarcasm, such as "*I enjoy waiting forever for my appointment*", which is common in social media. Therefore, the automation of this process through computation is a challenging task as there are general conventions (hashtags, mentions, URLs, and etc.) and improper English, such as spelling mistakes (e.g. *bcuz* instead of *because*), shared on this communication platform. Many approaches to STSS measures have been based upon adaptation of existing document similarity methods of general English, with no comprehensive consideration of the language used in Twitter. As such, these methods are less applicable to the problem domain of Twitter analysis.

Several key points with regards to the challenges of the STSS approach in social media datasets, particularly Twitter, have been observed:

1) Topological-based approaches use ontologies to capture the semantic similarity between concepts. These approaches often demonstrate scalable and acceptable performance, however, when applied in the context of social media, their performance degrades. This is due to the informal terms used in these sites that are absent from these English dictionaries. To minimize this problem, some approaches suggest using external informal dictionaries for dealing with out-of-vocabulary tokens.
2) Statistical-based methodologies are not effective for measuring the semantic similarity for short and sparse text as they are for long and rich text. However, they tend to perform better when the utilized corpus consists of the same domain than the case of general corpus, such as the Brown corpus. This is due to the fact that these corpora contain information from traditional media and therefore may fail to capture specific terms and trends dynamically propagated through social media networks.
3) Although not many hybrid based systems were developed for the intended approach, it can be observed that these approaches outperform single measures of determining the semantic similarity between short segments of texts. However, they tend to consume high computational resources.

Moreover, it has been observed that a robust pre-processing and feature extractor function that is able to normalize and extract Twitter specific text features may significantly improve the performance of STSS measures in the context of social media data [42], [43], [11].

V. Experiment Methodology

As demonstrated in Section III, STSS measures differ according to their core body of components and functionality. Therefore, an experiment was designed and implemented in order to evaluate the validity of different semantic versus non-semantic STSS when applied in the context of Twitter OSN. These experiments require a dataset that is subjectively annotated with human ratings of the actual similarity score by a predefined class of annotators. Part of the SemEval-2014 shared task comprises a published annotated news tweets training and testing dataset [44]. A corpus of the training data was built for weighting the terms and for the statistical analysis performed by LSA.

This section describes the experiment conducted to evaluate the level of effectiveness of the measures explained in Section III. The results of the measures were normalized as each measure scores on different scale. The empirical evaluation of the measures were made through several statistical analysis and tests in order to answer **RQ3**. These are further elaborated in the subsequent sections.

A. Hypothesis

The hypothesis to be tested relates to the accuracy of the similarity measure compared to typical human cognition similarity assessment, which is as follows:

$H0_a$ - *The similarity measure deployed can accurately approximate human cognition of semantic interpretation. That is, there is no statistically significant difference between the actual (human) and predicted (measure) values.*

$H0_b$ - Actual and predicted values are numerically close.

$H1_a$ - *The similarity measure is unable to produce a relatively accurate similarity judgment. That is, there is a statistically significant difference between the actual (human) and predicted (measure) values.*

$H1_b$ - Actual and predicted values are numerically not close.

B. Experiment Design

An implementation of the measures under consideration was developed and the outcome was evaluated against a benchmark. The experiment carried out was set to test the correlation between the similarity scores of the human judges and results of the implemented measures. The experimental analysis outcome will provide insights on the direction and potential measure improvement that can be addressed through further research.

The effectiveness of the designed experiment is tested through a representative random sample of the SemEval-2014 dataset. The analysis of the experiment results will be used in further research towards approximating human cognition in similarity assignment and adjusting features and measure's parameters to maximize its accuracy.

C. Dataset and Sample Size

SemEval-2014 is a collection of computational semantic analysis tasks intended to explore the nature of meaning in language. It carried out several semantic tasks, including evaluation of compositional distributional semantic measures through entailment and multilingual semantic textual similarity in Twitter. Multiple datasets were published for

system training and testing in order to unify the evaluation and allow for a fair comparison of all contributions. However, as this experiment is aimed at evaluating the capability of a measure to capture the semantic between pairs of tweets, it is necessary have a dataset that is labelled with human ratings. Part of the published trial datasets is a tweet-news dataset containing 750 annotated pairs [44]. The gold standard implements a 5-point Likert scale to interpret the degree of similarity between pairs, as defined by Agirre [45].

D. Experiment STSS Measures

1) Weighted keyword-based similarity: The first implemented similarity approach is based on shared keywords rather than semantic similarity. Given the corpus that was generated from the evaluation dataset, each document (tweet) is represented by a vector of weighted terms in that corpus. Each term is then represented by the number of its occurrences in the document multiplied by its frequency of occurrence in the whole corpus as in

$$tf - idf_{t,d} = \mathrm{tf}_{t,d} \cdot \log \frac{N}{\mathrm{df}_t} \qquad (1)$$

Where, $tf_{t,d}$ is the total number of occurrences of t in d, df_t is the total number of documents containing i, and N is the total number of documents in the corpus. Finally, the cosine of the two vectors (representation of the two short-texts under consideration) yields the similarity value.

2) LSA: Several statistical-based similarity measures have been reviewed and LSA was nominated as it has been reported to outperform LDA in a system that measures the similarity between movies based on their metadata [46]. Although the movies dataset is different than a dataset of tweets, it might uncover potential insights as both datasets share mutual prominent factor, which is the short-text content. There has not been found any equivalent or similar study that was performed on a Twitter dataset.

3) STASIS: STASIS is selected as it accounts for word order as part of its system components. STASIS assigns the similarity score based on a combination of the syntactic and semantic ratio of similarity. Hence, it may have potential capabilities for the domain under consideration. However, this measure was tested on a dataset of short formal English sentences that utilizes WordNet and the Brown corpus, whereas the data under consideration has lots of informality and out of dictionary terms. Therefore, it is necessary to determine and evaluate its applicability through experiments.

E. Feature Set

A feature extractor module has been implemented to parse the text input and generate a set of features that represents the given tweet. In the conducted experiment, the input was represented by the set of weighted unigrams that are presented in a tweet, which are non-function words. The term weights were calculated according to (1).

F. Experimental Analysis and Evaluation Metrics

The data gathered from each run was collected and subsequently analyzed to explore the findings from the experiment. The experiment results are evaluated through several measures to ensure that they are thoroughly analyzed. These measures include the Pearson correlation coefficient, Spearman's rank correlation coefficient, MSE, and a statistical hypothesis test. These are further elaborated in Section VI.

VI. Experiment Results and Analysis

This section discusses the result of the evaluation metrics.

A. Rational for the Selection of Evaluation Measures

Correlation coefficient: Pearson correlation has been a common practice for assessing the performance of STSS systems through computing the correlation between human judgments and machine assigned semantic similarity scores [1]. Systems that record higher correlations are generally considered "accurate", and would often be among the top choices for the system designer of an STSS based evaluation task. However, this common practice of STSS evaluation through Pearson correlation has been questioned previously.

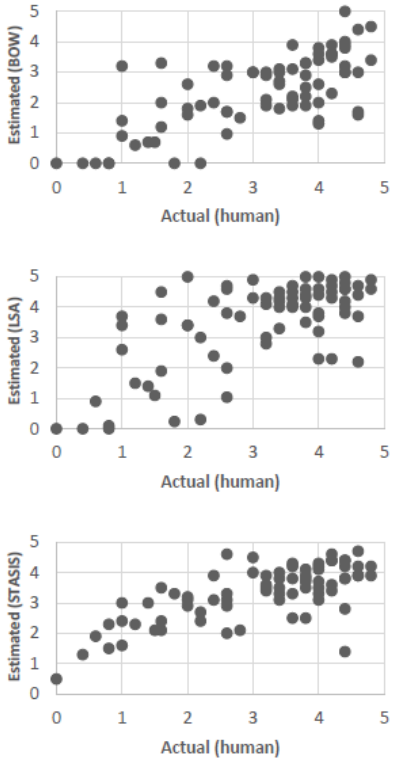


Fig. 3. Correlation scatterplots between actual and estimated values

Zesch [47], reported several limitations of the Pearson correlation,

1) Sensitive to outliers.
2) Limited to measuring linear relationships.
3) The two variables need to be approximately normally distributed.

Zesch recommended the usage of Spearman's rank ρ correlation coefficient as it is not sensitive to outliers, non-linear relationships, and non-normally distributed data. However, most evaluation methods of STSS systems only report the Pearson correlation. Nevertheless, the experiment results were evaluated via computing both Pearson and Spearman's correlation coefficient to avoid uncertainty.

Although Pearson and Spearman's tend to perform different calculations, both outcomes are interpreted in the same way that is mentioned above. Correlation scatterplots between the measures and human annotations are shown in Fig. 3, where each point represent a pair in the dataset.

1) MSE: Agirre [1] mentioned in SemEval-2013 discussion: "*Evaluation of STS is still an open issue*" and in addition to the Pearson correlation, *"...other alternatives need to be considered, depending on the requirements of the target application*". Therefore, it is reasonable to compute the average error rate between the actual and estimated values, and assess the STSS measures accordingly.

TABLE I. TEST SET RESULTS ON SEMEVAL-2014

Measure	*r*	*ρ*	*MSE*
Weighted_BOW	0.7102	0.6517	1.4009
LSA	0.6753	0.5692	1.3304
STASIS	0.7086	0.6567	0.8168

The least MSE results are the closest to human judgments. The results on the SemEval-2014 dataset with gold standards are summarized in Table 1, showing Pearson's *r*, Spearman's *ρ*, and MSE.

B. Statistical Test

Selecting an appropriate statistical technique for testing the hypothesis is the most difficult part when conducting research [48]. This is attributed to the lack of a universal methodology that clearly guides researchers on the right statistical test choice [49]. The challenge of this choice refers to the variations in the nature of research, as it depends on the type of research questions that need to be addressed. In terms of the STSS measures, it also depends on the scale of similarity assignment, the variables to be analyzed, the underlying assumptions for specific statistical techniques, and the nature of the data itself [48].

Parametric tests are inferential statistical analysis based on assumptions regarding the population and require numerical score [50]. Non-parametric techniques do not employ such strict requirements nor do they make distribution assumptions, and therefore sometimes referred to as distribution free tests. These tests are most often used with categorical and ordinal data as they do not require the data to be normally distributed and are not based on a set of assumptions about the population [51].

The "Test of normality" is investigated to test the distribution of the data. It is generally agreed that significant values greater than 0.05 indicate that the data is similar to a normal distribution, otherwise it is not normally distributed.

TABLE II. TEST OF NORMALITY

	Kolmogorov-Smirnov			Shapiro-Wilk		
	Statistic	df	Sig.(p)	Statistic	df	Sig.(p)
Human	.145	75	.000	.924.	75	.000
BOW	.125	75	.006	.963	75	.028
LSA	.188	75	.000	.840	75	.000
STASIS	.105	75	.039	.946	75	.003

Table 2 presents the results of the normality test. As the data is not normally distributed, a nonparametric test will be utilized for the data analysis. Hence, the Wilcoxon Signed Rank Test will be used to test the hypothesis. This test is the nonparametric alternative to the repeated measure *t*-test, however, Wilcoxon converts scores to ranks and compares them instead of comparing the means of the two systems under study. It can be concluded that the differences between the two scores is statistically significant, if the significance level (p-value) is equal to or less than .05 [48].

In addition to classifying the data in terms of normality, inferential statistical analysis tests were carried out to investigate whether the similarity results obtained from each measure are any close to human judgments.

C. Inferential Statistical Analysis

Wilcoxon Signed Rank was used to test the following hypothesis:

$H0_a$: $\mu d = 0$ (No significant difference between the actual and measured values)

$H1_a$: $\mu d \neq 1$ (Significant difference between the actual and measured values)

1) Hypothesis Result: A Wilcoxon Signed Rank test was established on each measure paired with the gold standard, were actual refers to human judgments and estimated refers to similarity measurements.

TABLE III. WILCOXON SIGNED RANK TEST RESULTS

Test Statistics			
Actual	**Predicted**	**Z**	**Asymp. Sig.**
Human annotation	Weighted_BOW	-5.633	.000
	LSA	-3.125	.002
	STASIS	-2.320	.020

The results demonstrated that for each of the similarity measures tested to evaluate the accuracy of the measures in the context of Twitter short-text, there is a statistically significant difference (p-value < 0.05) between the similarity obtained by the measures and the gold standard (accept **$H1_a$** and reject **$H0_a$**). Consequently, this means that the actual and predicted values are numerically not close (accept **$H1_b$** and reject **$H0_b$**). The results of the statistical analysis are present in Table 3. The evaluation methods are further discussed in Section VII.

VII. Discussion

The goal of the evaluation criteria utilized to gauge the performant of the STSS measures are twofold. The first part involved employing metrics to assess and compare the accuracy between measures under investigation in relation to the gold standard. Whereas the next part involved performing an inferential statistical analysis to test how close are the measures to human judgment.

The evaluation using Pearson correlation demonstrated the highest result for the weighted BOW (0.7102) and the lowest for LSA (0.6753). However, these results might not be reliable as the data contained outliers, such as a tweet that is composed of two words or even one, in which Pearson correlation is sensitive. Therefore, the correlations were better represented using Spearman's rank, which employs rankings instead of the actual scores. The results on the SemEval-2014 dataset based on Spearman's showed that there is no strong correlation for the three measures; however, STASIS and the weighted BOW approach were more correlated to human judgments than LSA, with STASIS slightly higher. However, the intrinsic common evaluation based on only correlation in the differentiation between STSS systems might be ill suited as mentioned earlier in Section VI. Therefore, the need of an additional evaluation measure has led to calculating the MSE in order to find out which one had the least error rate. STASIS had an average error of 0.8168, LSA 1.3304, and weighted BOW recorded 1.4009 when compared with the gold standard. It can be concluded that the semantic-based measures performed better than the keyword-based, although LSA was not substantially less than the weighted BOW (0.1), but STASIS was less by 0.6.

The inferential analysis revealed negative statistics not only for the keyword-based approach, but also for the statistical and for hybrid based approaches. The Wilcoxon Signed Rank test showed that there is a significant difference between the similarity scores obtained by the three measures, and the gold standard. This is attributed to the dataset that these measures were applied to. While the evaluated measures may be effective in approximating the human ratings in different settings of short-text data, it is evident that the challenges present in Twitter language (discussed in section IV) are hampering the accuracy and effectiveness of these measures. These require further research to enhance the performance of the semantic similarity measure.

The analysis of the results are useful in guiding further work of measure adaptation to deal with the textual challenges present in Twitter. This can be achieved through examining cases where the measure performed poorly and adjusting parameters, such as redesigning the feature set in a way that had better capture a tweet's semantical structure.

VIII. Colclusion and Future Work

This paper presents the work conducted to address the research questions provided in Section I.B. The evaluation of different STSS measures revealed insights for the development of new STSS measures to overcome the weaknesses of existing ones in capturing the semantics of Twitter data.

The experimental results showed evidence that, although the evaluated measures may produce high correlations when dealing with proper English text, the nature of most short-textual data propagated in social media, are hindering the performance of these measures. Thus, it is imperative to adapt the components of such measures in a way that can understand the modern natural language generated in Twitter. This is particularly useful for applications of Machine Learning handling social media data.

Towards proceeding with future research, the preliminary evaluation revealed key information regarding the accuracy of STSS measures compared to a non-semantic based measure in the context of Twitter data. The main observations are summarized as follows:

- The features used in the implemented experiment are not adequate to handle the challenges presented in the language and structure of Twitter data, and therefore additional preprocessing and features need to be utilized.
- Semantic-based measures performed better than the keyword-based measure in detecting the degree of semantic equivalence between pairs of tweets.
- While STASIS performed better than LSA, they are both potential contenders for estimating the semantic similarity between tweets and therefore require further investigation, as some of their components may be integrated and utilized for developing a Twitter-specific semantic similarity measure.

Further research continue on towards determining new methodologies for adapting and developing scalable and robust STSS measures that can handle the unstructured and noisy microblogging data.

References

[1] Agirre, E., et al. *SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation*. in *SemEval@ NAACL-HLT*. 2016.

[2] O'Shea, J., et al., *A comparative study of two short text semantic similarity measures*. Agent and Multi-Agent Systems: Technologies and Applications, 2008: p. 172-181.

[3] Rocchio, J.J., *Relevance feedback in information retrieval*. The SMART retrieval system: experiments in automatic document processing, 1971: p. 313-323.

[4] Lesk, M. *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone*. in *Proceedings of the 5th annual international conference on Systems documentation*. 1986. ACM.

[5] Salton, G. and C. Buckley, *Term-weighting approaches in automatic text retrieval*. Information processing & management, 1988. **24**(5): p. 513-523.

[6] O'Shea, K., Z. Bandar, and K. Crockett, *A conversational agent framework using semantic analysis*. International Journal of Intelligent Computing Research (IJICR), 2010. **1**(1/2).

[7] Lin, Y.-S., J.-Y. Jiang, and S.-J. Lee, *A similarity measure for text classification and clustering*. IEEE transactions on knowledge and data engineering, 2014. **26**(7): p. 1575-1590.

[8] Albitar, S., S. Fournier, and B. Espinasse. *An effective TF/IDF-based text-to-text semantic similarity measure for text classification*. in *International Conference on Web Information Systems Engineering*. 2014. Springer.

[9] Alnajran, N., et al. *Cluster Analysis of Twitter Data: A Review of Algorithms*. in *9th International Conference on Agents and Artificial Intelligence*. 2017. SCITEPRESS.

[10] Farzindar, A. and D. Inkpen, *Natural language processing for social media*. Synthesis Lectures on Human Language Technologies, 2017. **10**(2): p. 1-195.

[11] Gómez-Adorno, H., et al., *Improving feature representation based on a neural network for author profiling in social media texts*. Computational intelligence and neuroscience, 2016. **2016**: p. 2.

[12] Guo, W. and M. Diab. *A simple unsupervised latent semantics based approach for sentence similarity*. in *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. 2012. Association for Computational Linguistics.

[13] Zanzotto, F.M., M. Pennacchiotti, and K. Tsioutsiouliklis. *Linguistic redundancy in twitter*. in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2011. Association for Computational Linguistics.

[14] Xu, W., A. Ritter, and R. Grishman. *Gathering and generating paraphrases from twitter with application to normalization*. in *Proceedings of the sixth workshop on building and using comparable corpora*. 2013.

[15] Liu, H. and P. Wang, *Assessing Text Semantic Similarity Using Ontology*. JSW, 2014. **9**(2): p. 490-497.

[16] Lastra-Díaz, J.J., et al., *HESML: A scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset*. Information Systems, 2017. **66**: p. 97-118.

[17] Botsch, M., et al., *Openmesh-a generic and efficient polygon mesh data structure*. 2002.

[18] Rudrapal, D., A. Das, and B. Bhattacharya. *Measuring Semantic Similarity for Bengali Tweets Using WordNet*. in *Proceedings of the International Conference Recent Advances in Natural Language Processing*. 2015.

[19] Das, D. and S. Bandyopadhyay. *Developing Bengali WordNet affect for analyzing emotion*. in *International Conference on the Computer Processing of Oriental Languages*. 2010.

[20] Chen, X., et al. *Recommending Related Microblogs: A Comparison Between Topic and WordNet based Approaches*. in *AAAI*. 2012.

[21] Miller, G.A., *WordNet: a lexical database for English*. Communications of the ACM, 1995. **38**(11): p. 39-41.

[22] Fang, A., et al. *Topics in tweets: A user study of topic coherence metrics for Twitter data*. in *European Conference on Information Retrieval*. 2016. Springer.

[23] Leacock, C. and M. Chodorow, *Combining local context and WordNet similarity for word sense identification*. WordNet: An electronic lexical database, 1998. **49**(2): p. 265-283.

[24] Jiang, J.J. and D.W. Conrath, *Semantic similarity based on corpus statistics and lexical taxonomy*. arXiv preprint cmp-lg/9709008, 1997.

[25] Deerwester, S., et al., *Indexing by latent semantic analysis*. Journal of the American society for information science, 1990. **41**(6): p. 391-407.

[26] Harris, Z.S., *Mathematical structures of language*. 1968.

[27] Landauer, T.K. and S.T. Dumais, *A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge*. Psychological review, 1997. **104**(2): p. 211.

[28] Blei, D.M., A.Y. Ng, and M.I. Jordan, *Latent dirichlet allocation*. Journal of machine Learning research, 2003. **3**(Jan): p. 993-1022.

[29] Crossno, P.J., et al. *Topicview: Visually comparing topic models of text collections*. in *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*. 2011. IEEE.

[30] Christoph, L., *Measuring Semantic Similarity and Relatedness with Distributional and Knowledge-based Approaches*. 2016.

[31] Mnih, A. and G.E. Hinton. *A scalable hierarchical distributed language model*. in *Advances in neural information processing systems*. 2009.

[32] Steiger, E., et al., *Twitter as an indicator for whereabouts of people? Correlating Twitter with UK census data*. Computers, Environment and Urban Systems, 2015. **54**: p. 255-265.

[33] Zhao, W.X., et al. *Comparing twitter and traditional media using topic models*. in *European Conference on Information Retrieval*. 2011. Springer.

[34] Aggarwal, N., K. Asooja, and P. Buitelaar. *DERI&UPM: Pushing corpus based relatedness to similarity: Shared task system description*. in *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. 2012. Association for Computational Linguistics.

[35] Li, Y., et al., *Sentence similarity based on semantic nets and corpus statistics*. IEEE transactions on knowledge and data engineering, 2006. **18**(8): p. 1138-1150.

[36] Das, D. and N.A. Smith. *Paraphrase identification as probabilistic quasi-synchronous recognition*. in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. 2009. Association for Computational Linguistics.

[37] Kashyap, A., et al., *Robust semantic text similarity using LSA, machine learning, and linguistic resources*. Language Resources and Evaluation, 2016. **50**(1): p. 125-161.

[38] Bär, D., et al. *Ukp: Computing semantic textual similarity by combining multiple content similarity measures*. in *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. 2012. Association for Computational Linguistics.

[39] Francis, W.N. and H. Kucera, *Brown corpus*. Department of Linguistics, Brown University, Providence, Rhode Island, 1964. **1**.

[40] Mihalcea, R., C. Corley, and C. Strapparava. *Corpus-based and knowledge-based measures of text semantic similarity*. in *AAAI*. 2006.

[41] Islam, A. and D. Inkpen, *Semantic text similarity using corpus-based word similarity and string similarity*. ACM Transactions on Knowledge Discovery from Data (TKDD), 2008. **2**(2): p. 10.

[42] Duong, P.H., H.T. Nguyen, and N.-T. Huynh. *Measuring Similarity for Short Texts on Social Media*. in *International Conference on Computational Social Networks*. 2016. Springer.

[43] Demirsoz, O. and R. Ozcan, *Classification of news-related tweets*. Journal of Information Science, 2016: p. 0165551516653082.

[44] Guo, W., et al. *Linking Tweets to News: A Framework to Enrich Short Text Data in Social Media*. in *ACL (1)*. 2013.

[45] Agirre, E., et al. *Semeval-2012 task 6: A pilot on semantic textual similarity*. in *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. 2012. Association for Computational Linguistics.

[46] Bergamaschi, S. and L. Po. *Comparing LDA and LSA Topic Models for Content-Based Movie Recommendation Systems*. in *International Conference on Web Information Systems and Technologies*. 2014. Springer.

[47] Zesch, T., *Study of semantic relatedness of words using collaboratively constructed semantic resources*. 2010, Technische Universität.

[48] Pallant, J., *SPSS survival manual*. 2013: McGraw-Hill Education (UK).

[49] Kinear, P.R. and C.D. Gray, *SPSS for Windows made simple: release 10*. 2001: Psychology Press.

[50] Gravetter, F.J. and L.B. Wallnau, *Statistics for the behavioral sciences*. 2016: Cengage Learning.

[51] Nolan, S.A. and T. Heinzen, *Statistics for the behavioral sciences*. 2011: Macmillan.

A Heuristic Based Pre-processing Methodology for Short Text Similarity Measures in Microblogs

Noufa Alnajran*, Keeley Crockett*, *Senior Member, IEEE*, David McLean*, and Annabel Latham*, *Member, IEEE*

Abstract—**Short text similarity measures have lots of applications in online social networks (OSN), as they are being integrated in machine learning algorithms. However, the data quality is a major challenge in most OSNs, particularly Twitter. The sparse, ambiguous, informal, and unstructured nature of the medium impose difficulties to capture the underlying semantics of the text. Therefore, text pre-processing is a crucial phase in similarity identification applications, such as clustering and classification. This is because selecting the appropriate data processing methods contributes to the increase in correlations of the similarity measure. This research proposes a novel heuristic-driven pre-processing methodology for enhancing the performance of similarity measures in the context of Twitter tweets. The components of the proposed pre-processing methodology are discussed and evaluated on an annotated dataset that was published as part of SemEval-2014 shared task. An experimental analysis was conducted using the cosine angle as a similarity measure to assess the effect of our method against a baseline (C-Method). Experimental results indicate that our approach outperforms the baseline in terms of correlations and error rates.**

Keywords—Twitter, Short Text Similarity, Text Mining, Natural Language Processing

I. INTRODUCTION

The remarkable growth of user generated content (UGC) in OSN has offered individuals and organisations the ability to maintain and enhance their influence and reputation. Twitter has monthly active users of over 300 million and over half a billion tweets propagated through the medium [1]. The existence of such massive textual data has encouraged researchers and practitioners to collect and perform various machine learning applications, such as clustering in order to draw insightful conclusions about the data. Since the main component in an unsupervised learning algorithm is a distance measure, adapting text similarity measures to the context of tweets have gained much interest recently. This is due to the significant importance of such measures in performing Twitter-based similarity tasks such as classification and clustering [2]. Capturing similarities between tweets can reveal critical information in various domains of modern human life: politics, educations, healthcare, business, security, and so on. Therefore, developing short text semantic similarity (STSS) measures to produce human-like assessments has been a problem in contemporary natural language processing (NLP). In Twitter, this problem is particularly challenging due to the low quality of text in tweets as demonstrated in the following factors:

- **Volume** – existence of massive content generated in the same topic include lots of re-tweets (redundant tweets), which introduce noise and bias in the dataset. For example, the existence of retweets in the dataset can be affecting the terms weighting process.
- **Lack of structure** – users create conventions such as hashtags, mentions, and reference to URLs, which interrupt the structured performance of an STSS measure.
- **Out-of-vocabulary (OOV) words** – users do not usually use proper words that exist in a dictionary. Rather, they create their own words shortcuts, slangs, abbreviations, and genre specific terminology.
- **Emoticons** – users tend to replace words with emoji, which offer room for more text and rich meaning while still conforming to the length restriction.
- **Ambiguity** – as tweets are restricted to 140 characters, they may be ambiguous to interpret due to the lack of context. For example, a tweet containing "New York" could refer to the one in the state of New York or to the one in the state of Missouri. Similarly, including the term "apple" could refer to both the fruit or to the company.

STSS is the process of automatically measuring the degree to which two short texts are semantically equivalent to each other [3]. In Twitter, short texts are "tweets" of informal human utterances that are of sentence length limited to 140 characters. Due to their informality and length restriction, tweets are commonly subject to textual and grammatical inaccuracies. Therefore, they do not conform to the typical syntactical structure of sentences. The aforementioned challenging factors are degrading the performance of STSS measures due to the highly noisy nature of the data. Therefore, it is necessary to integrate a robust pre-processing methodology in the analysis phase. This methodology is required to be capable of cleaning the text to a level that can be analysed by STSS measures while still maintaining the information carried out by the tweet.

This paper proposes an intensive, yet effective pre-processing methodology for reducing noise in the data before

* School of Computing, Mathematics, and Digital Technology
Manchester Metropolitan University
Manchester, Uk
noufa.alnajran@stu.mmu.ac.uk, {k.crockett, d.mclean, a.latham}@mmu.ac.uk

feeding into an STSS algorithm. Unlike existing Twitter-based pre-processing approaches that focus their pre-processing on extracting polarity and sentiment features of the text [1, 4-7], our approach aims at capturing all textual semantic and syntactic features despite the existing noise. This is achieved through performing several pre-processing heuristics, which build up the methodology presented in this paper. The components of this methodology can be adjusted according to the target OSN application.

A. Problem Statement

Text pre-processing plays a significant role in text mining algorithms. This is due to being a primary factor contributing to the pureness of the feature set, and thus accuracy of the produced results. A major problem has emerged as pre-preprocessing becomes a reuse component that is not being customized according to the target application. Therefore, the analysis phase may fail to generate expected results because the data has not been properly processed in the previous stage. For example, in the context of Twitter analysis, one may apply a pre-processing methodology that works well for a sentiment analysis algorithm in a semantic similarity identification task. This will obviously reduce the resulting algorithm's performance due to the persistent noise from the perspective of the algorithm under consideration. This problem is particularly common in applications of STSS measures [2, 3, 8] employing one or more of the following pre-processing pitfalls:

- Following common practices for data scrubbing such as tokenization, part-of-speech (POS) tagging, stemming, lemmatization, and etc. regardless of the required features set contents and target application. As an example of application-based pre-processing, retaining terms with repeated characters is of high value for sentiments analysis applications, but should be standardized for STSS applications in order to map to a vocabulary for interpretation.
- Preforming a crude and comprehensive pre-processing steps, which result in losing important information. Performing stemming and removal of stop words, abbreviations, punctuations, numbers, hashtags, mentions, URLs, and emoji altogether from a very short text (tweet) will result in loss of information.
- Performing inadequate pre-processing steps, which retain unwanted noise in the data. For example, missing to remove redundant data such as re-tweets when performing cluster analysis will result in false clusters.

The lack of a standard structured pre-processing methodology for measuring the semantic similarity of short text messages propagated in Twitter is the motivation for conducting this research.

B. Contributions and Outline

This paper contributes to the research community in the following ways:

1. While the effect of the pre-processing stage have been widely discussed in context of sentiment analysis, it has not been studied yet in applications of STSS measures and its impact on their performance. In this study, an analysis of the pre-processing problem is conducted in relation to measuring the textual semantic similarity.
2. A heuristic-based pre-processing methodology is proposed for Twitter-driven STSS tasks. Rather than harvesting the dataset for extracting sentimental features, this methodology focuses on textual segment that contribute to the meaning carried out by the text.
3. Providing a statistical quantification of the effect of the proposed methodology on the performance of STSS measures in comparison to other preprocessing approaches.

The remainder of the paper is organized as follows: section II discusses and critically analyses existing related works. Section III describes the proposed heuristic-driven pre-processing methodology and its components. Section IV explains the experimental methodology and results and discussion are provided in section V. Finally, section VI presents the conclusions and future work.

II. RELATED WORK

Most existing approaches to Twitter-based STSS measures employ a pre-processing phase to reduce the amount of noise in the tweets [2, 8-11].

However, to the best of our knowledge, there does not exist research that studies the impact of pre-processing practices on applications of STSS. Nevertheless, many research have studied the role and effect of pre-processing on sentiment analysis applications [4-7, 12].

Haddi *et al*. [6] investigated the effect of text pre-processing in the sentiment analysis of online movie reviews. Their study reported that the right text pre-processing methods can remarkably enhance the accuracy of sentiment classification. Saif *et al*. [4] studied the impact of stop words removal on the accuracy of a sentiment classifier. Six stop word identification methods were been applied to six Twitter datasets. The experiment observed the effect of stop words removal on two supervised sentiment classifiers. Results shown that while there is a similar pattern of pre-processing effect on sentiment classifiers across different stop words removal methods, Naïve Bayes Classifiers are more sensitive to stop words removal than the maximum entropy ones. Bao *et al*. [12] explored the role of pre-processing practices in Twitter sentiment classification. The methods they studied are: removal of URLs, standardizing words with repeated letters, negation, stemming, and lemmatization. Experimental results recorded a sentiment classification accuracy of 85.5% when a URL featured reservation, negation transformation, and repeated letters normalization were employed on the Stanford Twitter Sentiment Dataset. Moreover, the impact of URLs, repeated letters, negation, stop words, acronyms, and numbers has been examined in an supervised classification task on Twitter [5]. In their study, the experimental results reported an increase in the classifier accuracy in terms of precision and recall when replacing negation and expanding acronym. It has been further reported that the accuracy hardly change when removing stop words, numbers, and URLs. Singh and Kumari [7] analyzed the

impact of normalization and pre-processing on tweets sentiments. In their work, they investigated the importance of slang words and their effect on measuring the sentiment polarity of a tweet. For experimentation, the authors used a Twitter dataset that comprises of six fields: sentiment class, tweet id, date, query, user, and the text. Experimental results suggest that their proposed scheme perform better in terms of sentiment classifier accuracy.

Fig. 1. A typical text mining process

It can be observed from the above reviews that there is a lack of proper and structured practice of a pre-processing methodology for applications that measure the semantic similarity between tweets, rather than sentiment polarity. To fill this gap, this paper proposes a pre-processing methodology for STSS measures and evaluates the effects of the proposed methodology on a labelled Twitter dataset [13].

III. PROPOSED HEURISTIC-DRIVEN PRE-PROCESSING METHODOLOGY

Pre-processing is considered to be the second step after data collection and one of the most important steps in a typical text mining process (Fig. 1). In text analysis applications, each text is represented by a feature vector. These vectors are derived from the raw text after it has been processed. Towards extracting efficient feature sets for STSS measures, we propose a novel heuristic-driven comprehensive list of pre-processing practices. The novelty of our proposed methodology lies in the compound rule-based steps of pre-processing that is aimed at enhancing the performance of STSS measures, which has not been investigated previously. The selection of our methodology's components was derived upon empirical experiments. In the subsequent sections, we describe these components for processing tweets before being transmitted to the measure for similarity computation. The effectiveness of our proposed pre-processing methodology is validated and provided in the experiment section IV. Fig. 2 shows a flowchart of the heuristic-driven pre-processing methodology proposed in this study.

A. HTML Tags

Lots of html characters such as < > & are embedded in the original data that is retrieved from the web. Our solution for this is the use of regular expressions to convert them to standard html tags. For instance, & is converted to "and". Python provides some packages and modules such as *htmlparser* that does the conversion.

B. Decoding

This form of pre-processing consists of transforming the text into a simple machine readable format. Text may exist in di□erent formats such as Latin, UTF-8, and etc. For an STSS measure, it is necessary to format text consistently in a standard encoding format. For better analysis, it is recommended to use UTF-8 as it is widely accepted.

C. Tokenization

The n-gram language model [14] is the basic building block in constructing a feature vector. For STSS measures we transform the input text into tokens of unigrams (n-gram, n=1). For this task, it is recommended to use the Natural Language Toolkit (NLTK) tokenizer instead of Stanford tokenizer. This is because NLTK tokenizer is familiar with Twitter conventions and emoji, and therefore will not split hashtags or emoticons. Take T_1 as an example, T_1 = "*#Remain 44% #Leave 46%*", Stanford tokenizer will transform T_1 to '#' 'Remain' '44' '%' '#' 'Leave' '46' '%', while NLTK tokenizer will result in: '#Remain' '44%' '#Leave' '46%'. The latter tokenization scheme produces more logical tokens in terms of twitter features and conventions.

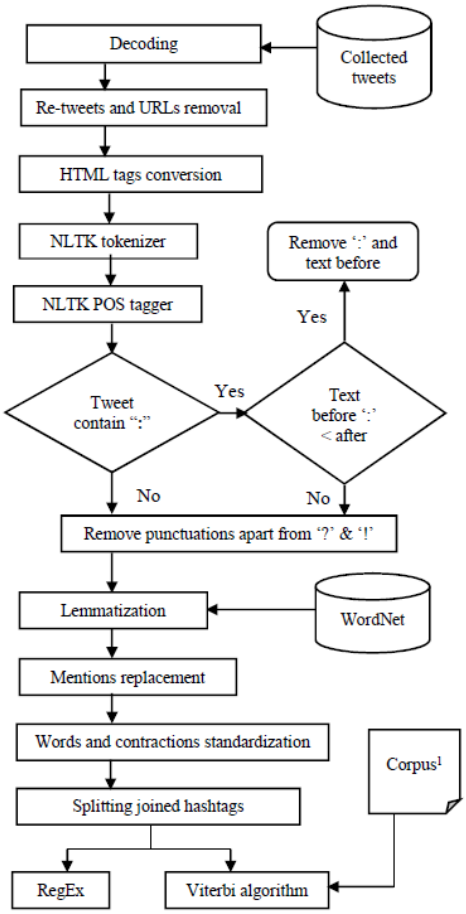


Fig. 2. Proposed heuristic-driven pre-processing components

D. Part-of-Speech (POS)tex Tagging

For STSS measures, performing POS tagging is necessary to identify the syntactical similarity based on the grammatical structure of the text. The process of Named Entity Recognition (NER) is embedded within the POS tagging task. In our methodology, we used NLTK's simple statistical unigram tagging algorithm, which assigns the tag that is most likely for a given token. For example, it will assign the tag *jj* to any occurrence of the word "beautiful", since "beautiful" is used as an adjective (e.g. a *beautiful* city) more often than it is used as other parts of speech.

E. Punctuations

Unlike common approaches of removing all punctuations, we develop a heuristic-based approach for dealing with punctuations to refine the tweet content

- If the tweet contains a ':' and the amount of text after this punctuation is larger than the text before it, then anything before is discarded. For example, "*RT @ronnyhansen1: @CORCAS_AUTONOMY: yes, #Saharawi are sovereign in #WesternSahara, not Morocco. Why not hold agreed referendum to find out…*" becomes "*yes, #Saharawi are sovereign in #WesternSahara, not Morocco. Why not hold agreed referendum to find out…*"
- Question marks and exclamation marks carry structural information contributing to the syntactical similarity between tweets, and therefore are retained. The rest of the punctuations, such as commas and full stops are removed.

F. Hashtags

In this paper, much focus of the proposed heuristic-driven methodology is towards processing hashtags. These Twitter specific annotation formats are the main indicators of a tweet topic. Hashtags are user conventions to create and follow a thread of discussion by prefixing a word with the '#' character [15]. Many studies performed topic identification based on classifying hashtags as these greatly contribute to the meaning of a tweet [16]. Therefore, these are important pieces of information that should be represented in the feature set for a STSS measure. However, hashtags are not usually intuitive to interpret by a computer program.

TABLE I. EXAMPLES OF PREFERRED AND AMBIGUOUS HASHTAG TOKENIZATIONS

Hashtag	Target tokenization	Ambiguous tokenization
#longisland	long island	Long is land
#isreal	isreal	is real
#facebook	Facebook	face book
#healthexchange	health exchange	heal the x change

A major problem with hashtags is that they are often composed of joined words. While some hashtags are composed of joined words starting with capital letters, such as "*#JoyDivision*", most joined words are lowered cased. In the latter case, the challenge lies in determining where the boundaries are between the joined words. For example, given a hashtag such as *#talksofthemonth* return "talks of the month" and not "talk soft he month". Table 1 shows samples of joined hashtags and their possible interpretations. Due to this challenge, most approaches to STSS measures in Twitter either ignore hashtags [2] or simply remove the hash character and treat the rest as a single word [17]. Consequently, a portion of the similarity between the two texts will be missing.

In this work, we propose a heuristic-based pre-processing methodology for handling the problem of hashtag compound segmentation. Let h be a hashtag of compound words, our algorithm works as follows

1. If the regular expression based conditional statement S <h is composed of upper and lower case characters> is *true*, the boundaries upon which the words in h are split, are the change in character case.
2. If S if *false*, we perform dynamic programming using the Viterbi algorithm [18]. As this algorithm uses language model of words distributions to calculate the most probable sequence, we have used an English corpus[1] from which we computed word frequencies.

The hashtag segmentation component takes the compound hashtag and the words distribution model as input, and converts the hashtag to a vector of words composing them.

G. Stop Words

It is a common practice to remove stop words (also known as function words) from the dataset in Twitter applications of STSS as well as traditional information retrieval systems that analyze large pieces of text [2, 19, 20]. However, while stop words are not very useful in tasks computing documents similarity, stop words carry structural information and therefore cannot be ignored in a very short text such as tweets. Nevertheless, although stop words are retained in the dataset, they should contribute less to the meaning compared to other uncommon words.

H. URLs

URLs are common in Twitter where users refer to articles, videos or images. In STSS tasks, we are interested is measuring the similarity between the short text. Therefore, URLs are removed from the dataset although they may be utilized in tasks related to word sense disambiguation, which will be further investigated in future work.

I. Mentions

Users use the @ sign to mention to other users as a way of referring or having discussions with them in a public realm (e.g. @RubyAS came yesterday). Therefore, these common Twitter conventions may be useful in modelling user behaviour or community detection applications. They do not contribute to the meaning of the text, and hence are replaced with the string 'USER' to refine the tweet content.

[1] http://norvig.com/big.txt

J. Re-tweets

In Twitter, the "retweet" option allows users to share other user's tweets and consequently generating redundant information. Retweets are therefore removed for two reasons:

1. Retaining them in the dataset will result in an increased feature space.
2. Introducing bias when transforming the dataset into a corpus to compute information contents of terms. Distinctive terms that carry rich meaning will contribute less to the similarity score because they appear in retweets and thus weigh less, yielding misleading results.

K. Apostrophes

This step aims at reducing word sense disambiguation by means of structure. It involves converting apostrophes to its standard lexicon (e.g. *should've* becomes *should have*). This is particularly important to avoid confusion between contractions and possessiveness (e.g. *it's* versus *its*).

L. Stemming and Lemmatization

Stemming and lemmatization are special forms of normalization. They aim to reduce inflectional morphology of words through identifying a canonical representative as a common base form for a set of related word forms. The choice of employing either technique is a trade-off between effectiveness and efficiency. Stemming employs a crude heuristic operating on a single word without accounting for the context, and therefore does not take into consideration part of speech tags to discriminate between them. Although stemmers are faster and easier to implement, we use lemmatization to reduce the feature space as it operates based on a vocabulary and morphological analysis of a word form to link it back to its lemma. For example, the word "worst" has "bad" as its lemma. As this link requires a dictionary lookup, it is missed by stemming. We use WordNet [21] for our lemmatization algorithm as a lookup for word roots in order to reduce the feature space by unifying multiple word forms.

M. Numbers

Unlike most pre-processing strategies followed by researchers that remove numbers, as with stop words, we keep numbers because they carry information and contribute to the meaning of a very short text such as a tweet. Dealing with a number as a strings or as an integer is the work of the similarity measure. In the experiment, we handle a number as strings of unigrams.

N. Slangs

While being of high value for sentiment analysis applications, words that contain repeated letters, such as "looooove" do not carry much information for a STSS measure to capture the similarity. Therefore, these words are standardized by reverting them to their original English form to allow an algorithm to recognize and identify them.

O. Emoji

Emoticons are retained as they carry structural information which may be part of a syntactical function that contribute to the overall similarity computation.

IV. Experimental Methodology

The goal of the current research is to propose a pre-processing methodology that enhances the performance of STSS measures. This section describes the experiment conducted to evaluate the effectiveness of our pre-processing methodology on the performance of a textual similarity measure.

A. Dataset

Due to the lack of benchmark datasets of human scored similarity labelled tweets, we used one dataset for the evaluation experiment. Part of the SemEval-2014 shared task published a trial gold standard tweet-news dataset of 750 annotated pairs [13]. This benchmark adopted a 5-point Likert scale to measure the degree of similarity score between pairs. People undertaking the experiment were requested to assign each pair a similarity score as defined by Agirre [22]:

(0) On different topics.
(1) Not equivalent, but are on the same topic.
(2) Not equivalent, but share some details.
(3) Roughly equivalent, but some important information differs/missing.
(4) Mostly equivalent, but some unimportant details differ.
(5) Completely equivalent, as they mean the same thing.

B. STSS Measure

To assess the effect of pre-processing on an STSS measure, we used cosine similarity on a *tf-idf* weighted corpus to scale down the value of common occurring words and scale up the value of rare words. We used the scikit-learn Python library to perform the vectorization and weighting. Given two tweets, T_1 and T_2, we derive a joint feature vector V that is composed of the unique unigrams in $T1$ and T_2. T_1 and T_2 are then represented by v_1 and v_2 respectively, which are frequency vectors calculated based on V. The cosine similarity is then computed between v_1 and v_2.

C. Baseline and Evaluation Criteria

The baseline method for performing pre-processing is the classic method (C-Method) using N-grams, which has been used in most STSS approached [13, 23]. This method applies six classical pre-processing steps, including removing URLs, removing stop words, removing numbers, standardizing words, and removing punctuations. The evaluation metrics are also computed for the raw data.

A good predictive model is one with high correlations and low error rates. Therefore, the Pearson correlation coefficient and error rates were selected to evaluate the overall performance of the STSS measure as follows:

- Correlations are used to detect whether a linear relationship can be modelled between the actual (human) and estimated (STSS measure) readings. The effect of

the pre-processing techniques are assessed by a comparison of the correlations between the human judgments and the estimations recorded by the measure for the baseline and the proposed methodology.

- Error rates are negatively oriented scores that are used in predictive modelling. In addition to correlations, the mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE) were calculated. As MAE does not make use of square, it is much robust to outliers, whereas MSE emphasizes the extremes. This means that the square of a very small number (smaller than 1) is even smaller, and the square of a big number is even bigger. The root of MSE gives a relatively high weight to large errors and therefore is also included in the evaluation criteria.

V. Experiment Results and Discussion

In this section, we report the results obtained on raw data before and after the application of our proposed methodology and the baseline applied individually. The baseline (C-Method) is the method that applies the classical pre-processing steps as described in section IV.*C* and our proposed methodology described in section III to the SemEval 2014 trial gold standard tweet-new dataset. The cosine similarity measure was computed on all pre-processing approaches and the impact is analyzed and assessed through computing the evaluation criteria discussed in section IV.*C*.

TABLE II. Results of Evaluation Critera for Baseline and Proposed Pre-processing Techniques

Pre-processing Method	Correlation	MAE	MSE	RMSE
Raw Data	0.7017	1.1296	2.0281	1.4241
C-Method	0.7264	1.1288	1.94	1.3928
Our Method	**0.7585**	**1.0759**	**1.7425**	**1.32**

Table II demonstrates the performance of the cosine similarity measure depending on the pre-processing method applied. Regarding the pre-processing representations, the measure's behaviour is not uniform. It is apparent that our proposed methodology brings systematically better results in comparison with the baseline.

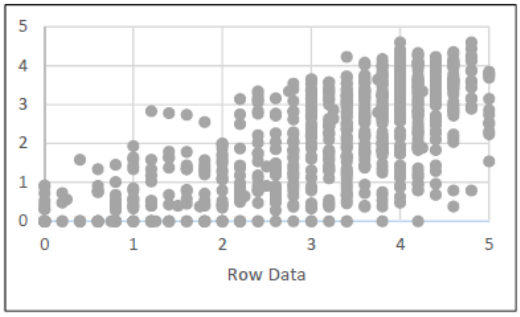


Fig. 3. Row-human data correlations scatterplot

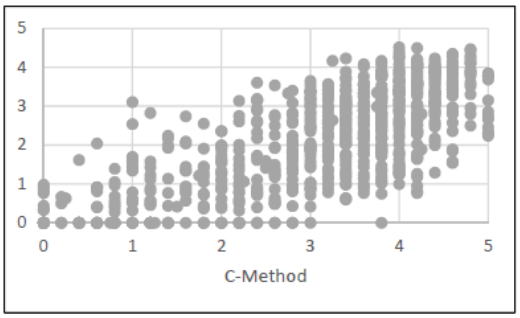


Fig. 4. C Method-Human correlations scatterplot

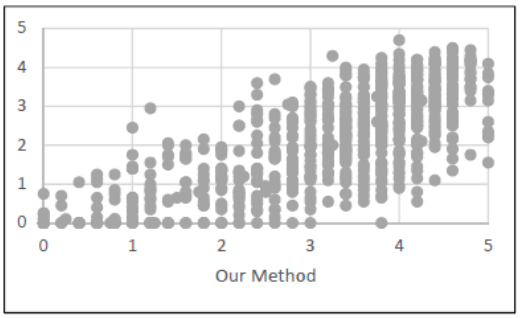


Fig. 5. Our Method-Human correlations scatterplot

The evaluation results indicate that our proposed method outperforms the baseline in terms of correlation and error rates. It 0.03 more correlated to human readings than the C-Method and 0.06 compared to raw dataset. Fig. 3 shows the correlation scatterplots between the actual and estimated values. With regards to error rates, our method generates the least variance among the others. By observing the readings of MAE and MSE, it can be concluded that the dataset has lots of outliers. This is because MSE is 0.7 higher than MAE which is more robust to outliers.

While the overall evaluation results may indicate low accuracy of the similarity measure, the purpose of this research is not to evaluate the performance of the similarity measure. It is aimed at evaluating the effect of the proposed pre-processing methodology in enhancing the results of the similarity measure compared to common practices of pre-processing (C-Method).

VI. Conclusion

In this paper, we proposed a pre-processing methodology for enhancing the performance of STSS measures. This methodology is composed of several heuristic-based preprocessing steps that were configured upon empirical experiments. We conducted an experiment using the cosine angle as the similarity measure to verify the effectiveness of our proposed method against the baseline on a Twitter labelled

dataset. Experimental results showed evidence that our methodology outperforms the current state-of-the-art in terms of correlation and error rates.

Towards proceeding with further research, the evaluation results revealed key information regarding the importance of the pre-processing stage in leveraging the performance of measuring the similarity between microblogs textual data, such as Twitter. This research indicates promising results of data quality in the context of twitter-bases similarity and paraphrase identification.

REFERENCES

1. Jianqiang, Z. and G. Xiaolin, *Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis*. IEEE Access, 2017. **5**: p. 2870-2879.
2. Satyapanich, T., H. Gao, and T. Finin. *Ebiquity: Paraphrase and semantic similarity in Twitter using skipgrams*. in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. 2015.
3. Sultan, M.A., *Short-Text Semantic Similarity: Algorithms and Applications*. 2016, University of Colorado at Boulder.
4. Saif, H., et al., *On stopwords, filtering and data sparsity for sentiment analysis of twitter*. 2014.
5. Jianqiang, Z. *Pre-processing boosting Twitter sentiment analysis?* in *Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on*. 2015. IEEE.
6. Haddi, E., X. Liu, and Y. Shi, *The role of text pre-processing in sentiment analysis*. Procedia Computer Science, 2013. **17**: p. 26-32.
7. Singh, T. and M. Kumari, *Role of text pre-processing in twitter sentiment analysis*. Procedia Computer Science, 2016. **89**: p. 549-554.
8. Zhang, Z. and M. Lan. *Estimating Semantic Similarity between Expanded Query and Tweet Content for Microblog Retrieval*. in *TREC*. 2014.
9. Xu, W., C. Callison-Burch, and B. Dolan. *SemEval-2015 Task 1: Paraphrase and semantic similarity in Twitter (PIT)*. in *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*. 2015.
10. Biçici, E. *RTM-DCU: Predicting semantic similarity with referential translation machines*. in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. 2015.
11. Steiger, E., et al., *Twitter as an indicator for whereabouts of people? Correlating Twitter with UK census data*. Computers, Environment and Urban Systems, 2015. **54**: p. 255-265.
12. Bao, Y., et al. *The role of pre-processing in twitter sentiment analysis*. in *International Conference on Intelligent Computing*. 2014. Springer.
13. Guo, W., et al. *Linking tweets to news: A framework to enrich short text data in social media*. in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2013.
14. Brown, P.F., et al., *Class-based n-gram models of natural language*. Computational linguistics, 1992. **18**(4): p. 467-479.
15. Wang, Z., et al., *TwiInsight: Discovering Topics and Sentiments from Social Media Datasets*. arXiv preprint arXiv:1705.08094, 2017.
16. Antenucci, D., et al., *Classification of tweets via clustering of hashtags*. EECS, 2011. **545**: p. 1-11.
17. Fócil-Arias, C., et al., *A tweets classifier based on cosine similarity*.
18. Forney, G.D., *The viterbi algorithm*. Proceedings of the IEEE, 1973. **61**(3): p. 268-278.
19. Yoon, S., N. Elhadad, and S. Bakken, *A practical approach for content mining of tweets*. American journal of preventive medicine, 2013. **45**(1): p. 122-129.
20. Shah, C., *INLS 490-154W: Information Retrieval Systems Design and Implementation. Fall 2009*. 2008.
21. Miller, G.A., et al., *Introduction to WordNet: An on-line lexical database*. International journal of lexicography, 1990. **3**(4): p. 235-244.
22. Agirre, E., et al. *Semeval-2012 task 6: A pilot on semantic textual similarity*. in *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. 2012. Association for Computational Linguistics.
23. Hajjem, M. and C. Latiri, *Features extraction to improve comparable tweet corpora building*. JADT Acte, Nice, France, 2016.

A Word Embedding Model Learned from Political Tweets

Noufa N. Alnajran, Keeley A. Crockett, David McLean, Annabel Latham
Department of Computing, Mathematics, and Digital Technology
Manchester Metropolitan University
Noufa.alnajran@stu.mmu.ac.uk, {k.crockett, d.mclean, a.latham}@mmu.ac.uk

Abstract— **Distributed word representations have recently contributed to significant improvements in many natural language processing (NLP) tasks. Distributional semantics have become amongst the important trends in machine learning (ML) applications. Word embeddings are distributed representations of words that learn semantic relationships from a large corpus of text. In the social context, the distributed representation of a word is likely to be different from general text word embeddings. This is relatively due to the unique lexical semantic features and morphological structure of social media text such as tweets, which implies different word vector representations. In this paper, we collect and present a political social dataset that consists of over four million English tweets. An artificial neural network (NN) is trained to learn word co-occurrence and generate word vectors from the political corpus of tweets. The model is 136MB and includes word representations for a vocabulary of over 86K unique words and phrases. The learned model shall contribute to the success of many ML and NLP applications in microblogging Social Network Analysis (OSN), such as semantic similarity and cluster analysis tasks.**

Keywords– Word Embedding, Language Modelling, Deep Learning, Social Network Analysis, Twitter Analysis

I. Introduction

The concept of "word embedding" is based on the linguistic distributional hypothesis that words occurring in similar contexts tend to have similar meanings [1]. However, the curse of dimensionality have always been a fundamental issue in most language modelling and learning representations. High dimensionality usually require thousands or millions of dimensions for sparse word vectors [2], which experience memory latencies when traversing the sparse data structures. Word-embedding models are less prone to this problem as they are generally composed of dense continuous-valued vector representations. These representations are produced such that vectors that are closer to each other in the vector space should represent words with similar meanings. Conventional word frequency models such as bag-of-words (BOW) fail to capture the semantic distances between words. That is, words *write*, *draw* and *drive* are considered equally distant despite the fact that *write* is semantically less distant to *draw* than it is to *drive*. The training iterations in the neural embedding model updates the context entries in the words' associated embedding vectors, which leads to the pre-trained model recognizing the little semantic distance between *draw* and *write* in the vector space.

Word embedding models have shown significant improvements in the performance of many NLP applications such as sentiment analysis [3-5] and text classification [6-8] and recommendation [9]. However, the lack of neural embedding models trained on social corpora has led microblogging computational linguistic related tasks to use embedding models trained on general data. These models represent words vectors according to the appearance of the words in a more formal context compared to their colloquial use in social contexts, such as Twitter. Tweets have unique lexical and structural features that are different from general English texts found in traditional documents. Out-of-vocabulary (OOV) words are prevalent in tweets, which are not found in models trained on general text corpora. The user generated content found in microblogging OSN, particularly Twitter, is usually a fertile environment for noise and common user conventions and emoticons. The informal nature of this social medium and the character limit restriction lead people to cut off conjunctions, pronouns, and substitute expressive terms with emoji in order to, ultimately, use the allowed range of characters in delivering the intended meaning. These special features of short texts posted in microblogs require NLP applications to have embeddings that model the behavior of words used in the social context.

This paper presents the methodology and training of a political word embedding model learned from a corpus of over four million political tweets. Politics is an active domain in Twitter and a rich source of controversial views. The EU Referendum event that took place on 23rd of June 2016 was targeted for data collection. The dataset not only included political news tweets and tweets related to politicians, but also daily chitchat on people's views and expectations on the event of "Brexit". However, vectors that are generated from raw tweets generally exhibit lots of noise and introduce inaccuracies to target applications. Therefore, due to the high level of noise and redundancy in Twitter, the collected tweets underwent several pre-processing stages in order to construct a rich corpus of positive examples, from which an accurate embedding model can be generated. In this research, the authors aim to train a model to learn embeddings not only for unigrams (i.e. single tokens), but for bi-grams (two-word phrases) as well. Phrases are commonly observed as hashtags in Twitter, particularly in the domain under consideration. Therefore, it is important to learn embeddings for these phrases such as *EU referendum*, *vote leave*, *stronger in*, etc. instead of each word separately. Towards detecting possible bi-grams, this research computes the probabilities of words occurring together using the Chi-squared test.

In the proposed embedding model, the iterative learning process is computed based on implementing a single hidden layer NN that generates word vectors encoding linguistic regularities and patterns, which can be represented as linear translations. The major contributions of this paper are demonstrated as follows:

1. Streaming real-time tweets in the political domain and constructing a preprocessed corpus of over four million tweet and 12.3 million words and phrases.
2. Generating a word embedding model that is learned from the constructed corpus, which shall be useful for different computational linguistic and NLP tasks in the context of microblogging social media posts, particularly tweets.

The rest of the paper is organized as follows: Section II presents related work in the field, Section III describes methods used in this study, Section IV represents the data collection and corpus construction methodology, Section V demonstrate the learned embedding model. Finally, the conclusion and future work are provided in Section VI.

II. Related Work

Language modelling and word embedding models have become a subject of intense discussion. Previous work have been investigating the significance of dense vector models in reducing the curse of dimensionality and improving the performance of NLP applications. In this section, we review the related work that were conducted in this field and discuss limitations and potential research extensions.

An early language model have been proposed by Bengio, Ducharme [2] and Schwenk and Gauvain [10]. The authors proposed a neural embedding model that estimates the probability of a word based on a context window of previous words in a sentence. The model estimates conditional probabilities of words in order to learn: 1) a distributed representation for each word, 2) the probability function for word sequences using a corpus of over 1 million examples. Collobert, Weston [11] introduced C&W deep learning model based on a convolutional neural network (CNN). The CNN learns word embedding vectors based on the syntactic contexts of words. Towards a generalizable model that can handle a number of NLP tasks, the authors performed unsupervised training on the entire Wikipedia corpus, which contains about 631 million words. Although these approaches represent vectors with less dimensionality than one-hot encoding, there is large room for model improvement in term of scalability and computational efficiency.

The revolution of digital user generated content in the era of big data has contributed to further implementations of language models. These neural embedding models have improved the learning speed and capacity in order to handle corpora with thousands of millions of words. Mikolov, Chen [12] introduced a Word2Vec model representing words as real-valued vectors. Word2Vec can have two training architectures, 1) the continuous bag-of-words (CBOW) and 2) the Skip-gram model. Based on the Skip-gram model, the authors published a pre-trained model on a Google News corpus. This model contains 300-dimention vector representations that capture both syntactic and semantic word relationships for the 1 million most frequent words in that corpus. On the other hand, the Global Vectors for Word Representation (GloVe) [13] is an extension to the Word2vec model, which rather than using a window to define local context, GloVe uses a statistical computation across the entire corpus in order to construct an explicit word co-occurrence matrix. Word2vec and GloVe have demonstrated better performance than traditional embedding models such as LSA in the field of topic segmentation [14]. Furthermore, compared to GloVe, Word2Vec produces better word vector representations with a small dimensional semantic space.

In terms of embedding models generated from microblogging social media posts, there is not much research conducting in this area. Tang, Wei [4] extended the word embedding model presented by Collobert, Weston [11]. The authors developed three neural networks to effectively capture sentiment-specific word co-occurrences learned from tweets. The artificial NN are trained through incorporating the sentiment information into the networks' loss functions. The training was performed on a corpus of distant-supervised five million positive and five million negative tweets with emoticons. The effectiveness of the model was demonstrated by using it as a feature in a sentiment classification task. The evaluation was performed using the benchmark dataset of SemEval-2013 [15] and verified by measuring sentiment lexicon similarity. Li, Shah [16] presented several embedding models trained on tweets as well as general text corpora. The authors trained NN models on both raw and pre-processed tweets and demonstrated that the latter generally performs better in tasks related to tweet semantic topic identification. The models were extrinsically evaluated on two tasks, which are sentiment analysis and topic classification. Results show that combining tweets and general texts improves the word embedding quality in terms of the topic classifier performance.

It has been observed from the literature around word embedding models in microblogging OSN that there is a lack of pre-trained models learned from domain-specific social media text corpora. This paper presents the training process and methodology of a NN embedding model that generates real-valued vectors from a corpus of political controversial tweets.

III. Methods

This section briefly describes the general methodology undertaken towards building the word embedding model learned from the Twitter-based corpus under consideration.

A. Data Collection and Storage Layer

This layer involves setting up the Twitter Streaming API and its configuration on the political domain for data collection. The streamed tweets are stored in Mongo DB NoSQL database on the flow. That is, in a real-time mode rather than storing them to an external file and transferring them to Mongo DB in batches afterwards.

B. Corpus Manipulation Layer

The input to this layer is the raw tweets obtained from the previous layer. Corpus manipulation includes pre-processing steps including n-gram identification and corpus annotation.

C. Neural Embedding Layer

In this layer, the actual training of the word embedding model is performed on the pre-processed and annotated corpus. The goal is to learn the weights of the neural networks hidden layer, which are actually the distributed word representations.

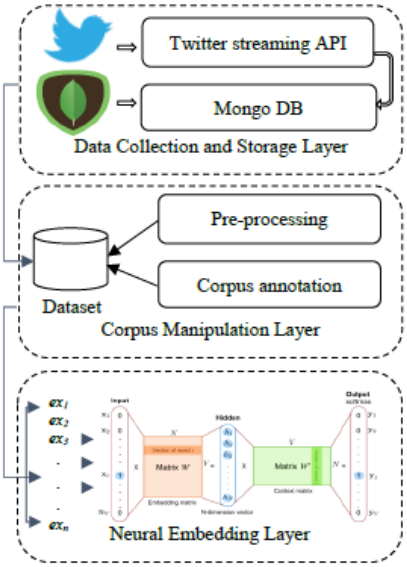


Fig. 1. Layers of the Twitter-based word embedding model framework

Figure 1 shows a hierarchical representation of the layers in the proposed model's training process. The processes undertaken in each layer and the training configurations are further elaborated in the subsequent sections.

IV. Building the Word Embedding Dataset

In this section, the methods used for building the Twitter-based political corpus, through data collection, harvesting and cleaning, where tweets undergo several pre-processing stages before getting into the learning iterations are described.

A. The Domain

In this study, the political domain of the EU Referendum is considered as it has been an active trend in OSNs and a rich source of controversial views. The United Kingdom European Union Membership (known as *EU Referendum*) took place on June 23, 2016 in the UK. Based on a voting criteria, the voters were exposed to two opposing campaigns supporting remaining or leaving the EU. Three months prior to the day of the referendum, the data collection process has commenced using the Twitter Application Programming Interface (API), and continued until one month past that day.

B. Data Collection

The Twitter streaming API allows for establishing a connection and continuously streaming real-time tweets according to a predefined set of search terms. Communicating with the Twitter platform was made possible via the Open Authentication (OAuth) mechanism. This mechanism requires an application registration on the Twitter platform beforehand. Kumar, Morstatter [17] provided a comprehensive overview of the authentication process required by the Twitter API. Amongst various programming languages that interface with the API, Python has been used for its flexibility and prebuilt selection of Twitter software packages and NLP libraries.

Twitter streamed instances are returned as JavaScript Object Notations (JSON) data structures, which are composed of multiple metadata per tweet. These JSON objects were stored in a NoSQL database called MongoDB [18]. MongoDB is used as it is a fully scalable non-relational database, intended for storing unstructured data, such as text, as documents instead of tuples in tables. It has been trusted by several web 2.0 big data sites such as Foursquare, Disney Interactive Media Group, The Guardian, GitHub, and Forbes [18]. The entire 1.2TB text corpus of Wordnik online social dictionary [19] is also stored in over five billion MongoDB records. In this study, the documents inserted into MongoDB are the tweets JSON objects that were retrieved by Twitter API.

```
client = MongoClient('localhost', 27017)
db = client['twitter_db']
collection = db['twitter_collection']
tweet = json.loads(data)
collection.insert(tweet)
```

Fig. 2. Tweets streaming and storing script

The Python-based implemented code snippet for retrieving JSON objects from the Twitter streaming API and storing them in a MongoDB database is shown in Figure 2. In a relational database, *twitter_db* would be the name of the database instance and *twiter_collection* would be the table in which the data objects are stored.

C. Dataset Size and Features

Following the data collection methodology described in Section IV.B, a dataset of four million tweets have been collected and stored in MongoDB. Each instance in the dataset is a tweet associated with multiple metadata. These metadata (i.e. features) contain information relating to the text of a tweet, users, and entities. Tweets are associated with multiple features that represent their syntactic and semantic status. However, this research is concerned with the textual features that make up a tweet. These are the features from which the embedding vectors will be generated.

The collected raw tweets had undergone preliminary scraping stages as discussed in Section IV.D. The tweets semantic features are preserved, while the unwanted noise such as redundant tweets (i.e. retweets) and tweets where length is less than a certain threshold are eliminated. Tokenization and phrase identification are performed to identify *n*-gram features. The removal of reposts and non-

informative instances has reduced the dataset to one million examples.

D. Pre-processing

Accuracy of the learned word embedding vectors is linearly related to the dataset size of training examples. However, the quality of the training dataset is of no less importance than the quantity. According to the research report of The Data Warehousing Institute, '*Poor quality customer data costs U.S. business an estimated 611 billion dollars ...*' [20]. Pre-processing techniques are, therefore, a prerequisite for various information systems to maintain data quality. While structured data is usually manipulated in relational databases and schemas, features of free natural text often require special means of management and storage due to its lack of structure. Unstructured text data is highly susceptible to noise, redundancy, and inconsistency as they are generated from heterogeneous sources. Pre-processing techniques are required to remove redundancies and inconsistencies as analyzing low-quality data usually result in low-quality mining results [21].

The focus of this research is to train a neural embedding model to learn real-valued vectors from the collected tweets (i.e. examples). However, the majority of raw tweets are erroneous and highly unstructured, due to the informal nature of the communication channel in which these tweets are propagated as discussed in Section I. Therefore, in order to learn efficient embedding models that accurately capture the semantic relations between words, it is necessary to learn from clean data. The pre-processing stages carried out on the raw corpus of tweets are as follows:

1) Removal of redundant and non-informative tweets: in this stage, all duplicate tweets and reposts are excluded. Tweets that contain nothing but a URL are also removed from the dataset. Similarly, tweets that are composed of only one word are eliminated as these lack sufficient context for the embedding model to learn.

2) Removal of URLs and punctuations: URLs and punctuations such as interrogation and exclamation marks are removed. While these parts may carry structural and syntactic information for other applications, they provide nothing but noise to the learning process of the embedding model, which tries to capture the relationships and latent semantics between words.

3) Canonicalization of hashtags and mentions: common user conventions such as #hashtags and @mentions are prevalent in Twitter. Hashtags are actual words that contribute to the meaning of a tweet and may occure in a different tweet without the hash sign. For example, some tweets may contain the word *brexit* and others may contain it as *#brexit*, wich are different representations for the same word. If the hash sign is left in the training corpus, the model will generate different embedding vectors for each form of the word even though they carry the same meaning. Therefore, only the hash sign (i.e. prefix) is removed and the rest of the word is retained. Mentions in tweets are references to other users in Twitter. Different usernames do not contibute to the relationship between word. However, because tweets are very short text, the plot where a username appears has an impact on the morphological structure of the sentence. Thus, all user meantions are replaced with special symbols rather than words such as 'user', as these may appear in the corpus and therefore cannot be used for replacement.

4) Splitting joint words: joint words and hashtags such as 'BetterOffOut' and 'strongerin' are commmon in tweets due to the character limit. These are splitted following the probability driven hueristic proposed in [22]. Phrases are identified as discribed in Section IV.E.

5) Normalization of special symbols: the proposed pre-trained model is meant to learn embeddings for words and thus, all integer and decimal numbers are replaced with special characters.

The pre-processing stages discussed in this section are performed on each instance retrieved by the Twitter streaming API. Considering the raw tweet, T, and the preprocessed version of it, $\check{T}$, in the following illustrative example:

T: *#skydebate #EUvote The more people like @Barak_Obama stick their noses in to #Brexit vote, the more I want to vote #leave*

Ť: *sky debate EU vote The more people like xxx stick their noses in to Brexit vote, the more I want to vote leave*

These consecutive steps aim at reducing confusion during the learning iterations and consequently, generating efficient embedding vectors, which shall contribute to the enhancement of social media NLP applications.

E. N-gram Identification and Corpus Annotation

Theoretically, training a NN embedding model assuming all words in the corpus are isolated from each other is memory intensive [23]. Additionally, many phrases have a single meaning that is not simply a composition of the meaning of its individual words, such as 'New Jersey'. In this research, the authors perform a composite method that commence with detecting common phrases in the pre-processed tweets, then annotating the corpus with these words that are most likely phrases.

1) Discovering phrases in the corpus: the data driven approach used in [23] for identifying phrases in a corpus is followed. In this approach, phrases are identified based on the frequently occurring bigrams that are commonly embedded in discourse, such as 'vote leave' and 'stronger in'. The following formula is used:

$$Score\left(w_i, w_j\right) = \frac{freq(w_i w_j) - \delta}{freq(w_i) \times freq(w_j)} \qquad (1)$$

Where w_i and w_j are words occurring in a phrase and δ is a discounting coefficient that prevents bigrams of infrequently occurring words to be considered as phrases. The bigrams of frequency scores above the predefined threshold are formed as phrases.

2) Corpus tagging: this process involves annotating the corpus with the two-word phrases identified in the previous step. The words that make a phrases are joined using the underscore character. For example, '*… visited New York and San Francisco…*' would become '*…visited new_york and san_francisco…*'. Finally, the resulting corpus consists of unigrams and explicitly tagged bigrams.

V. The Word Embedding Model

This section describes the methodology undertaken in constructing the word embedding model and learning from the political tweets corpus that was collected and pre-processed ad discussed in Section IV.

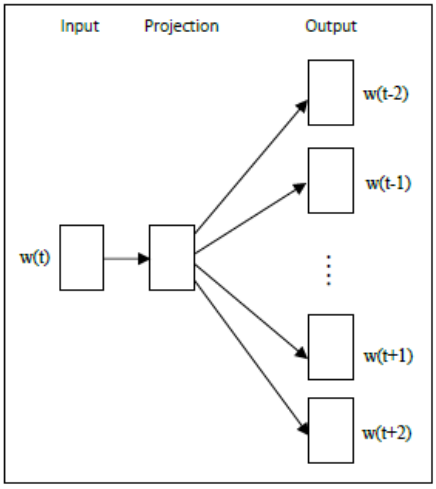


Fig. 3. Skip-gram model architecture [12]

A. Vocabulary Trimming

A vocabulary of 12.3 million words and phrases are included in the corpus. However, this vocabulary may contain rarely occurring words that lack enough context. Therefore, the minimum word frequency threshold is set to *min_count* = 3. Words and phrases that do not satisfy the *min_count* are discarded due to two reasons: 1) the NN model does not have adequate trainings examples to learn meaningful embedding vectors for those words, and 2) through performing basic corpus statistics, words occurring less than 3 times in the entire corpus appear to be mostly typos. The value of the *min_count* threshold has been determined empirically. The application of the minimum frequency threshold has generated a vocabulary V = 86K unique words and phrases in the training embedding model.

B. Model Architecture and Hyperparameter Configuration

In this research, a Word2Vec Skip-gram NN model with negative sub-sampling is used [23]. The use of the Skip-gram model and sub-sampling frequently occurring words decreases the number of training examples, and consequently, reduces the computational burden of the training process. This model is a shallow neural network with a single hidden layer. The learning process is unsupervised, in which the goal is to learn the weights between the input layer and the hidden layer that are actually the embedding vector representations of words. This is similar to the unsupervised feature learning in training an auto-encoder. The architecture of the implemented neural network model is shown in Figure 3.

1) Input layer: in this layer, the training examples (i.e. word pairs) are fed into the network. It has been reported that a context window size of w' = 5 is considered a good trade-off between efficiency and accuracy [16]. Empirical experiments were performed on different window sizes, $w' \in \{3, 4, 5, 6\}$ and have shown that w' = 5 produces the best embedding vectors for tweets. The output probabilities predict the likelihood of a word occurring in the domain of the input word (i.e. the word's context window). For example, training the network on the word 'TTIP', which is a typical acronym of *translatic trade and investment partnership* in the event of brexit, the output probabilities are higher for words like 'trade' and 'union'.

TABLE 1
Illustrative Example of Model's Training Input for w' = 5

Sliding window (w' = 5)	Target word	Context
[Brexit issue no organization afford to]	Brexit	issue, no, organization, afford, to
[Brexit issue no organization afford to ignore]	Issue	Brexit, no, organization, afford, to, ignore
[Brexit issue no organization afford to ignore]	No	Brexit, issue, organization, afford, to, ignore
[Brexit issue no organization afford to ignore]	Organization	Brexit, issue, no, afford, to, ignore
[Brexit issue no organization afford to ignore]	Afford	Brexit, issue, no, organization, to, ignore
[Brexit issue no organization afford to ignore]	To	Brexit, issue, no, organization, afford, ignore
[issue no organization afford to ignore]	Ignore	issue, no, organization, afford, to

Considering *T, 'Brexit issue no organization afford to ignore'* as an example tweet in the annotated corpus described in Section IV.E, the training samples for *T* at w' = 5 are shown in Table 1. Subsampling is implemented to eliminate highly frequent words with marginal information content, such as 'the'. The probability, p, of which a given word is retained in the vocabulary, is calculated as follows:

$$p(w_i) = \left(\sqrt{\frac{z(w_i)}{0.001}} + 1\right) \times \frac{0.001}{z(w_i)} \quad (2)$$

$$p(w_i) = \begin{cases} 1, & z(w_i) < 0.0026 \\ 0.5, & z(w_i) = 0.00746 \\ 0.033, & (w_i) = 1.0 \end{cases}$$

Where $z(w_i)$ is the fraction of the total occurrence of the word w_i in the corpus and the sample value of 0.001 is the default sampling parameter [22].

2) Hidden layer: in this layer, the dimensions of the embedding vectors is set to d = 300. That is, the configured model is learning word vectors with 300 features instead of the high dimensional vocabulary size. The hidden layer is thus represented by a weight matrix A (86K×300d), with 86K raws

(1 per each record in the vocabulary) and 300 columns (1 per each hidden neuron).

3) Output layer: A vector for each word in the vocabulary acts as an input to the output layer. To optimize the computational burden in this layer, a negative sampling is performed to avoid updating every neuron weights for each vector in the vocabulary during training. Rather, only a small ratio of the weights are modified by each training vector. We randomly select five negative words, in which their weights are updated as well as the weights of the word in the training iteration. It has been reported in [22] that negative sampling value of five words works well for our dataset size range. The selection of the negative samples is based on a unigram distribution approach, in which more frequent words are more likely to be sampled.

TABLE 2
CORPUS AND MODEL METADATA AND HYPER-PARAMETERS

Metadata and Hyper-parameters	Political Corpus of Tweets
Raw tweets	4 million
Pre-processed tweets	1 million
Words in the corpus	12.3 million
Unique tokens in the trained embedding model	V = 86K
Neural network architecture	*Word2Vec Skip-gram / negative sub-sampling*
Negative samples	*5*
Vector dimension	$d = 300$
Minimum frequency threshold	$min\_count = 3$
Learning context window	$w' = 5$
Training time	17 minutes
Training complexity	$O(\log_2(\|V\|))$
Trained model size	136MB
Processor and memory	intel core i7 CPU / 16GB RAM

C. Trained Model and Complexity

The model's training complexity is $O(\log_2(|V|))$, where V is the vocabulary size. Training the Word2Vec model on the political tweets dataset has taken about seventeen minutes on intel core i7 CPU and 16GB RAM. The statistical information on the learning corpus, trained embedding model, training configurations, and processor and memory specifications are shown in Table 2.

VI. CONCLUSION AND FUTURE WORK

Distributed word representations have shown to be successful in many computational linguistic applications as discussed in Section I. However, upon conducting a literature review, it has been observed that there is a lack of embedding models trained on domain specific microblogging posts, particularly tweets. This paper contributes to the literature in several significant ways. First, a corpus of over four million political tweets on the EU Referendum rich domain of controversial views is collected. The constructed corpus is pre-processed according to the methodology described in Section IV. Second, a word embedding model is trained on the collected and pre-processed corpus of tweets in order to learn meaningful words representations. The generated pre-trained model contains representations for 86K unique words and phrases.

Future work carries on as follows:

- The performance of the trained embedding model will be extrinsically evaluated through a semantic similarity measure for tweets. Alongside the NLP application evaluation, a machine learning based application of semantic cluster analysis will be carried out to evaluate the learned word vectors..
- A further extension political dataset on the event of 'leaving the EU' on March 2019 will be collected, and the trained model will be augmented to learn extended representations and increase the vocabulary size. Maintenance works to configure the artificial neural network's hyper-parameters and metadata will be performed accordingly.
- The augmented and optimised pre-trained model will be evaluated in different microblogging OSN computational intelligent applications.

VII. REFERENCES

1. Harris, Z.S., *Distributional structure.* Word, 1954. **10**(2-3): p. 146-162.
2. Bengio, Y., et al., *A neural probabilistic language model.* Journal of machine learning research, 2003. **3**(Feb): p. 1137-1155.
3. Socher, R., et al. *Recursive deep models for semantic compositionality over a sentiment treebank.* in *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013.
4. Tang, D., et al. *Learning sentiment-specific word embedding for twitter sentiment classification.* in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2014.
5. Li, Q., et al. *Tweet Sentiment Analysis by Incorporating Sentiment-Specific Word Embedding and Weighted Text Features.* in *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on*. 2016. IEEE.
6. Kim, Y., *Convolutional neural networks for sentence classification.* arXiv preprint arXiv:1408.5882, 2014.
7. Li, Q., et al. *Tweetsift: Tweet topic classification based on entity knowledge base and topic enhanced word embedding.* in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 2016. ACM.
8. Taddy, M., *Document classification by inversion of distributed language representations.* arXiv preprint arXiv:1504.07295, 2015.
9. Li, Q., et al. *Hashtag recommendation based on topic enhanced embedding, tweet entity data and learning to rank.* in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 2016. ACM.
10. Schwenk, H. and J.-L. Gauvain. *Connectionist language modeling for large vocabulary continuous speech recognition.* in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*. 2002. IEEE.
11. Collobert, R., et al., *Natural language processing (almost) from scratch.* Journal of Machine Learning Research, 2011. **12**(Aug): p. 2493-2537.
12. Mikolov, T., et al., *Efficient estimation of word representations in vector space.* arXiv preprint arXiv:1301.3781, 2013.

13. Pennington, J., R. Socher, and C. Manning. *Glove: Global vectors for word representation*. in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.
14. Naili, M., A.H. Chaibi, and H.H.B. Ghezala, *Comparative study of word embedding methods in topic segmentation.* Procedia Computer Science, 2017. **112**: p. 340-349.
15. Kozareva, Z., et al. *Sentiment analysis in twitter*. in *Proceedings of the 7th International Workshop on Semantic Evaluation. Association for Computation Linguistics*. 2013.
16. Li, Q., et al., *Data sets: Word embeddings learned from tweets and general data.* arXiv preprint arXiv:1708.03994, 2017.
17. Kumar, S., F. Morstatter, and H. Liu, *Twitter data analytics*. 2014: Springer.
18. Banker, K., *MongoDB in action*. 2011: Manning Publications Co.
19. Davidson, S., *Wordnik.* The Charleston Advisor, 2013. **15**(2): p. 54-58.
20. Eckerson, W.W., *Data quality and the bottom line: Achieving business success through a commitment to high quality data.* The Data Warehousing Institute, 2002: p. 1-36.
21. Ciszak, L. *Application of clustering and association methods in data cleaning*. in *Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on*. 2008. IEEE.
22. Alnajran, N., et al. *A Heuristic Based Pre-processing Methodology for Short Text Similarity Measures in Microblogs*. in *High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2018 IEEE 20th International Conference on*. 2018. IEEE.
23. Mikolov, T., et al. *Distributed representations of words and phrases and their compositionality*. in *Advances in neural information processing systems*. 2013.

A Comparison of Unsupervised Learning Algorithms and Challenges in Microblogging Data Analysis

Noufa Alnajran, Keeley Crockett, David McLean and Annabel Latham
School of Computing, Math and Digital Technology, Manchester Metropolitan University
John Dalton Building, All Saints, Manchester, M1 5GD, U.K.
Noufa.alnajran@stu.mmu.ac.uk, (k.crockett, d.mclean, a.latham)@mmu.ac.uk

Abstract The microblogging Online Social Network (OSN), Twitter, has quickly gained considerable prominence recently. This is due to its distinctive features providing people with the opportunity to communicate efficiently and share posts and topics. The process of automated analysis and reasoning of this user generated content has tremendous value in deriving meaningful insights. These insights carry potential opportunities for businesses, users, and consumers. However, the nature of the data propagated in such informal communication medium impose several challenges hampering the efficiency of traditional unsupervised algorithms, such as observing groups of data (i.e. clusters) with minimum variance. These challenges include sheer volume, noise, sparseness, credibility, and dynamism of short-text messages. This in-depth review focuses on research that has used various clustering algorithms to analyze Twitter data streams where text is highly unstructured. Additionally, it discusses different approaches to alleviating textual challenges. This paper performs a systematic comparative analysis through establishing a generalized cluster analysis comparison criteria, which is applied to analyze Twitter-based unsupervised learning applications. A review of the literature identified fifteen studies that implemented different clustering methods. This novel comparison on multiple criteria has been conducted to allow for a thorough analysis. These criteria included clustering methods, algorithms, number of clusters, dataset(s) size, distance measure, clustering feature set, evaluation methods, and results. The conclusion reports current shortcomings and general recommendations for applications of unsupervised learning in OSN. Success criteria and future directions for research and practice communities are further discussed.

Keywords Unsupervised Learning, Clustering, Social Network Analysis, Natural Language Processing, Computational Linguistics, Machine Learning

1 Introduction

Web 2.0 technologies such as the rapid evolution of OSN applications, has led to the continuous generation of massive volumes of digital heterogeneous data being published at an unprecedented rate. These technologies have been very useful throughout multiple domains as they have significantly changed the way people communicate and share information. A lot of people have shifted from traditional media channels such as televisions and newspapers, to online social media. In this context, Twitter has gained tremendous popularity as it provides an informal and simple platform where people can easily publish and broadcast messages on different worldwide areas. It had an important role in spreading awareness of natural disasters

such as Hurricane Sandy and socio-political events such as the Arab Spring (Kumar et al., 2013). This has made Twitter an important source of information for synthesizing evidence in argumentation, and a goldmine of potential cross domain opportunities for both businesses and decision makers. However, the exponential amount of user-generated content on this site is too vast for manual analysis. More than 500 million short-text messages, referred to as "tweets", are published every day (Krestel et al., 2015). This requires an automated and scalable mining process to discover patterns in the unstructured data.

Clustering is a prominent component of exploratory data analysis. It is the unsupervised process of grouping data instances into relatively similar categories, without prior understanding of the groups' structure or class labels (Han et al., 2011). A subfield of clustering includes text mining, where large volumes of text are analysed to find patterns between documents (Godfrey et al., 2014). The growth of these unstructured data collections, advances in technology and computer power, and enhanced software capabilities, has made text mining an independent academic field. Moreover, the emergence of OSNs has yielded new frontiers for academic research, where researchers in the broad area of Natural Language Processing consider text analysis one of the most important research areas. Recent studies in various disciplines have shown increasing interest in micro-blogging services, particularly Twitter (Sheela, 2016). The applications of text mining tools for studying features of content and semantics in tweets propagating through the network has been widely studied (Kumar et al., 2013).

Several studies have aimed at analysing social data from Twitter through performing data mining techniques such as classification (Castillo et al., 2011). However, these techniques could be considered to have limited capabilities due to the unpredictable nature of the dataset. Applications of unsupervised algorithms on tweets have been reported to be particularly suitable for this kind of data for two reasons (Go et al., 2009): (1) the amount of data required for training is too vast for manual labelling. (2) The nature of the data implies the existence of unforeseen groups that may carry important nuggets of information, which can only be revealed by unsupervised learning. The main purpose of this paper is to:

- Provide a thorough insight on the different textual challenges presented in social media data, particularly Twitter, and discuss various approaches from the literature to alleviate them.
- Review wide range of clustering algorithms that were implemented on different features of Twitter datasets.
- Review various applications, domains, and success criteria that are used for measuring and evaluating the algorithms' performance in terms of resource consumption.
- Compare relevant approaches in terms of clustering methods, algorithms, number of clusters, dataset(s) size, distance measure, clustering features, evaluation methods, and results.
- Recommend future directions for research and practice to the research community.

(Jain et al., 1999, Xu and Wunsch, 2005, Berkhin, 2006) There are very limited research that reviews the prominent clustering algorithms available to use on challenging, large, and unstructured data such as Twitter (Alnajran et al., 2017). This review extends the work presented by Alnajran et al. (2017) and provides a comprehensive review of more expanded unsupervised approaches to analysing Twitter textual datasets. Moreover, in this novel and comprehensive review, the different challenges that hinder traditional unsupervised algorithms

from performing as well on such data are discussed and approaches to mitigate each challenge are provided. Therefore, this new review adds value to the literature as it:

- Establishes a generalized comparison criteria, upon which a systematic comparison and generalized conclusions are derived.
- Discusses the main challenges faced by unsupervised analytical algorithms in social textual data.
- For each of these challenges, provide different alleviating practices in the context of the unstructured textual data published and propagated through social communication channels.
- Investigates applications of graph-based unsupervised algorithms to the comparative breadth of unsupervised learning approaches under consideration.

Therefore, this paper integrates and provides a comprehensive literature review and a valuable source of information on the state of the art for relevant research in an interdisciplinary field.

The rest of this paper is organised as follows: section 2 describes the methods that are used in this review. Section 3 explains the challenges faced in clustering social data and reviews approaches for alleviating each challenge. Section 4 includes the approaches for mining Twitter datasets that use five clustering methods: (1) partition-based (hard and fuzzy), (2) hierarchical-based (agglomerative and divisive), (3) density-based, (4) graph-based, and (5) hybrid-based. Section 5 contains the discussion and section 6 has the conclusion and future work. A table providing a summary of the studies featured in this review is located at the end of the paper.

2 Methods

2.1 Literature Search Procedures

Towards conducting this review, multiple research databases were investigated, such as Google Scholar and DeepDyve, to perform online searches. This process includes searching for the following terms: "Twitter challenges", "mining Twitter short-text", "unsupervised learning on Twitter", "clustering tweets", and "categorization of tweets".

2.2 Inclusion Criteria

The inclusion criteria for the challenges and unsupervised learning approaches in Twitter applications provided in this paper includes research that involve:

- An approach to alleviate one of the following Twitter textual challenges: sparseness, out-of-vocabulary (OOV) words, volume, and credibility.
- The development of one of the following unsupervised learning approaches: partition, hierarchical, density, graph, and hybrid, on Twitter short-text messages. These approaches cover the majority of unsupervised algorithms and, to the best of our knowledge, have not been comprehensively reviewed in the context of the informal and unstructured textual data in Twitter.
- Studies aiming to reveal hidden patterns and similar granularities in the data through applications of unsupervised learning models.

A total of five articles in relation to Twitter challenges, and fifteen articles from 2011 to present that utilized Twitter-based text mining applications using unsupervised learning have met the defined inclusion criteria for this review.

2.3 Comparison Criteria

In this study, a comparison criteria has been established to provide a systematic analysis of the unsupervised learning approaches. This criteria identifies general factors in a cluster analysis problem. Each criterion has impact on others and contributes to the overall performance of the resulting clusters.

Table 1. A general comparison criteria for unsupervised learning problems

ID	Criterion	Definition
C1	Problem Domain	The task that the clustering method is required to address. A proper understanding of the problem domain is key to the accurate decision on which unsupervised learning approach to use.
C2	Dataset Size (*dependent on C1*)	Defines the total number of objects (i.e. data points) to be clustered. No rule-of-thumb exist about the exact dataset size for cluster analysis. Decision on the sample size is a tradeoff between efficiency and effectiveness as small datasets lead to uncritical applications while large datasets raise scalability issues.
C3	Feature Set (*dependent on C1*)	An unordered list of unique variables that represent the raw data and used to build a predictive model.
C4	Distance Measure (*dependent on C1, C3*)	A method for quantifying the dissimilarity between points, which determines their cluster belongingness. Hence, *d* is a distance measure if it is a function from pairs of points to reals.
C5	Algorithm (*dependent on C1-C4*)	An automatic method of assigning data objects into homogeneous groups (i.e. clusters) and ensuring that objects in different groups are dissimilar (Aggarwal and Reddy, 2013). Clustering algorithms are generally distinguished into partition-based, hierarchical-based, density-based, graph-based, and hybrid-based.
C6	Number of Clusters (*dependent on C1, C2, and C5*)	Determines the number of clusters that will be generated. While partition-based algorithms require the number of clusters to be pre-specified, hierarchical approaches allow for selecting the number of clusters after the clustering results has been obtained. Density based clustering does not require either but require specifying the minimum number of points in a neighborhood. Clustering based on graph theory only requires a predefined distance threshold, which will determine the resulting number of clusters.
C7	Evaluation Method (*dependent on C1*)	An objective or subjective function that validates the extent to which a clustering algorithm achieves the optimal goal of attaining high intra-cluster similarity and low inter-cluster similarity.

Table 1 presents a general criteria for a systematic comparison of unsupervised learning applications. Figure 1 shows a dependency graph of the cluster analysis comparison criteria defined in table 1.

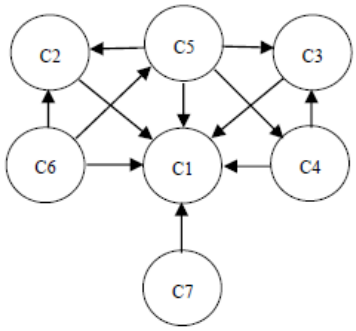


Fig. 1. Dependency graph of the cluster analysis comparison criteria

3 Challenges of Clustering Tweet Features

Most of the research conducted in clustering tweets aim to interpret these short-texts through text mining applications to relevant and meaningful information that support reasoning on potential conclusions, such as inferring users' interests and identifying emergent topics. However, several natural challenges of such data prevent standard clustering algorithms being applied with their full potentials. These text challenges present in Twitter datasets necessitate intelligent techniques and comprehensive preprocessing stages that depend on the application domain. The incorporation of statistical or ontological semantic techniques should provide dynamic algorithms that can process and analyze such complex datasets and convey meanings and correlations (Alnajran et al., 2017).

Table 2. Summary of the studies approaching Twitter challenges.

Author & Year	Alleviated Challenge	Application	Methodology	Feature Set
(Saif et al.) 2012	Sparseness	Sentiment classification	Utilization of semantic feature set and sentiment topic feature set.	Semantic and sentiment topic feature sets.
(Maity et al.) 2016	OOV	Classification	Categorization of OOV words and analysis of the feature set in the surrounding text.	Content, lexical, and context features.
(Palguna et al.) 2015	Volume	Statistics	Employing statistical metrics to quantify the representativeness of the tweet sample.	Frequent keyword identification and associated sentiments.
(Castillo et al.) 2011	Credibility	Classification	Classification of tweets related to trending topics as either credible or not credible.	Posting and retweeting behavior, text, and citations to external resources.
(Ito et al.) 2015	Credibility	Classification	Utilization of features from LDA model to recognize reliable trendy users and topics.	"Tweet topic" and "user topic".

The text challenges present in Twitter and how they have been approached in relevant research are discussed in the subsequent sections (Abbasi and Liu, 2013, Castillo et al., 2011, Ito et al., 2015, Maity et al., 2016, Palguna et al., 2015, Saif et al., 2012) and summarized in table 2.

3.1 Sparseness

Unlike traditional methods of clustering documents, which are performed on rich context, Twitter imposes a textual length restriction of 140 characters. Therefore, users tend to pro-

duce short pieces of texts that may be rich in meaning, which implies the usage of abbreviations and other syntactic conventions in order to fit the specified limit.

The problem of data sparseness has been approached by Saif et al. (2012). Aiming to train sentiment classifiers, the authors proposed an approach to alleviate this problem using two different sets of features. One is the semantic feature set, where semantically hidden concepts were extracted from tweets and incorporated into the classifier training process via interpolation. The other is the sentiment topic feature set, in which latent topics and associated topic sentiment were extracted, and the original feature space was augmented with these sentiment topics. Experimental results on the Stanford Twitter Dataset (Yang and Leskovec, 2011) have shown that the implemented criteria outperformed existing approaches achieving 86.3% accuracy.

3.2 Out-of-Vocabulary Words

The English lexicon is witnessing a high deviation from the formal written version. This is due to the language used in social media, which is mostly driven by new words and spellings that are constantly polluting traditional English. In Twitter, users have invented many ways to expand the semantics that are carried out by the short text. This includes the usage of slang, misspelled, and connected words, besides self-defined hashtags to identify topics or events. These out-of-vocabulary (OOV) words form the primary entities of such language. Examples of word lengthening OOVs include "noooo, pleaseeee, okk, and damnnn", expression OOVs include "haha, uhh, ughh, ahah, and grr", and word shortening OOVs include "lol, omg, yolo, rofl, oomf".

Maity and Chaudhary et al. (2016) studied various sociolinguistic properties of the OOV terms in order to approach this problem. They proposed a classification model to categorize these words into at least six categories, which achieved 81.26% accuracy. They observed that the content, lexical, and context features, respectively, are the most discriminative ones.

3.3 Volume

The rapid generation of user content in Twitter has led to massive volumes of unstructured data, most of which is text. The analysis of these huge streams of data for different applications require high scalability techniques, such as parallel processing, that scale well with the number of data instances. In Twitter, even using the live public streaming API, the maximum sample retrieved is approximately 1% of all tweets that are currently being published by users . Therefore, it is imperative to develop algorithms that work with the data in a scalable fashion.

This problem has been approached by Palguna and Joshi et al. (2015), in which a theoretical formulation for sampling Twitter data was proposed. In this approach, the number of samples needed for obtaining highly representative tweet samples were derived through application of statistical metrics to quantify the statistical goodness of the tweet sample. The representativeness of the sample is quantified in relation to frequent keyword identification and restoring public sentiments associated with these keywords. However, having mentioned the fact stated earlier with regards to the 1% maximum twitter feeds, this sampling approach may not be viable as it will be too small to represent the actual tweets that are published and cover the general community. Rather, Scalable methods should implement effective means to deal with the huge streams of data.

3.4 Credibility

Twitter allows users to instantly report events, news, and incidents acting as social sensors. Therefore, this platform provides first-hand data, however, distinguishing truthful information from rumors and misinformation is one critical problem (Abbasi and Liu, 2013). In most cases Twitter data is user generated and thus can be subjective, biased, and misleading. In consequence, information propagated in Twitter is not necessarily trustworthy, and therefore means of credibility assessment should be applied prior to decision making.

Castillo et al. (2011) proposed an automatic method for assessing the credibility of a given dataset of tweets. This was implementing through extracting features related to trending topics and classifying them as either credible or not. The method was evaluated through the usage of a massive number of human credibility assessments on a sample of twitter postings. However, this method can be very time consuming and the results may be biased by human subjective opinions. Ito et al. (2015) approached the credibility problem in a different way. Trendy tweets in Japan have been collected and the way people judge whether a tweet is credible or not has been analyzed. From their analysis, they derived three important factors that contribute to this judgement. These factors are, whether a tweet has an information source, whether the tweet is on a serious topic (i.e. news topics such as trends), and whether the tweet's author is reliable (i.e. whether the writer was a journalist or was available when the incident happened). This analysis forms the basis on which the assessment method was developed. The method utilizes features obtained from the Latent Dirichlet Allocation (LDA) model to recognize reliable trendy topics and users.

4 Unsupervised Mining of Twitter

Many unsupervised learning methods exist in the literature, and it is difficult to provide a crisp categorization of these methods as they may overlap and share features. Nevertheless, the most prominent unsupervised methods are included in this review (Han et al., 2011).

Clustering has been widely studied in the context of Twitter mining. It has been applied to analyze social behaviors in a variety of domains to achieve different tasks, such as tailoring advertisements for groups with similar interests (Friedemann, 2015), event detection (De Boom et al., 2015), trending issues extraction (Purwitasari et al., 2015), and prediction of micro-populations (Sinnott and Wang, 2017). This review focuses on the major clustering methods: partition, hierarchical, density, graph, and hybrid, which have been used in the context of Twitter data.

4.1 Partition-Based Clustering

Partitioning algorithms attempt to organize the data objects into k partitions ($k \leq n$), each representing a cluster, where n is the number of objects in a dataset. Based on a distance function, clusters are formed such that objects within the cluster are similar (intra-similarity), whereas dissimilar objects lie in different clusters (inter-similarity). Partitioning algorithms can be further divided into hard and fuzzy (soft) clustering. In this section, six articles are summarized in which partitioning-based clustering algorithms has been applied in the exploratory analysis of Twitter.

4.1.1 Hard Clustering

Methods of hard partitioning of data assign a discrete value label (0, 1), in order to describe the belonging relationship of objects to clusters. These conventional clustering methods provide crisp membership assignments of the data to clusters. *K*-means and *k*-medoids are the most popular hard clustering algorithms (Arora and Varshney, 2016).

K-means is a centroid-based iterative technique which takes the number of representative instances, around which the clusters are built. Data instances are assigned to these clusters based on a dissimilarity function (i.e. distance measure). In each iteration, the mean of the assigned points to the cluster is calculated and used to replace the centroid of the last iteration until some criteria of convergence is met. The square-error criterion can be used, which is defined as (Han et al., 2011),

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 \quad (1)$$

Which means that for each data point p in each cluster space, the distances from the data points to their centroids are squared and summed. This criterion aims to provide as compact and separate k clusters as possible. *K*-means has been adapted in numerous ways to suit different datasets including numerical, binary, and categorical features.

In the context of Twitter mining applications, the *k*-means approach for clustering customers of a company using social media data from Twitter was proposed (Friedemann, 2015). The technique constructs features from a massive Twitter dataset and clusters them using a similarity measure to produce groupings of users. The study performed *k*-means clustering and produced satisfactory experimental results. It is considered to be relatively computational efficient. Soni and Mathai (2015) proposed a 'cluster-then-predict' model to improve the accuracy of predicting Twitter sentiment through a composition of both supervised and unsupervised learning. After building the dataset, *k*-means was performed such that tweets with similar words are clustered together. This unsupervised phase was performed after a feature extraction process. After the clustering phase, classification was done on the same data. The data was divided into training and testing sets, with 70% and 30% of the data respectively. Finally, the Random Forest learning algorithm was used for building the learning model, which was applied to each of the training datasets individually (Breiman, 2001). This algorithm has been chosen as it provides satisfactory trade-off between accuracy, interpretability, and execution time. Empirical evaluation shows that combining both supervised and unsupervised learning (*k*-means then Random Forest) performed better than various stand-alone learning algorithms.

K-medoids is an object-based representative technique that deals with discrete data. It is an improvement to *k*-means in relation to its sensitivity to outliers. Instead of referring to the mean value of cluster objects, *k*-medoids picks the nearest point to the center of data points as the representative of the corresponding cluster. Thus, minimizing the sum of distances between each object, o, and its corresponding center point. That is, the sum of the error for all objects in each cluster is calculated as (Han et al., 2011),

$$E = \sum_{j=1}^{k} \sum_{p \in o_j} |p - o_j| \quad (2)$$

Where k is the number of clusters, p is an object in the cluster C_j, while o_j is the representative object of C_j. The lower the value of E, the higher clustering quality.

A recent study focused on the usage of *k*-medoids algorithm for tweets clustering due to its simplicity and low computational time (Purwitasari et al., 2015). In this study, the author applied this algorithm to extract issues related to news that is posted on Twitter such as "flight passengers asking for refund" in Indonesia. Their proposed methodology for Twitter trending issues extraction consists of clustering tweets with *k*-medoids, in which they divided the tweets dataset into groups and used a representative tweet as the cluster center. Terms that are related to topic issues are then selected from the clusters result and assigned higher weight values. The terms that weigh over a certain threshold are extracted as trending issues. Weight score is calculated as the frequency of word occurrences in the dataset. Average Silhouette Width (Rousseeuw, 1987), a method for validating clusters' consistency, was used to measure and evaluate the clustering performance (Ramaswamy, no date). In the work, the experiments demonstrated good results of using *k*-medoids for this purpose; however, re-tweets (i.e. duplicates) had influenced the clustering results. Another study used k-means and *k*-medoids respectively to cluster a single Twitter dataset and compare the results of each algorithm (Zhao, 2012). Initially, *k*-means was applied, which took the values in the term-document matrix as numeric, and set the number of clusters, *k*, to eight. After that, the term-document matrix was transformed to a document-term matrix and the clustering was performed. Then, the frequent words in each cluster and the cluster centers were computed in order to discover the meaning of the cluster centroid. The first experiment showed that the clusters were of different topics. The second experiment was conducted using *k*-medoids, which used representative objects instead of means to represent the cluster center. However, the resulting clusters tend to be overlapping and not well separated.

K-means vs. k-medoids. *K*-medoids has the advantage of robustness over *k*-means as it is less influenced by noise and outliers. However, this comes at the cost of efficiency. This is due to the high processing time that is required by *k*-medoids compared to *k*-means. Both methods require the number of clusters, *k*, to be fixed. In terms of clustering sparse data such as tweets, *k*-medoids may not be the best choice as these do not have many words in common and the similarities between them are small and noisy (Aggarwal and Zhai, 2012). Thus, a representative sentence does not often contain the required concepts in order to effectively build a cluster around it.

4.1.2 Fuzzy Clustering

This partition-based method is particularly suitable in the case of no clear groupings in the data set. Unlike hard clustering, fuzzy algorithms assign a continuous value [0, 1] to provide reasonable clustering. Multiple fuzzy clustering algorithms exist in the literature, however fuzzy *c*-means (FCM) (Bezdek et al., 1984) is the most prominent. FCM provides a criteria on grouping data points into different clusters to varying degrees that are specified by a membership grade. It incorporates a membership function that represents the fuzziness of its behavior. The data are bound to each cluster by means of this function.

In the context of Twitter analysis, a recent study presented a simple approach using fuzzy clustering for pre-processing and analysis of hashtags (Zadeh et al., 2015). The resulting fuzzy clusters are used to gain insights related to patterns of hashtags popularity and temporal trends. To analyze hashtags' dynamics, the authors identified groups of hashtags that have similar temporal patterns and looked at their linguistic characteristics. They recognized the most and least representative hashtags of these groups. The adopted methodology is fuzzy clustering based and multiple conclusions were drawn on the resulting clusters with regards to variations of hashtags throughout a period of time. Their clustering was based on the fact that categorization of hashtags is not crisp, rather, most data points belong to several clusters according to certain degrees of membership. Another study compared the performance of supervised learning against unsupervised learning in discriminating the gender of a Twitter user (Vicente et al., 2015). Given only the unstructured information available for each tweet in the user's profile, the aim is to predict the gender of the user. The unsupervised learning involved the usage of soft in conjunction with hard clustering algorithms. *K*-means and FCM were applied on a 242K Twitter users' dataset. The unsupervised approach based on FCM proved to be highly suitable for detecting the user's gender, achieving a performance of about 96%. It also has the privilege of not requiring a labelled training set and the possibility of scaling up to large datasets with improved accuracy.

Fuzzy vs. hard Clustering. Experiments have shown that fuzzy-based clustering is more complex than clustering with crisp boundaries. This is because the former requires more computation time for the involved kernel (Bora et al., 2014). Fuzzy methods provide relatively high clustering accuracy and more realistic probability of belonging. Therefore, they can be considered an effective method that excludes the need of a labelled dataset. This is particularly useful for sheer volumes of tweets, where human annotations can be highly expensive. However, these methods generally have low scalability and results can be sensitive to the initial parameter values. In terms of optimization, fuzzy clustering methods can be easily drawn into local optimal (Khan et al., 2012).

4.2 Hierarchical-Based Clustering

In hierarchical clustering algorithms, data objects are grouped into a tree like (i.e. hierarchy) of clusters. These algorithms can be further classified depending on whether their composition is formed in a top-down (divisive) or bottom-up (agglomerative) manner. This section reviews three studies that performed hierarchical-based clustering algorithms in applications of Twitter mining.

Ifrim et al. (2014) used hierarchical clustering for topic detection in Twitter streams, based on aggressive tweets/terms filtering. The clustering process was performed in two phases, first the tweets and second the resulting headlines from the first clustering step. Their methodology is composed of initially computing tweets pair-wise distances using the cosine metric. Then computing a hierarchical clustering so that tweets belonging to the same topic shall cluster together, and thus each cluster is considered as a detected topic. Afterwards, they controlled the tightness of clusters by cutting the resulting dendrogram at 0.5 distance threshold. In this way they will not have to provide the number of required clusters a-priori as in k-Means. The threshold was set to 0.5 in order to avoid having loose or tight clusters, rather, a value of 0.5 worked well for their method. Each resulting cluster is then assigned the score of

the term with highest weight in the cluster and ranked according to that score. The top 20 clusters are then assigned headlines, which are the first tweet in each of them (with respect to publication time). The final step involved re-clustering the headlines to avoid topic fragmentation, also using hierarchical clustering, the resulting headlines are then ranked by the one with the highest score inside a cluster. The headlines with the earliest publication time are selected and their tweet text is presented as a final topic headline. Another study implemented a hierarchical approach for the purpose of helping users parse tweets results better by grouping them into clusters (Ramaswamy, no date). The aim was for fewer clusters that are tightly packed, rather than too many large clusters. The work involved using a dataset of tweets to see how the choice of the distance function affects the behavior of hierarchical clustering algorithms. Ramaswamy (no date) conducted a survey of two clustering algorithms that are both hierarchical in nature but differ in the implementation of their distance functions. A total of 925 tweets comprising of various topics with common keyword have been used in the experiments. In the first algorithm, the author considered each of the given objects to be in different clusters. Then determining if the object *o* is close enough to cluster *c*, and if so, add *o* to *c*. This process continues until the maximum size of the desired clusters is reached or no more new clusters can be formed. In this first algorithm, the notion of the distance between an object and a cluster has been defined using concepts from association rule problems – support and confidence. The second algorithm maintained the average distance of an object from each element in the cluster as the similarity measure. If the average is small enough, the object is added to the cluster. Both clustering algorithms involve reading the tweets, tokenizing them, clustering them and returning the clustered output. Although the overall behavior was found to be similar for both algorithms, the second one seemed to fare better for each of the confidence and support level value. An integrated hierarchical approach of agglomerative and divisive clustering was proposed to dynamically create broad categories of similar tweets based on the appearance of nouns (Kaur, 2015). In this study, only nouns have been utilized as features as the authors claim they are the most meaningful entities among other part of speech tags, such as verbs, adjectives, and adverbs. Therefore, their approach tends to discard all sentence tokens but nouns. The adopted bottom-up technique merges similar clusters together to reduce their redundancy, in which a recursive and incremental process of dividing and combining clusters has been applied in order to produce more meaningful sorted clusters. The divisive stage works by dividing clusters down the hierarchy to arrange most similar tweets in different clusters. Afterwards, the bottom-up procedure is applied to remove or merge redundant information, if any. This proposed combinatorial approach showed increase in clustering effectiveness and quality compared to standard hierarchical algorithms. However, due to the problem of tweets' sparsity discussed earlier in section 3.1, some tweets might lack the presence of nouns to form a rich nouns foundation in the clustering dataset. Therefore, it might be useful to consider other textual features in addition to nouns to enhance the system's performance.

In this context, empirical evaluations provided that hierarchical methods performed slower than hard partition-based clustering, particularly *k*-means (Kaur and Kaur, 2013). Therefore, for massive social media datasets, hard partitioning methods are considered to be relatively computationally efficient as well as producing acceptable experimental results.

4.3 Hybrid-Based Clustering

The robustness of hierarchical clustering algorithms is relatively high as they tend to compare all pairs of data. However, this makes them not very efficient due to their tendency to require at least $O(n^2)$ computation time. On the other hand, partitioning algorithms may not be the optimal choice despite being more efficient than hierarchical algorithms. This is because the former may not be very effective as they tend to rely on small number of initial cluster representatives. This trade-off has led researchers to propose several clustering algorithms that combined the features of hierarchical and partitioning methods in order to improve their performance and efficiency. These hybrid algorithms include any aggregations between clustering algorithms. In general, they initially partition the input dataset into sub clusters and then construct a new hierarchical cluster based on these sub clusters.

There is not much research conducted using a hybrid clustering approach in the area of Twitter mining. Nevertheless, one approach implemented clustering of keywords that are presented in the tweets using agglomerative hierarchical clustering and crisp *c*-means (Miyamoto et al., 2012). The clustering features was based on a series of tweets as one long sequence of keywords. The approach involved building two datasets, each composed of 50 tweets in different timeframes. Several observations of agglomerative clusters obtained by cutting the dendrogram and *c*-means clusters, with and without pair-wise constrains were analyzed. Better clustering results are provided using pair-wise constrains, however, the size of datasets is relatively small for a generalization.

4.4 Density-Based Clustering

This method groups data located in the region with high density of the data space to belong to the same cluster. Therefore, it is capable of discovering clusters with arbitrary shape. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is the prominent density-based algorithm. It grows regions with sufficiently high density into clusters (Ester et al., 1996). In this section, three articles are summarized in which density-based algorithms has been applied in the exploratory analysis of Twitter.

A density-based clustering has been adopted in the context of Twitter textual data analysis to discover cohesively the information posted by users about an event as well as the user's perception about it (Baralis et al., 2013). The provided framework adopts a multiple-level clustering strategy, which focuses on disjoint dataset portions iteratively and identifies clusters locally. DBSCAN has been exploited for the cluster analysis as it allows discovering arbitrarily shaped clusters, and increases cluster homogeneity by filtering out noise and outliers. Additionally, it does not require prior specification of the number of expected clusters in the data. In this approach, DBSCAN has been applied iteratively on separate dataset portions and identifying clusters locally. All the original dataset is clustered at the first level, then tweets labelled as outliers in the previous level are re-clustered at each subsequent level. To discover representative clusters for their Twitter dataset, they attempt to avoid clusters containing few tweets. They also attempt to limit the number of tweets labelled as outliers and thus unclustered, in order to consider all different posted information. Through addressing these issues, DBSCAN parameters were properly set at each level. A recent study employed DBSCAN as part of its novel method for creating an event detection ground truth through utilizing tweets hashtags (De Boom et al., 2015). The authors clustered co-occurring hashtags using DBSCAN. The method required setting two thresholds: the minimum number of

hashtags per cluster and a minimum similarity measure between two hashtags, above which the two hashtags belong to the same neighborhood. A collection of clusters of sufficiently co-occurring hashtags on the same day were obtained by running DBSCAN for every day in the dataset. A recent study has introduced the application of DBSCAN for representing meaningful segments of tweets in batch mode (Anumol Babu, 2016). The segmentation was done based on calculations of the stickiness score. This score considers the probability of a segment being a phrase within the batch of tweets (i.e. local context) and the probability of it being a phrase in English (i.e. global context) (Li et al., 2015). Sentimental variations in tweets were then analyzed based on these segments. Each word in the text is assigned a sentiment score according to a predetermined sentiment lexicon. The sentiment of a tweet is then denoted as the summation of the most positive score and the most negative score among individual words in the tweet. In this approach, the core of the clustering consisted of integrating DBSCAN with Jaccard Coefficient similarity function. Empirical evaluations indicated an enhancement of the existing system as a result of using DBSCAN for clustering.

It can be observed from the literature surrounding Density-based algorithms in Twitter mining, that they are highly efficient and can be particularly suitable for clustering unstructured data, such as tweets, as it allows the identification of clusters with arbitrary shape. Moreover, it is less prone to outliers and noise, and does not require initial identification of the required number of clusters. However, clustering high data volumes requires a large amount of memory.

4.5 Graph-Based Clustering

These clustering methods are effective in providing results similar to human intuition (Jaromczyk and Toussaint, 1992). Graph-based clustering construct a graph from the set of data and then use the built graph during the clustering process. In these methods, objects are considered as graph vertices and edges are treated in different ways depending on the implemented algorithm (Vathy-Fogarassy and Abonyi, 2013). The graph is a complete graph in its simplest case, and the edges are labelled with the degree of similarity between the objects, which in this case is considered a weighted complete graph. Two articles are reviewed in this section, in which graph-based clustering was utilized in the context of Twitter mining applications.

An approach to graph-based clustering for multi tweet summarization was proposed by Liu et al. (2012), where Twitter-specific features were incorporated to make up for the information shortage in a tweet. In their approach, the number of input varies from hundreds to tens of million tweets. Hundred trending topics were searched and retrieved and a maximum of thousand English tweet was collected in relation to each trending topic. A set of representative tweets were manually selected to from the "gold standard" summarization dataset. This is the optimal data set with human annotations in which the system's output will be evaluated against. It was used for evaluating the proposed graph-based system which showed improvements compared to the LexRank (Erkan and Radev, 2004) baseline. However, these results may not be considered reliable as the manual annotation methodology of the gold standard might be biased.

Dutta et al. (2015), developed a methodology for summarizing tweets based on the graph approach, in which a tweet dataset is taken as input, and a subset of the tweets are derived as the summary of the entire set. This methodology incorporated WordNet (Fellbaum, 1998) to

account for the semantic similarities among tweets which may not use common terms to express the same information. Community detection techniques, which detects the existence of non-trivial network organizations (Yang et al., 2016), are then applied to the constructed graph of tweet similarity in order to cluster similar tweets, and the summary includes a representative tweet from each cluster. In their research, the authors collected 2921 tweets related to the flood in Uttaranchal region of India in 2013, through Twitter API. A set of human generated summaries were obtained for performing evaluations, which were assessed through application of precision (P), recall (R), and F-measure (F).

The main issue in using graph-based algorithms for unsupervised learning on large Twitter datasets, is that computation of the complete weighted graph consumes lots of resources in terms of time and storage. This complexity can be reduced with several methods. This may be through working only with sparse matrices rather than utilizing the complete graph. These matrices contain information about the small subset of the edges corresponding to higher degrees of similarity. Graphs based on these sparse matrices visualize these similarities in a graphical way. The complexity may also be reduced through the application of Vector Quantization (VQ) technique, such as k-means and Neural Gas (NG) (Martinetz and Schulten, 1991), to represent the entire set of objects by a set of representative instances that has a lower cardinality than the one of the original dataset.

5 Discussion

Several approaches of unsupervised learning applications for mining unstructured social media data have been reviewed and presented in table 3. It applies the criteria defined in section 2.3 to perform a systematic comparison of the unsupervised learning applications in Twitter. The featured surveys are discussed in terms of: research approach, clustering method, algorithm, number of clusters, dataset size, distance measure, clustering features, evaluation methods, and results. The review comprises fifteen studies spanning from 2011 to the present. These studies have different approaches, in which the clustering of Twitter data was performed in various settings and domains to achieve different business goals or satisfy certain requirements. The subsequent sections provide a discussion on the studies performing unsupervised learning in Twitter in relation to the general cluster analysis comparison criteria defined in section 2.3. The impact of each criterion on the clustering performance will be further analyzed.

5.1 Problem domain

The unsupervised learning approaches in Twitter range from pure clustering perspectives, such as determining the impact of a distance function choice on a clustering behavior, to a more general pattern recognition application, such as targeting advertisements and event detection. It has been observed that the majority of Twitter-based unsupervised learning applications perform clustering in order to detect news, topics, events, and facts and to predict sentiments. Moreover, there are several different unsupervised machine learning algorithms that can be used to identify patterns. Therefore, understanding the problem domain is key to deriving the right decision on which clustering algorithm is the most appropriate and will ultimately yield valuable analysis.

5.2 Dataset size

Generally, there is no rule-of-thumb about the optimal sample size for cluster analysis. However, the sample size is expected to be correlated with the number of features (i.e. variables) and critically evaluated before the cluster analysis is computed. In 2002, a study that explored unsupervised learning segmentation has reported that the smallest sample size detected contains only 10 elements while the biggest one contains 20,000 (Dolnicar, 2002). In less than ten years, the massive user generated content in OSN, has led to a dramatic increase in the dataset sizes as observed in the reviewed Twitter-based unsupervised approaches. Among these explored studies, which span the period from 2011-present, the average dataset size detected contains 757,255 tweets, ranging from 50 to 10 million. Moreover, the average Twitter user accounts was found to be 126,329, ranging from 10,000 to 242,658 distinct user accounts. Consequently, this massive increase in datasets raises scalability issues in the performance of unsupervised learning in applications of Twitter predictive analysis. However, the majority of the dataset sizes observed in the surveys are considered relatively small with regards to the high volume challenge of Twitter data. Therefore, scalability issues have not been taken into consideration. Effective unsupervised algorithms are expected to scale well to the massive amounts of Twitter data. In this matter, the scalability (in terms of clustering performance) of most of the algorithms implemented in the surveys is questionable as these algorithms have not been tested on considerably large datasets.

In relation to dataset sizes and feature set for unsupervised learning, it has been recommended that the dimensionality is not too high compared to the number of observations to be grouped by the clustering algorithm. Formann (1984) suggests the minimal dataset size should be no less than 2^k objects (k = number of features), preferably $5*2^k$.

5.3 Feature Set

The set of variables are extracted from the raw data to form feature vectors that represent the dataset points. The process of feature selection is critical to the performance of the resulting clusters. Depending on the problem domain, these variables can be numerical, categorical, or a combination of both. In Twitter-based unsupervised applications, textual clustering using the common BOW method raises a problem of high dimensionality feature space and inherent data sparsity. This problem will cause scalability issues and the performance of the clustering algorithm will consequently decline dramatically (Aggarwal and Yu, 2000).

Therefore, it is recommended that the dimensionality of the feature space is reduced through performing one of the following techniques (Liu et al., 2003):

- *Feature extraction* –a process of functional mapping to derive a reduced set of features from the original feature set (e.g. Principal Component Analysis (Wold et al., 1987)). One drawback of these methods is that the reduced feature space may not have a clear meaning, which leads to difficulties in interpreting the clustering results (Dash and Liu, 2000).
- *Feature selection* –a process of choosing a subset features from the original feature set according to some criteria (e.g. Information Gain (Yang and Pedersen, 1997)).

Table 3 shows the set of features that have been used for different problem domains in Twitter-based unsupervised analysis. It has been observed that different sets were used depending on the problem domain. These features include some or all of the following:

- Hashtags –31% of the reviewed surveys included hashtags in the features set and considered their impact, 23% treated hashtags as normal words in the text, and 31% removed hashtags before analysis (excluding the 15% studies that are clustering upon user accounts).
- Account metadata –username, date, status, latitude, longitude, followers, and account followings.
- Tweet metadata –tweet id, published date, and language.
- Maintaining a bag-of-words (BOW) of the unique words contained in each textual data of a tweet and their frequencies as the feature vector. Some included hashtags in the BOW while others ignored them.

Whilst "retweets" and "mentions" conventions in Twitter are claimed to have an impact in boosting tweet popularity (Pramanik et al., 2017), none of the surveys studied the impact of these conventions in assessing the granularities of the unsupervised algorithms in applications of Twitter analysis. Rather, some datasets did not remove the retweeted tweets which affected the resulting clustering credibility. Because tweets commonly get large number of retweets, keeping them in the dataset will produce large clusters containing redundant tweets rather than tweets with similar features. This will consequently reinforce false patterns and increase run time. Therefore, it is imperative that the raw data undergo a complete set of pre-processing to ensure that it is ready for the unsupervised learning process with minimal noise possible.

5.4 Distance Measure

In unsupervised algorithms, the results are strongly influenced by the choice of distance measures. It has been observed from the literature that the choice of the selected distance measure is not often justified for Twitter-based unsupervised applications. Euclidean distance is the default for partitioning algorithms, whereas hierarchical algorithms commonly implemented the cosine similarity measure.

However, it is recommended that the distance measure is chosen based upon a thorough understanding of the problem domain and a critical analysis of the feature set. In general, if the magnitude of the feature vector does not matter, cosine is used because it measures the angle between two vectors rather than their distance in the feature space. Thus, it is a measure of orientation and not magnitude. For example, consider a text with the word "sea" appearing 8 times and another text with the word "sea" appearing 3 times, the Euclidean distance between their feature vectors will be higher but the angle will still be small. This is due to the two vectors pointing to the same direction, which is what matters when performing unsupervised learning in the context of Twitter (e.g. clustering tweets). Therefore, it is ultimately important to choose the right distance function for the unsupervised problem under consideration. Table 3 compares applications of different distance measures used in relation to the problem domain.

5.5 Clustering Algorithm

It has been observed from the literature surrounding unsupervised Twitter analysis that partition-based algorithms are used when the problem domain implies knowledge on the granularities present in the dataset. That is, the number of required clusters to be generated is known a priori. Hierarchical algorithms are generally used for topic detection applications where there

is lack of knowledge on the themes in the dataset. Density-based methods are used in event detection applications where hashtag features are utilized to identify dense areas in the feature space, which are considered as events (i.e. clusters of arbitrary shapes). Furthermore, it has been observed that graph-based clustering is used for tweets summarization, in which the algorithm only requires pre-specifying the threshold of similarity between pairs in the dataset.

5.6 Number of Clusters

As partitioning algorithms require the number of clusters, c, to be pre-specified, c has been included in this study to provide a generalized indication on the number of clusters that might be appropriate for similar tasks. From the featured surveys, the average number of clusters maintained is 7, with 2 as the minimum clusters and 10 as the maximum. Generally, the number of clusters depends on the target application as large c indicates, optimally, fine grained granularities (i.e. more similarity between data points), whereas small c indicates coarse grained granularities (i.e. more towards topic modelling than pairs semantic similarity).

However, when the number of clusters is unknown, a common practice is to perform an iterative method in order to find the most pure segmentation that provides the minimum intra-cluster variance and maximum inter-cluster variance.

5.7 Evaluation Method

Evaluation methods vary from robust measures, such as ASW to manual observations, such as manually comparing an algorithm's detected topics with Google news headlines. It can be observed that objective evaluation of clusters quality such as ASW has been utilized by most of the studies in Twitter to measure the clustering performance. Some of the evaluation methods are derived from other data mining techniques such as association rules and classification. These methods include clustering based on confidence and support levels, and calculating precision, recall and the F measure from a confusion matrix.

In unsupervised text clustering applications, it is generally recommended to incorporate subjective evaluation of clusters as these will reveal the semantic relations between the centroids and the data points in the same clusters and their degree of belongingness. Theoretically, subjective evaluation methods may involve a researcher to acquire an intuition for the results evaluation. However, in practice, the massive amounts of social data and the specific details and variety of vocabulary used in these textual data representations make the intuitive judgment difficult for application over the whole dataset. The existence of a benchmark dataset, which is ideally produced by human judges with a good level of inter-judge agreement can be used as a surrogate for user judgments. However, this is not always available and can be expensive to generate.

6 Conclusion

The contribution of this review to the literature is demonstrated in several significant ways:

1. It presents a detailed explanation on the different forms of textual challenges presented in the unstructured data of Twitter. In addition, for each of these challenges, provides different implemented approaches in the literature for alleviating them and discusses their effectiveness. This is extremely important for research, not only in unsupervised

learning, but also for other data mining and NLP research that require textual data pre-processing in the context of Twitter analysis.

2. The review established a general comparison criteria for unsupervised leaning in Twitter, which defines each criterion in a cluster analysis problem and associated dependencies. This criteria has been used to conduct a systematic comparative analysis on applications that utilized and tuned unsupervised approaches to the characteristics of Twitter unstructured data.
3. It concentrated on algorithms of the general unsupervised methods: (1) partition-based, (2) hierarchical-based, (3) hybrid-based, (4) density-based, and (5) graph-based, in Twitter mining, and discuss them in the context of Twitter analysis.
4. Unlike existing reviews which provides high level and abstract specification of surveys, this review was comprehensive in that it provided comparative information and discussion across the dataset size, approach, clustering methods, algorithm, number of clusters, distance measure, clustering feature, evaluation methods, and results.

Fifteen articles were reviewed in this paper, and the results indicated that there is a sufficient improvement in the exploratory analysis of social media data. However, many of the existing methodologies have limited capabilities in their performance and thus limited potential abilities in recognizing patterns in the data:

- Most of the dataset sizes are relatively small which is not indicative of the patterns in social behaviors and therefore generalized conclusions cannot be drawn. Because of the sparsity of Twitter textual data, it is difficult to discover representative information in small datasets. Therefore, future studies should aim to increase the size of the dataset.
- Some of the algorithms implemented may have provided effective results in terms of efficiency and accuracy. However, this may be attributed to the small size of dataset as the scalability has not been evaluated.
- Some of the reviewed datasets included redundant tweets (i.e. retweets) which yields inaccurate clustering. Therefore, future studies should perform a comprehensive pre-processing phase in which retweets and other noise, such as URLs, are removed from the dataset prior to clustering.
- Most of the studies implemented keyword-based techniques, such as term frequencies and BOW which ignores the respective order of appearance of the words and does not account for correlations between text segments. Therefore, future research should incorporate and measure the underlying semantic similarities in the dataset.
- In terms of clustering evaluation, objective techniques that measure the granularity compactness, such as ASW, have been applied. However, it is imperative to incorporate subjective procedure to the evaluation process to ensure the evaluation of the semantic belongingness and similarities among clusters' data points.

With reference to the comparison criteria discussed in section 2.3, general conclusions and recommendations can be made on the state-of-the art unsupervised learning in Twitter:

- *(C1)* –the massive user generated content in microblogs (e.g. Twitter) provide potential value for different applications. The use of unsupervised algorithms for Twitter can reveal hidden patterns due to several reasons as discussed in section 1.
- *(C2)* –the dataset sizes has dramatically increased since 2002 due to huge data volume in Twitter. Hence, for an unsupervised learning algorithm to provide high performance predictions, it requires large datasets. However, this raises scalability issues.

- *(C3)* –depends on the problem domain. Dimensionality reduction methods can be applied carefully when the feature space it too big in order to enhance the performance of the unsupervised learning algorithm.
- *(C4)* –depends on the target application and the representation of features. Empirical experiments can be performed to find the best performing measure for the problem under consideration.
- *(C5)* –the choice of the algorithm is influenced by the dataset size as some algorithms are more efficient in dealing with the massive Twitter data.
- *(C6)* –the experimentation of different clusters to find the best segmentation of the dataset is recommended. However, this does not always translate into good effectiveness in an application and therefore an efficient evaluation criteria is required.
- *(C7)* –where possible, integrating objective and subjective methods yield the best evaluation method

In conclusion, it can be clearly established that unsupervised learning is an important element of exploratory text analysis in Twitter, where unstructured data can be useful in pattern recognition as well as identification of user potentials and interests. However, future research must demonstrate the effectiveness of such approaches through acquiring larger datasets in order for the algorithms to be useful in discovering knowledge and applicable in several contexts and domains. A meta-analysis review is recommended as a future work, which will provide a quantitative estimate for the impact and usefulness of unsupervised learning methods in providing insights for different Twitter-based applications.

Table 3: Summary of the studies featured in this review

Author	Title	Problem Domain	Clustering Method	Algorithm and Number of Clusters (C)	Dataset Size	Distance Measure	Feature Set	Evaluation Methods	Results
Friedemann (Friedemann, 2015)	Clustering a Customer Base Using Twitter Data	Targeting advertisements	Partitioning-Based Clustering – Hard Partitioning	*k*-Means C: 5	10,000 Twitter user accounts	Euclidean distance	posted status, number of followers and account followings, latitude, longitude, whether a popular Twitter account (*influencer*) is followed	Computing a metric of clustering quality *q*. The lower the value of *q*, the better clustering performance	Achieved clustering is midway between ideal and randomized data. Experiments emphasized the credibility of Twitter data for market analysis
Soni & Mathi (Soni and Mathai, 2015)	Improved Twitter Sentiment Prediction through Cluster-then-Predict Model	Sentiment prediction	Partitioning-Based Clustering – Hard Partitioning	*k*-Means C: 2	1200 "Apple" tweets	Squared Euclidean distance	Bag-of-Words (BOW) from twitter corpus (frequency of word occurrences)	Confusion matrix and ROC (Receiver Operator Characteristic) graph	Model integration of supervised and unsupervised *k*-Means learning improved twitter sentiment prediction
Purwitasari et al. (Purwitasari et al., 2015)	*K*-medoids Algorithm on Indonesian Twitter feeds for Clustering Trending Issues as Important Terms in News Summarisation	News summary	Partitioning-Based Clustering – Hard Partitioning	*k*-Medoids C: 10	200 tweets (geolocation: Indonesia)	Cosine similarity	Term frequencies and weight in tweet text. Hashtags omitted	The larger ASW value, the more homogeneous the cluster result	Inclusion of retweets affected cluster result quality
Zhao (Zhao, 2012)	R and Data Mining: Examples and Case Studies	R Data Mining	Partitioning-Based Clustering – Hard Partitioning	*k*-Means C: 8	1st 200 tweets from @rdatamining account	Euclidean distance	Term frequencies in tweet text (document-term matrix). Hashtags omitted	Checked the top 3 terms in every cluster	Clusters of different topics
				k-Medoids C: 9		Manhattan distance		ASW	Clusters overlap and not well separated
Vicente et al. (Vicente et al., 2015)	Twitter Gender Classification using User Unstructured Information	Gender detection	Partitioning-Based Clustering – Hard Partitioning	*k*-Means C: 2	242,658 unique Twitter users	Euclidean distance	Screen name and user name	Two experiments: 1st used labelled data for building clusters and evaluating performance. 2nd used unlabelled data for clustering and labelled for evaluation	*C*-Means provided better clustering performance than *k*-Means. More usage of unlabelled data significantly enhanced *c*-means but got *k*-Means worse
			Partitioning-Based Clustering – Fuzzy Partitioning	*c*-Means C: 2					
Zadeh et al. (Zadeh et al., 2015)	Analysis of Twitter Hashtags: Fuzzy Clustering Approach	Events and facts detection	Partitioning-Based Clustering – Fuzzy Partitioning	FANNY (Kaufman and Rousseeuw, 2009) C: 6	40 distinct hashtags	Manhattan Distance	Temporal aspects of hashtags	Defined a *misfit* measure to identify elements' degree of "not fitting" into a cluster. Clustering performance measured using ASW	Insights into patterns associated with each cluster for hashtags changing popularities over time

Table 3: Summary of the studies featured in this review (cont.)

Author	Title	Problem Domain	Clustering Method	Algorithm and Number of Clusters (C)	Dataset Size	Distance Measure	Feature Set	Evaluation Methods	Results
Ifrim et al. (Ifrim et al., 2014)	Event detection in Twitter using aggresive filtering and hierarchical tweet clustering	Topic detection	Hierarchical-Based Clustering	Agglomerative (dendrogram cut at 0.5)	1st: 1,084,200 2nd: 943,175 JSON format English tweets	Cosine Similarity	Date, tweet id, text, mentions, hashtags, URLs, and retweeted or not	(1) A subset of ground truth topics. (2) Manual assessment of how many detected topics are actually published news in traditional media	This clustering application can detect topics with 80% accuracy. However, not efficient for real-time data analysis
Ramaswamy (Ramaswamy)	Comparing the efficiency of two clustering techniques	Impact of distance function choice on clustering behaviour	Hierarchical-Based Clustering	Two Ward (Jr., 1963) algorithms C: 5	925 tweets	Ratio of tweets appearing in different clusters / Avg. distance between tokens and clusters	Tweets text tokenization	Experiments conducted to determine appropriate values of confidence and support levels that determine further clustering	Similar behaviour for both algorithms. In terms of fewer, tightly packed clusters, 2nd algorithm fared better for confidence and support values
Kaur (Kaur, 2015)	A combinatorial tweet clustering methodology utilizing inter and intra cosine similarity	Noun-based tweet categorization	Hierarchical-Based Clustering	Agglomerative / Divisive	"stem cell" 15062 tweets	Inter-cosine similarity / Intra-cosine similarity	Frequency of occurrences for nouns in tweets. Hashtags omitted	Experimental comparisons of clustering quality against: *k*-means, Ward, and DBSCAN clustering.	Higher accuracy compared to existing methodologies, however, at the cost of performance. Clustering runtime: 1hour
Baralis et al. (Baralis et al., 2013)	Analysis of twitter data using a multiple-level clustering strategy	Cohesive information discovery	Density-Based Clustering	DBSCAN	"Paralympics" 1969 tweets / "Concert" 2960 tweets	Cosine similarity	BOW of tweets including hashtags	ASW	Effective in discovering knowledge. Performance relatively low and may not scale well to massive datasets. Clustering runtime: 2min 9sec
De Boom et al. (De Boom et al., 2015)	Semantics-driven event clustering in twitter feeds	Event detection	Density-Based Clustering	DBSCAN	63,067 tweets (geolocation: Belgium)	∑ (avg. occurrences of both hashtags per day)/2	Hashtags co-occurrence matrix	Precision, recall, and F measures	Improvement in event detection and clustering through high-level semantic information
Anumol Babu (Anumol Babu, 2016)	Efficient density based clustering of tweets and sentimental analysis based on segmentation	Sentiment Anaslysis	Density-Based Clustering	DBSCAN	100 synthetic tweets	Jaccard Similarity	Tweet text and publication time. Hashtags omitted	Evaluating tweets segmentation and its accuracy through an experiment	Enhancement of the present system as DBSCAN was integrated

Table 3: Summary of the studies featured in this review (cont.)

Author	Title	Problem Domain	Clustering Method	Algorithm and Number of Clusters (C)	Dataset Size	Distance Measure	Feature Set	Evaluation Methods	Results
Liu et al. (Liu et al., 2012)	Graph-based multi-tweet summarization using social signals	Representative tweets extraction	Graph-Based Clustering	Adapting LexRank C: 0.05*dataset size	10 million tweets	Cosine Similarity	Social network features, readability, and user diversity	N-gram recall based statistic (ROUGE-N) on manually annotated dataset.	The use of augmented features improved the performance of ROUGE-1 and ROUGE-2. Manual annotation may be biased
Dutta et al. (Dutta et al., 2015)	A graph based clustering technique for tweet summarization	Tweet summarization	Graph-Based Clustering	Weighted graph based on community detection techniques	2921 tweets	A combination of term-level and semantic similarities	Term-level features Semantic features	Precision, recall, and F-measure on human generated summaries	Approach achieves better summarization performance over an existing summarization technique SumBasic (Vanderwende et al., 2007)
Miyamoto et al. (Miyamoto et al., 2012)	Clustering in tweets using a fuzzy neighborhood model	Keyword clustering	Hybrid-Based Clustering	Hard *c*-Means (partitioning) C: 2 / Agglomerative (hierarchical) C: 2	1st: 50 tweets (35 terms occur > 8 times) 2nd: 50 tweets (38 terms occur > 5 times)	Squared Euclidean distance	Sequence of word occurrences in a set of tweets	Several observations of: clusters with and without pair-wise constraints clusters obtained by cutting the dendrogram with and without pair-wise constraints	Application of pair-wise constraints improved clustering quality. However, dataset size is arguably small

References

ABBASI, M.-A. & LIU, H. Measuring user credibility in social media. International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction, 2013. Springer, 441-448.

AGGARWAL, C. C. & REDDY, C. K. 2013. *Data clustering: algorithms and applications*, CRC press.

AGGARWAL, C. C. & YU, P. S. 2000. *Finding generalized projected clusters in high dimensional spaces*, ACM.

AGGARWAL, C. C. & ZHAI, C. 2012. *Mining text data*, Springer Science & Business Media.

ALNAJRAN, N., CROCKETT, K., MCLEAN, D. & LATHAM, A. Cluster Analysis of Twitter Data: A Review of Algorithms. Proceedings of the 9th International Conference on Agents and Artificial Intelligence, 2017. Science and Technology Publications (SCITEPRESS)/Springer Books, 239-249.

ANUMOL BABU, R. V. P. 2016. Efficient Density Based Clustering of Tweets and Sentimental Analysis Based on Segmentation. *International Journal of Computer Techniques,* 3, 53-57.

ARORA, P. & VARSHNEY, S. 2016. Analysis of K-Means and K-Medoids algorithm for big data. *Procedia Computer Science,* 78, 507-512.

BARALIS, E., CERQUITELLI, T., CHIUSANO, S., GRIMAUDO, L. & XIAO, X. Analysis of twitter data using a multiple-level clustering strategy. International Conference on Model and Data Engineering, 2013. Springer, 13-24.

BERKHIN, P. 2006. A survey of clustering data mining techniques. *Grouping multidimensional data,* 25, 71.

BEZDEK, J. C., EHRLICH, R. & FULL, W. 1984. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences,* 10, 191-203.

BORA, D. J., GUPTA, D. & KUMAR, A. 2014. A comparative study between fuzzy clustering algorithm and hard clustering algorithm. *arXiv preprint arXiv:1404.6059*.

BREIMAN, L. 2001. Random forests. *Machine learning,* 45, 5-32.

CASTILLO, C., MENDOZA, M. & POBLETE, B. Information credibility on twitter. Proceedings of the 20th international conference on World wide web, 2011. ACM, 675-684.

DASH, M. & LIU, H. Feature selection for clustering. Pacific-Asia Conference on knowledge discovery and data mining, 2000. Springer, 110-121.

DE BOOM, C., VAN CANNEYT, S. & DHOEDT, B. Semantics-driven event clustering in twitter feeds. Making Sense of Microposts, 2015. CEUR, 2-9.

DOLNICAR, S. 2002. A review of unquestioned standards in using cluster analysis for data-driven market segmentation.

DUTTA, S., GHATAK, S., ROY, M., GHOSH, S. & DAS, A. K. A graph based clustering technique for tweet summarization. Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2015 4th International Conference on, 2015. IEEE, 1-6.

ERKAN, G. & RADEV, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research,* 22, 457-479.

ESTER, M., KRIEGEL, H.-P., SANDER, J. & XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd, 1996. 226-231.

FELLBAUM, C. 1998. *WordNet*, Wiley Online Library.

FORMANN, A. K. 1984. *Die latent-class-analyse: Einführung in Theorie und Anwendung*, Beltz.

FRIEDEMANN, V. 2015. Clustering a Customer Base Using Twitter Data.

GO, A., BHAYANI, R. & HUANG, L. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford,* 1.

GODFREY, D., JOHNS, C., MEYER, C., RACE, S. & SADEK, C. 2014. A case study in text mining: Interpreting twitter data from world cup tweets. *arXiv preprint arXiv:1408.5427*.

HAN, J., PEI, J. & KAMBER, M. 2011. *Data mining: concepts and techniques*, Elsevier.

IFRIM, G., SHI, B. & BRIGADIR, I. Event detection in twitter using aggressive filtering and hierarchical tweet clustering. Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014, 2014. ACM.

ITO, J., SONG, J., TODA, H., KOIKE, Y. & OYAMA, S. Assessment of tweet credibility with LDA features. Proceedings of the 24th International Conference on World Wide Web, 2015. ACM, 953-958.

JAIN, A. K., MURTY, M. N. & FLYNN, P. J. 1999. Data clustering: a review. *ACM computing surveys (CSUR),* 31, 264-323.

JAROMCZYK, J. W. & TOUSSAINT, G. T. 1992. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE,* 80, 1502-1517.

KAUFMAN, L. & ROUSSEEUW, P. J. 2009. *Finding groups in data: an introduction to cluster analysis*, John Wiley & Sons.

KAUR, M. & KAUR, U. 2013. Comparison between k-means and hierarchical algorithm using query redirection. *International Journal of Advanced Research in Computer Science and Software Engineering,* 3.

KAUR, N. 2015. *A Combinatorial Tweet Clustering Methodology Utilizing Inter and Intra Cosine Similarity*. Faculty of Graduate Studies and Research, University of Regina.

KHAN, K., SAHAI, A. & CAMPUS, A. 2012. A fuzzy c-means bi-sonar-based metaheuristic optomization algorithm. *IJIMAI,* 1, 26-32.

KRESTEL, R., WERKMEISTER, T., WIRADARMA, T. P. & KASNECI, G. Tweet-Recommender: Finding Relevant Tweets for News Articles. Proceedings of the 24th International Conference on World Wide Web, 2015. ACM, 53-54.

KUMAR, S., MORSTATTER, F. & LIU, H. 2013. *Twitter data analytics*, Springer Science & Business Media.

LI, C., SUN, A., WENG, J. & HE, Q. 2015. Tweet segmentation and its application to named entity recognition. *IEEE Transactions on Knowledge and Data Engineering,* 27, 558-570.

LIU, T., LIU, S., CHEN, Z. & MA, W.-Y. An evaluation on feature selection for text clustering. Proceedings of the 20th International Conference on Machine Learning (ICML-03), 2003. 488-495.

LIU, X., LI, Y., WEI, F. & ZHOU, M. Graph-Based Multi-Tweet Summarization using Social Signals. COLING, 2012. 1699-1714.

MAITY, S., CHAUDHARY, A., KUMAR, S., MUKHERJEE, A., SARDA, C., PATIL, A. & MONDAL, A. WASSUP? LOL: Characterizing Out-of-Vocabulary Words in Twitter. Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion, 2016. ACM, 341-344.

MARTINETZ, T. & SCHULTEN, K. 1991. A" neural-gas" network learns topologies.

MIYAMOTO, S., SUZUKI, S. & TAKUMI, S. Clustering in tweets using a fuzzy neighborhood model. Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on, 2012. IEEE, 1-6.

PALGUNA, D. S., JOSHI, V., CHAKARAVARTHY, V., KOTHARI, R. & SUBRAMANIAM, L. V. Analysis of sampling algorithms for twitter. Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

PRAMANIK, S., WANG, Q., DANISCH, M., GUILLAUME, J.-L. & MITRA, B. 2017. Modeling cascade formation in Twitter amidst mentions and retweets. *Social Network Analysis and Mining,* 7, 41.

PURWITASARI, D., FATICHAH, C., ARIESHANTI, I. & HAYATIN, N. K-medoids algorithm on Indonesian Twitter feeds for clustering trending issue as important terms in news summarization. Information & Communication Technology and Systems (ICTS), 2015 International Conference on, 2015. IEEE, 95-98.

RAMASWAMY, S. no date. Comparing the Efficiency of Two Clustering Techniques.

ROUSSEEUW, P. J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics,* 20, 53-65.

SAIF, H., HE, Y. & ALANI, H. Alleviating data sparsity for twitter sentiment analysis. 2012. CEUR Workshop Proceedings (CEUR-WS. org).

SHEELA, L. J. 2016. A Review of Sentiment Analysis in Twitter Data Using Hadoop. *International Journal of Database Theory and Application,* 9, 77-86.

SINNOTT, R. O. & WANG, W. 2017. Estimating micro-populations through social media analytics. *Social Network Analysis and Mining,* 7, 13.

SONI, R. & MATHAI, K. J. 2015. Improved Twitter Sentiment Prediction through Cluster-then-Predict Model. *arXiv preprint arXiv:1509.02437*.

VATHY-FOGARASSY, Á. & ABONYI, J. 2013. *Graph-based clustering and data visualization algorithms*, Springer.

VICENTE, M., BATISTA, F. & CARVALHO, J. P. Twitter gender classification using user unstructured information. Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on, 2015. IEEE, 1-7.

WOLD, S., ESBENSEN, K. & GELADI, P. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems,* 2**,** 37-52.

XU, R. & WUNSCH, D. 2005. Survey of clustering algorithms. *IEEE Transactions on neural networks,* 16**,** 645-678.

YANG, J. & LESKOVEC, J. Patterns of temporal variation in online media. Proceedings of the fourth ACM international conference on Web search and data mining, 2011. ACM, 177-186.

YANG, Y. & PEDERSEN, J. O. A comparative study on feature selection in text categorization. Icml, 1997. 412-420.

YANG, Z., ALGESHEIMER, R. & TESSONE, C. J. 2016. A Comparative Analysis of Community Detection Algorithms on Artificial Networks. *Scientific Reports,* 6.

ZADEH, L. A., ABBASOV, A. M. & SHAHBAZOVA, S. N. Analysis of Twitter hashtags: Fuzzy clustering approach. Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC), 2015 Annual Conference of the North American, 2015. IEEE, 1-6.

ZHAO, Y. 2011. R and Data Mining: Examples and Case Studies.

ZHAO, Y. 2012. *R and data mining: Examples and case studies*, Academic Press.

TREASURE: Tweet Similarity Measure Based on Semantic and Syntactic Computation

Noufa Alnajran, Keeley Crockett, David McLean and Annabel Latham

School of Computing, Math and Digital Technology, Manchester Metropolitan University
John Dalton Building, All Saints, Manchester, M1 5GD, U.K.
Noufa.alnajran@stu.mmu.ac.uk, (k.crockett, d.mclean, a.latham)@mmu.ac.uk

Abstract

This paper presents a novel approach for measuring the semantic similarity between English tweets. Previous work on semantic similarity methods, designed for microblogging posts, particularly tweets has focused on either external resources such as WordNet (e.g., path length and depth of the common subsumer), or on statistical semantics such as Latent Semantic Analysis (LSA). The reliance on semantic nets approaches fall short when applied to microblogging posts, such as tweets, where a high proportion of out-of-vocabulary (OOV) terms occur within a very short context. As such words are not present in WordNet, the post (e.g. tweet) representation vector becomes very sparse. Therefore, this research proposes a semantic similarity measure, namely TREASURE, to incorporate word embedding in the construction of the semantic feature vector, and utilizing information content (IC) to weight each term in a tweet. Through conventional corpus-based IC, each term is annotated with a weight that is computed from the distributions of concepts over a textual corpus, which is constructed from a large set of domain–related microblog posts. Through experiments performed on SemEval-2014 shared task dataset, results show that the TREASURE demonstrates a statistically significant improvement over state-of-the-art semantic similarity methods. Moreover, it has demonstrated the highest performance in terms of correlation on a political tweets dataset with reference to expert judgements with good inter-judge agreement.

Keywords

Text Similarity, Twitter Analysis, Word Embedding, Semantic Nets, Natural Language Processing, Semantic Similarity, Social Network Analysis, Twitter Analysis

1. Introduction

Twitter is gaining rapid prominence as a source of information sharing due to its massive user generated content. Consequently, Twitter has become a goldmine of potential insights and knowledge discovery. Recent applications of microblogging, particularly Twitter, present a need for an effective approach to compute the semantic similarity between tweets. Examples of such applications are political engineering [1], trend analysis, truth discovery, search ranking [2], paraphrase identification [3], and named entity disambiguation [4]. These applications are achieved through performing cluster analysis of tweets to generate a more concise and organized representation of the raw tweets, in which a tweet similarity measure is essential to the implementation. The employment of an effective tweet similarity measure in the clustering algorithm instead of conventional distance measures (e.g. Euclidean distance) can significantly enhance the accuracy of the resulting clusters, as it better captures the features in the text [5]. Measuring tweets similarities will have user-related applications as well. In detecting human behavior, tweets similarity can reveal hidden patterns on different human cognition and attitudes. For example, measuring similarities between users posts may indicate common behavior or similar views on a certain controversial topic (e.g. argumentation mining [6]). In machine learning, tweet similarity is used to classify tweets into pre-determined categories [7]. In business-based applications, the integration of sentiment features into the similarity measure can achieving competitive advantage [8]. Moreover, the incorporation of tweet similarity is beneficial to applications such as bilingual tweet translation evaluation [9], where the quality of the system translation output is assessed by measuring the degree of equivalence between a human translation and the machine output. These exemplar applications show that tweet similarity plays a significant role in computational linguistics and has become a generic component for the research community involved in OSN-related knowledge analysis and representation.

However, Twitter microblog text is characterized by short messages, common conventions (e.g. hashtags, retweets, mentions, etc.), threaded conversations, and inclusion of emoticons and URLs. It often provides colloquial content that contains erroneous words and acronyms. In addition to the genre's informal nature, character restriction (maximum of 140 characters per tweet) encourage "compressed" utterances, in which users omitting not only useless words but also those with grammatical or contextualizing function. This linguistic noise makes measuring tweet similarity very challenging.

Previous work on detecting short-text similarity have centered on the traditional approach of analyzing potential types of relations in ontologies such as WordNet [10]. These approaches consider hierarchical (e.g. is-a), associative (e.g. cause-effect), and equivalence (synonymy) relations of concepts. Such methods are usually effective when dealing with text of proper English in which most of the terms used are present in the lexical hierarchy [11]. However, in Twitter, most of the text used will not be present in semantic nets. This is mainly due to the 140 character limit, which imposes lots of shortened lingo of abbreviations and acronyms. Even after the official increase in this limit to 280 characters, which was applied in November 2017 by Jack Dorsey, CEO of Twitter, the company announced that only 5 per cent of the posted tweets during the two-month trial period has exceeded 140 characters. Furthermore, Aliza Rosen, Twitter product manager, mentioned that people are tweeting below 140 most of the time and the brevity of Twitter remained. Rosen acknowledged that people have been tweeting for years and thus, might have an "emotional attachment" to 140 characters. Therefore, the brevity and informality of such microblogging platform poses serious computational linguistic challenges. The focus of this paper is on predicting the semantic similarity between microblogging posts, particularly tweets. TREASURE, the proposed similarity measure, integrates semantic as well as syntactic features in a tweet pair to produce a similarity score.

Although tweet similarity is essential for a variety of applications, as described earlier in this paper, there is not much research on computing semantic similarity for tweets. Moreover, the use of existing measures to computing tweet similarity has three major drawbacks. First, sentence similarity measures configured on WordNet will perform poorly on a Twitter-based dataset as most terms are not present in the ontological hierarchy. Second, corpus-based semantic measures that are trained and designed for an application domain cannot be adapted easily to other domains. Third, some approaches require user's intensive involvement to manually preprocess the noisy text in tweets, which is immensely arduous and tedious task. This lack of adaptability corresponds to the informal nature of communication platform and common user generated conventions used in most OSN. To address these drawbacks, this paper aims to develop a hybrid approach to similarity measurement of microblogging posts that: 1) Undertake a pre-processing methodology that aims to model a tweet by extracting semantic and syntactic features. 2) Implements a new short-text semantic similarity (STSS) measure, known as TREASURE (Tweet similaRity mEASURE), for tweets. Based on acritical review of previous studies and state-of-the-art approaches and their associated weaknesses in handling microblogs computational linguistic challenges, an effective measure is considered to have the following characteristics:

- *Symmetric* –the similarity degree between two candidate tweets, T_1 and T_2, should be the same as that between T_2 and T_1.
- *Fully unsupervised* –does not require any kind of user manual intervention.
- *Hybrid feature set* –extracts and utilizes both semantic and syntactic features present in a tweet pair.
- *Dynamic pipeline* –creates a dynamic joint vector representing the tweet pair rather than a static high dimensional bag-of-words (BOW).
- *Adaptable* –readily replicated across the range of potential application domains in the context of microblogging OSN.

The next section briefly reviews key related work. In section 3, the new STSS approach for measuring tweet similarity is presented. This section describes the different modules in each component. Section 4 provides implementation considerations related to datasets and benchmark production and sampling

methodology. Section 5 shows the similarities calculated for a set of benchmarked pairs of tweets dataset with human similarity labels. In this section, experiments are carried out to evaluate our similarity method and compare its performance to state-of-art approaches. Section 5 concludes that the proposed method strongly correlate with human cognition perceptions on tweets similarities. Finally, section 6 summarizes the work, draws some conclusions, and proposes future related works.

2. Related Work

In general, there is much literature on measuring the similarity between sentences or short texts [11-13], but there are very few published work relating to the measurement of similarity between tweets. This section reviews some related work in order to explore the strengths and limitations of previous methods, and to identify the particular difficulties in computing tweet similarity. Related works on text similarity can be classified into four major categories: keyword-based methods, knowledge-based methods, corpus based methods, and hybrid-based methods.

The keyword-based methods are often known as the Bag-of-Words (BOW) representation, which is commonly used in Natural Language Processing (NLP) Information Retrieval (IR) applications [14]. This model represents text as an unordered list of the words it is composed of. It does not consider grammatical structure or word order. In case of IR systems, a query is considered as a document, and the relevant documents to be retrieved are the ones that share similar keywords vector with the query vector. This method relies of the assumption that the similarity between documents increases as the common words between them increase. If this technique was applied to tweet similarity, it would have three obvious limitations:

1. Each tweet is represented by a feature vector of a precompiled Twitter-based word list with n words, in which n is generally in the millions in order to include all unique keywords (i.e. features) in the dataset under consideration. Hence, the resulting vectors are very sparse as they would have many null components.
2. Most of the works in Twitter use a BOW model that ignores the discourse particles and stop words such as but, as, since, of, etc. However, these words cannot be ignored in tweet similarity computation as they carry structural information, which contributes to the interpretation of tweet semantics. The inclusion of such words will increase the vector dimensionality even greater.
3. Tweets that are similar in meaning do not necessarily share common words and sharing many words does not imply similarity. Thus, the precompiled static list of words doesn't reflect the correct semantic information in the context of compared tweets.

An enhancement to the keyword-based approach is the use of semantic dictionary information to augment the keywords vector with semantic features to compute the similarity of a pair of words taken from the two tweets that are under comparison. Similarity values of all word pairs are then aggregated to compute the overall tweet similarity [15]. Pawar and Mago [11] proposed a sentence similarity method using edge-counting based technique between joint words from the two compared sentences after removing stop words. Their method scales down the overall similarity by calculating the word order. Li *et al.* [12] (STASIS) and Tian *et al.* [16] proposed a knowledge-based approach that rely on structural knowledge in taxonomy (e.g. path length, depth, and common subsumer) to compute sentence similarity through calculating the IC of each word from WordNet. IC is computed using a formula that considers the set of synonyms (i.e. synsets) in WordNet and a constant that represents the total number of concepts in WordNet. While these approaches show high correlation when applied to an annotated English pairs, they will fall short when adapted to compute the similarity between tweets. This is due to the common Twitter-based features that contribute to the overall tweet similarity (e.g. hashtags, mentions, emoticons, etc.), which are not taken into consideration. In [17], a method is proposed for calculating Bengali tweet similarity based on computing word similarity using Bengali WordNet [18]. Calculating semantic similarity among two words, the authors compute a scalar distance of the given words in the meaning spaces based on the words synsets extracted from WordNet. Both words are considered synonyms if they belong to the same synset, otherwise, the distance between

them is 1 and similarity score is 0. An $n \times m$ matrix is generated, where n and m are the lengths of the compared tweets and each sell represents the word level similarity score. The overall tweet similarity is obtained by dividing the sum of synonym words by the sum of n and m. This method is pretty much similar to BOW as it presents a naïve approach to semantic similarity, which is due to the lack of consideration to the hierarchical relations such as path length or depth for words that are not the same synset. Rather, it assigns a distance of 1 between them (i.e. 0 similarity). Nevertheless, despite the authors' claim that Bengali tweets are less noisy in nature compared to English tweets, this method is still weak in capturing the underlying similarities in tweets.

Knowledge-based approaches fall short when applied within Twitter similarity applications due to three main reasons:

1. Due to its informal nature, Twitter contains lots of improper words (i.e. misspellings, jargons, acronyms, slangs, etc.) that people come up with rapidly. These words are usually not present in semantic nets as these are generally human crafted dictionaries that do not capture all possible words. Therefore, much of the similarity between tweets will be missing because of the lack of word presence in the semantic hierarchy.
2. The most widely used knowledge base, WordNet, is limited in the number of verbs and adverbs synsets compared to the available nouns synsets. Hence, referring to the first reason, WordNet is considered a limited resource to be used for tweets similarity computation for the English language.
3. Semantic nets model polysemy and synonymy relations between concepts (unigrams). Therefore, relations between bigrams such as 'computer science' (or trigrams) are not represented.

A well established and active field of research that contributes to semantic similarity computation is related to methods based on corpus statistical information of words. Corpus-based methods are generally categorized into: 1) word weighting methods and, 2) word co-occurrence methods.

Corpus weighting methods such as Term Frequency-Inverse Document Frequency (TF-IDF) [19] assumes that documents have common words [20, 21]. This method is generally used in IR systems, in which each word is normalized by the frequency of its occurrence over all documents. It aims to favor documents' discriminatory traits over nondiscriminatory ones such as 'Trump' vs. 'on'. While it is claimed in [22] that this method is not suitable for short-text of sentence length such as tweets because these may have null common words, one could argue that even though tweets are length-constrained, this creates an upper limit on the TF, reducing the importance of that portion of the weighting scheme. However, IDF should still give smaller weights for commonly occurring words in the corpus of all dataset tweets and higher weights for less occurring ones.

Latent Semantic Analysis (LSA) [23-25], Latent Dirichlet Allocation (LDA) [26], and Hyperspace Analogues to Language (HAL) [27] are among the early word co-occurrence statistical models contributing to text similarity computation based on estimating continuous representation of words in a huge corpus. In LSA, a possible number of concepts is extracted from a set of training documents, and a terms to concepts matrix is produced. This matrix is derived by applying Singular Value Decomposition (SVD), which decomposes the original word by document co-occurrence matrix into the product of three matrices, including the diagonal matrix of singular values [28]. The small singular values in the diagonal matrix are deleted to condense all the important features into a reduced dimension vector space. The initial word by document matrix is then reformed from the new smaller dimensional space. LSA acquires word knowledge that spreads in contexts through the process of decomposition and reformation. When LSA is used to calculate tweet similarity, a vector for each tweet is constructed in the reduced dimension space; similarity is then measured by calculating the similarity between these two vectors [23]. LSA will fall short for tweet similarity computation due to two reasons:

1. As the computational limitation of SVD imposes that the dimensionality of the reconstructed word to document matrix is limited is size. Therefore, the reduced dimension space of LSA may not include important words in tweets from an unconstrained domain (and thus not represented in the corpus of training documents).

2. The vector representation of a tweet is likely to be very sparse as the dimension in LSA is fixed and vectors are therefore fixed.
3. LSA does not take into consideration any syntactic information from the two tweets being compared.

Therefore, LSA is considered to be more appropriate for text segments that are larger than the short text dealt with in this work [25].

Another important work in corpus-based word co-occurrence methods is LDA. LDA is closely related to LSA as they are both vector space models that estimate a continuous representation of words from large corpora. However, LDA is a topic modeling method that performs different computations implementing three-layer Bayesian probabilities, composed of word, topic, and text. LDA is based on the notion that every document is a mixture of multiple hidden topics, and each hidden topic is a mixture of multiple words. A polynomial distribution defines the relation between topics and words and a Dirichlet prior distribution defines the relation between documents and topics. LDA assigns different topics to the words in a document and therefore each document is represented by a set of topics. The similarity between two documents is then computed according to the extent to which their topics are similar to each other. As with LSA, LDA falls short when applied to tweet similarity because, the idea behind LDA is that it assigns relevant topics for each document based on its context, and as tweets lack context due to brevity and are too sparse, it will yield poor representations [29].

HAL is another corpus-based method that is indeed similar to LSA. They both use lexical co-occurrence information to capture the meaning of text. Unlike LSA, HAL builds a word by word matrix instead of the word by document matrix. The former is built based on words that co-occur within a predefined moving window. This window moves over the entire corpus and constructs an N × N matrix for the N unique vocabulary in the corpus. Word co-occurrences within the moving window across the entire corpus is recorded in each entry of the matrix. The corresponding row and column in the word by document matrix are combined to represent the meaning of a word. Consequently, the word vectors for all words in the text segment are added together to build a representative vector for the text segment. The similarity is then computed between two representative vectors using a measure such as cosine similarity. Unlike LSA and LDA, HAL is memory-intensive as it does not perform any dimensionality reduction technique and therefore can be too resource-intensive when used in applications processing big datasets such as tweets.

In summary, as LSA, topic models, and HAL have been powerful in discovering latent semantic structures and traditional tasks for long document similarity computation, they fail in modeling tweets due to the severe sparseness and noise present in them [29, 30].

Based on the idea of corpus-based statistics, prediction based distributed representation of words learned by neural networks has emerged, generating dense and continuous valued vectors called *embedding* [31, 32]. These embedding of words have become one of the strongest trends in machine learning and NLP to represent sparse and high dimensional data in a vectorial space of semantic features [33]. Prediction based word embedding models, such as *word2vec* [32, 34] and *GloVe* [35] is gaining more attention over classical frequency-based vector representation models such as LSA, LDA, and HAL. Word embedding provides a more expressive and efficient representation of words by preserving their contextual similarity and constructing low dimensional vectors [36]. In word embedding, an unsupervised learning approach is performed on a huge corpus to learn word representations using a neural network. Naili, Chaibi [36] reported that prediction-based word embedding models outperform the classical counter-based word vector representation in LSA. Furthermore, it has been reported that Word2Vec outperform GloVe for both English and Arabic languages [36].

There is not much research conducted in OSN analysis using word embedding, particularly for tweet similarity computation. De Boom, Van Canneyt [37] trained a Word2Vec model on a dataset of 10 million Wikipedia couples to learn semantic similarities for short text fragments. The authors denote couples as *pairs* only if they are similar according to a benchmark otherwise they are *non-pairs*. Their proposed method combines knowledge from tf-idf and word embedding to measure the semantic

similarity between two fixed length pairs. The degree to which two pairs are semantically similar depends on the degree of similarity between their corresponding vector representations according to some distance measure. Their results show that Word2Vec vectorial representation of words, combined with tf-idf weightings might lead to a better model for semantic content within very short text fragments. Nevertheless, this conclusion needs further investigation for application in the context of Twitter. This is because Wikipedia contains structured information and is completely different textual platform than a social medium such as Twitter, in which the content is mostly slang, abbreviated and erroneous [37]. Moreover, the results are derived for short text of fixed length and have not analysed text of arbitrary length such as tweets.

Dey, Shrivastava [38] proposed a word embedding training model for single and multiple hashtags recommendation towards tweets. They developed one model for learning the embedding of each word in the corpus vocabulary and another model for learning the embedding of each word in the scope of an accompanying hashtag. Using word embedding, their system demonstrate a lift of 7.48 and 6.53 times for recommending a single hashtag and multiple hashtags to a given tweet respectively.

The observed literature around word embedding in the context of Twitter-based semantic textual analysis indicates and reveals potential capabilities of such techniques for OSN analysis. However, word embedding has not been used in semantic representation of tweets in the scope of semantic similarity computation. In addition, while syntactic information contributes to the overall meaning in a text fragment [12], most of the aforementioned methods consider only semantic information when computing the similarity. As discussed in section 2, texts present in Twitter can be challenging for knowledge-based methods as most of the terms used in Twitter are not present in a structured and formal language ontology. Furthermore, tweets are challenging for classical vector representations and topic modelling methods due to the inadequate information and lack of context for manipulation by a computational method [39]. Therefore, this research proposes a novel semantic similarity measure for tweets, addressing the limitations of existing methods and filling the gap of word embedding in the scope of microblogging OSN, particularly Twitter. This algorithm performs similarity computations by creating a dynamic vector that integrates semantic knowledge from word embedding and syntactic characteristics of the pair under consideration.

3. The Proposed Tweet Similarity Approach

The proposed hybrid approach consists of semantic and syntactic components that extract corresponding information from the compared tweets and derive similarity. A tweet is composed of maximum 280 characters considered to be a sequence of words hashtags, mentions, and URLs. The combination of words and hashtags in a tweet, along with their syntactical structure, make a tweet convey a specific meaning. However, the high level of noise present in tweets discussed in Section 2 requires a sound preprocessing methodology that reduces noise, yet preserves the information that contributes to the meaning of a tweet.

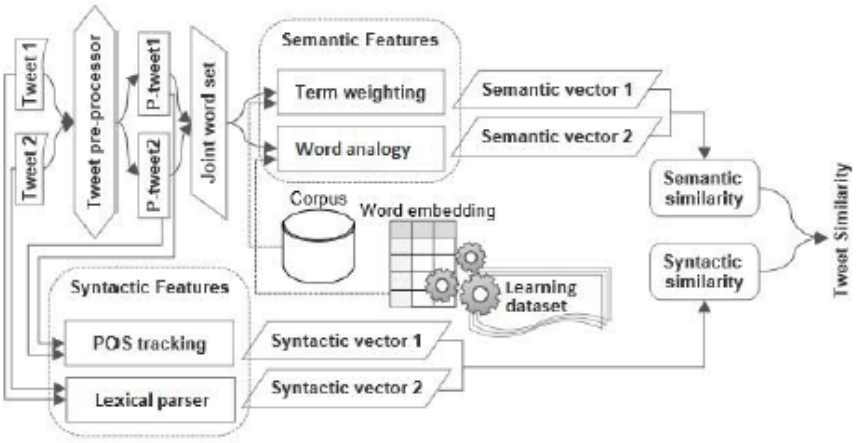


Figure 1. Tweet Similarity Computation Structure

Figure 1 presents the process undertaken for tweet similarity computation between two tweets being assessed for similarity. After going through pre-processing stages (Section 4.2.2), the proposed method generates a dynamic representation of the pair of tweets consisting of the distinct words in them. For each tweet, a semantic and a syntactic vector is constructed. The semantic vector is derived using a pre-trained word embedding model and the value of each term is calculated by applying a weighting scheme using a corpus. The syntactic vector is formed in the syntactic component, which extracts features that describe the syntactical structure of a tweet. The semantic and syntactic similarities are computed by calculating the distance between their corresponding vectors. Finally, the overall similarity between a pair of tweets is derived by combining the output of the semantic similarity and syntactic similarity. The subsequent sections present a detailed description of each component in the proposed tweet similarity algorithm.

3.1 Semantic Decomposition

Several STSS approaches have been proposed in the last decade, which include a number of semantic similarity methods that have contributed in the development of some useful computational intelligence applications [11, 12, 37, 40, 41]. These methods are typically categorized into three groups: ontological-based (lexical database / dictionary-based) methods, corpus-based (information theory-based) methods, and word embedding-based (word co-occurrence / statistical-based) methods; a detailed review on short-text similarity can be found in [39]. After a comprehensive investigation of the developed methods and an analysis of the user generated content in OSN, a novel semantic similarity computation algorithm that is composed of semantic and syntactic components, namely TREASURE, is proposed in this research. This measure has been evaluated against state-of-the-art methods and provides the best correlation to human judges for a benchmark dataset (Section 5.2.3). This section describes the semantic decomposition process computed in the semantic similarity component of the measure.

3.1.1 Word Analogy

Word embedding projects in computational linguistics encode meanings of words to low dimensional vector spaces. Unlike traditional distributional semantic vector space models such as LSA and LDA, these recent techniques generate dense, continuous valued vectors, called *embeddings*. Word embedding approaches have become the state-of-the-art performances in many intrinsic NLP tasks such as semantic textual similarity [37] due to their potential in capturing the semantic relations among words. The process of learning embeddings include neural network-based predictive methods, such as Word2Vec [34, 42] and count-based matrix factorization methods, such as GloVe [35].

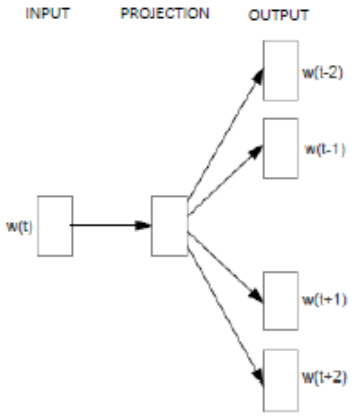


Figure 2. Skip-gram model architecture [34]

In this study, the shallow word embedding model, Word2Vec, is used as the source algorithm for learning dense word vectors. The 1st experiment uses Google's pre-trained model [32] (Section 4.3.1) on part of Google News dataset and the 2nd experiment uses the Political Tweets pre-trained model [43] (Section 4.3.2) on EU_Referendum dataset. Both pre-trained models use a skip-gram architecture shown in Figure 2. The skip-gram model predicts surrounding words $c_1, c_2, \ldots, c_n$ given the current word w (n is the size of the context window), such as $P(c1|w)$, $P(c2|w)$, and etc. The resulting trained embedding model consists of a word embedding vector denoted by $\check{v}$, for each word w in the model.
Given two words w_1 and w_2, the word analogy module computes the semantic similarity $S_{sem}(w_1, w_2)$. This is done by calculating the cosine coefficient between the two corresponding word embedding vectors $\check{v}_1$ and $\check{v}_2$ for w_1 and w_2 in the semantic embedding space. For example, the cosine similarity between $\check{v}_{Obama}$ and $\check{v}_{president}$ in the Google News pre-trained Word2Vec model is 0.31.

3.1.2 Weight Transformation

Unlike most text similarity algorithms, the proposed measure retain all stop words. However, as these words occur frequently, they contribute less to the meaning of a tweet than other words. Similarly, different words contribute differently towards the meaning of a tweet. The significance of a word is determined according to the assumption that words occurring more frequently in a corpus contain less information than less frequently occurring words [14]. Thus, the extent in which terms contribute to the overall meaning in a tweet is determined by how frequently they occur in a given corpus of tweets. The terms that occur more frequently tend to have less value compared to less frequent terms. However, common weighting techniques such as *tf-idf* falls short in favoring discriminatory traits over nondiscriminatory ones in a tweet. This is due to the short and constrained nature of tweets, which creates an upper limit on the term frequency reducing its importance in the weighting scheme. Moreover, the massive size and creative vocabulary generated by Twitter users makes the representation of tweets in *tf-idf* vectors sparse and less accurate. Therefore, the weight of a term (i.e. information it carries) is derived from calculating its probability in a corpus using a compound method as follows:

1. Chi-squared test is computed to capture two-word phrases (i.e. bigrams) that are not likely occurring together by random chance:

$$chisq_{x,y} = N * \emptyset^2 , \quad (1)$$

Where ϕ is essentially a normalized sum of squared deviations between the expected and observed frequencies, N is the total number of tokens in the corpus. The theoretical frequencies are derived from the base probabilities of every term appearing in the text. Whereas the

observed values come from the frequencies of the corresponding bigrams. Nltk's module of bigram association measure has been used to compute this test. This method not only captures intuitive phrases like '*thank you*' and '*I am*', but also the multifaceted composition of Twitter which describe certain event of phenomena, such as "#eureferendum", "#voteleave", and "#strongerin".

2. The probabilities of the bigrams and remaining words in the corpus is computed as the relative frequency:

$$\hat{p}(g) = \frac{n+1}{N+1}, \tag{2}$$

Where n is the frequency of the n-gram g in the corpus, and N is the total number of n-grams in the corpus (increased by 1 to avoid the case of undefined value). Weight of g in the corpus is defined as:

$$W(g) = 1 - \frac{\log(n+1)}{\log(N+1)}, \tag{3}$$

So $W \in [0, 1]$.

The composite weighting scheme is implemented to build a complex model of diverse and more accurate vectorized representation of tweets. The semantic similarity $S_{sem}(w_1, w_2)$ between words w_1 and w_2 is therefore a function of word embedding e and word weight h as follows:

$$S_{sem}(w_1, w_2) = f(e, h), \tag{4}$$

Where e is the cosine angle between embedding vectors $\vec{v}_1$ and $\vec{v}_2$ for words w_1 and w_2 in the pre-trained embedding model, h is the weight of w_1 and w_2 calculated following equation (3). The authors assume that (4) can be rewritten using two independent functions as follows:

$$S_{sem}(w_1, w_2) = f_1(e) \cdot f_2(h), \tag{5}$$

Where f_1 and f_2 are transfer functions of word embedding similarity and weighting scheme respectively.

3.1.3 Scaling Word Vectors Distance

Word embedding models generate word vector representations based on performing iterations over the training corpus in order to learn words co-occurrences in a predefined context window size. Thus, even highly dissimilar words tend to share commonalities in their distributed word vector representations. This behavior should be taken into account in calculating $S_{sem}(w_1, w_2)$ in order to avoid introducing noise to the semantic vector. Li, McLean [12] performed depth scaling of words in hierarchical semantic nets such that similarity of words at upper layers are scaled dawn and similarity of words at lower layers is scaled up. Similarly, scaling is performed to the similarity of words in TREASURE where the cosine coefficient of their corresponding vectors in the pre-trained embedding models is less than a certain threshold. A scaling parameter is defined as α, where $\alpha \in [0, 1]$. The optimal value of α is dependent on the word embedding model used and can be determined through the using a benchmark word pairs dataset with human similarity ratings. Empirical experiments were conducted to determine the optimal threshold value for the pre-trained embedding models used in the word analogy module, which turned to be $\alpha = 0.3$ for the proposed measure.

3.2 Syntactic Decomposition

Unlike semantic similarity methods, which only take into consideration the similarity derived through topological or statistical semantic computations, the proposed measure not only considers semantic interpretation, but also accounts for the contribution of the morphological structure of terms occurring in a tweet. Syntactical features are particularly important in social contexts such as Twitter because, although tweets are unstructured texts, users in Twitter often express their meaning using common conventions and certain punctuations due to the restriction over character limit. Therefore, ignoring

such features leads to missing nuggets of information in the representation of the feature vector for each transformed tweet. The syntactic component consist of the *POS* module, which captures derivational morphology structures of content words as well as the *lexical parser* module for Twitter-specific features.

3.2.1 POS Tracking

Word embedding models capture statistical semantics between words based on the distributional hypothesis that words occurring in similar contexts tend to have similar meanings, where all words are processed in similar manner. Such models also discard derivational morphology between words, such as the noun *beauty* and the adjective *beautiful*. To incorporate structural information, a syntactical feature vector of size 6 is constructed for each tweet to capture stop words, nouns, verbs, adjectives, adverbs, and digits respectively. Unlike most existing methods that ignore function words in similarity computation, the proposed approach includes these as they carry structural information [12], which contributes to the meaning in short texts such as tweets. However, function words contribute less to the meaning of a tweet as they appear frequently and therefore their value will be scaled down as discussed in Section 3.1.2. The POS tracking module tags each token in a tweet and populates its corresponding vector. For example, T_1 *'what a nicely written story!'* and T_2 *'is chapter 2 well structured?'* are represented in the syntactical vector space as [1, 1, 1, 0, 1, 0] for T_1 and [1, 1, 1, 0, 1, 1] for T_2. The syntactic similarity between T_1 and T_2 is the cosine between their vectors, which is 0.89. This computation is performed for candidate tweets and their syntactic similarity is derived by calculating the cosine angle [44] between their corresponding syntactic feature vectors.

3.2.2 Lexical Parser

Common Twitter conventions and punctuations are most likely to be omitted in methods of semantic inferences in social data. However, in this research, the authors hypothesis is that these symbolic structures are of no less importance than words in social contexts. Therefore, these symbolic conventions and punctuation provide information that cannot be discarded. This is particularly true in Twitter as users do not often follow a grammatical structure in tweets due to the informal nature of the social network. For example, consider the two tweets T_1 *'going to Rome this weekend!'* and T_2 *'going to Rome this weekend?'*, although both tweets are constructed from the same words, punctuating them differently changes the complete function of the tweet. The exclamation mark in T_1 expresses the user's excitement, whereas T_2 is an interrogative sentence expressing the user's uncertainty. Another common use in informal contexts such as Twitter (albeit out of scope) is the sarcastic case. To further elaborate the role of expressive punctuations (i.e. interrogation and exclamation marks) in Twitter, the tweet *'Do I really need to mention this again!'* has a latent rhetorical interrogation mark that indicates intended sarcasm.

Table 1. Syntactical features in a tweet

Feature Id	Syntactical group	Feature
1	*POS tags*	Stop word
2		Noun
3		Verb
4		Adjective
5		Adverb
6		Digit
7	*Twitter conventions*	Hashtag
8		Mention
9	*Punctuation marks*	Interrogation
10		Exclamation
11	*Special symbols*	Currency
12		Ratio

While highlighting the role of expressive punctuation marks in Twitter demonstrates their importance in delivering the overall meaning of a tweet, common Twitter conventions (e.g. *#hashtags* and *@mentions*) are taken into account as well. Hash-tagging timely events and mentioning users over the network are frequently apparent in Twitter and almost every tweet contains at least one of them. The lexical parser module breaks down the tokens in a tweet and produces a list of the hashtags and mentions. Furthermore, special symbols (e.g. $ and %) are prevalent in tweets and carry syntactic information that cannot be ignored. The syntactical feature vector discussed in Section 3.2.1 is thus extended to accommodate further syntactical features, which are expressive punctuation marks, Twitter-based conventions, and special symbols. The complete list of syntactical features are provided in Table 1.

3.3 Semantic Similarity between Tweets

After running tweets into the preprocessing module discussed in section 4.2.2, a tweet is decomposed into words and symbolic structures. Unlike classical methods that represents a sentence using only unigram features contained in it, the proposed method dynamically forms semantic and syntactic vectors solely based on the compared tweets. Recent research achievements in the complex field of computational linguistics and social media data are adapted as well to construct an efficient method of transforming a tweet into a representative semantic and syntactic feature vectors.

Given two tweets, T_1 and T_2, the proposed tweet similarity measure (TREASURE) forms a joint word set, from which the lexical semantic vectors are derived. The joint word set takes the following form:

$T = T_1 \cup T_2 = \{ w_{1_{T_1}}, w_{2_{T_2}} \dots w_m \}$.

Where the joint word set T consists of all the unique words from T_1 and T_2. Unlike existing methods that consider different forms of a word such as *mouse* and *mice*, *cat* and *cats* which are considered as four distinct words in the joint word set T [12], the proposed measure inserts the root of the word in T for two reasons:

1. Unlike derivational morphology discussed in section 3.2.1, in which the grammatical category of a word is changed, inflectional morphology does not change the essential meaning of a word.
2. Adding different forms of a words in the joint word set creates sparse vectors and introduces noise to the similarity computation algorithm.

Thus, the joint word set, T, for the two tweets, T_1 'EU Referendum briefing on living and working in the UK #ProtectJobs' and T_2 'You must stay in the #EU to protect your job!', is:

T = {EU Referendum briefing on living and working in the UK Protect Job you must stay to}.

Tracing shared words in the candidate tweets back to their morphemes in the joint word set creates a compact set with no redundant information. The joint word set, T, can be considered as the semantic features in the candidate tweets. Therefore, each pair of tweets is semantically represented by the use of T as follows: the joint word set is used to derive the lexical semantic vector, denoted by $\check{s}$, where each entry corresponds to a word in T. thus, the dimension of the semantic vector, $\check{s}$, is equal to the length of the joint word set (i.e. number of words). The lexical semantic vector is denoted by v_{sem}, and values in the lexical semantic vector, $\check{s}_i (i = 1, 2, 3, \dots, n)$, is derived by computing the semantic similarity of the corresponding words embedding vectors $\check{v}_i$ in the tweet. Considering T_1 as an example:

Case 1. If w_i is contained in the tweet T_1, $\check{s}_i$ is set to 1.

Case 2. If w_i does not appear in T_1, the cosine coefficient is computed between the word embedding vector $\check{v}_i$ for w_i and each embedding vector corresponding to every word in the tweet T_1, using the method presented in Section 3.1. The highest similarity score ς obtained denotes the most similar word in T_1 to w_i if ς exceeds α threshold discussed in Section 3.1.3; otherwise, $\check{s}_i$ is set to 0.

Following the weighting schema discussed in Section 3.1.2, the value of an entry in the semantic vector becomes:

$$s_i = \check{s}_i \,.\, W(g_i) \,.\, W(\tilde{g}_i), \tag{6}$$

Where $W(g_i)$ is the weight of an n-gram (i.e. a word or a two-word phrase) in the joint word set, $W(\tilde{g}_i)$ is the weight of the most similar n-gram to g_i in the tweet. The product of the similarity and weights of g_i and $\tilde{g}_i$ allows this entry of the semantic vector to contribute to the overall similarity based on their individual value. The semantic similarity between two tweets is derived by computing the cosine coefficient between the two semantic vectors corresponding to the tweets under consideration:

$$S_{sem}(T_1, T_2) = \frac{v_{sem}(T_1) \,.\, v_{sem}(T_2)}{\|v_{sem}(T_1)\| \, \|v_{sem}(T_2)\|} \tag{7}$$

It is worth noting that the proposed measure does not take into account the order of the words occurring in a tweet. This is based on two considerations: first, in tweets, unlike formal English sentences, users often use relaxed informal expressions that lack English grammatical structure rules. The character limit restriction impose users misplacing adjectives and adverbs (e.g. *old silly fool* instead of *silly old fool*) and cutting off elements such as pronouns and conjunctions (e.g. *voting leave?* instead of *are you voting for leave?*) while supporting their meaning with emoticons for an ultimate usage of characters. Therefore, although English is not a free word order language, the free grammar nature of Twitter reduces the significance of word order in analyzing the semantic and syntactic structure and its contribution to the overall similarity between two tweets. Second, the proposed model is composed of multiple modules to account for the necessary semantic and syntactic fragments of a tweet, at deferring computational costs, and incorporating a word order similarity module would scale up the complexity even further. The results indicate promising results of the proposed approach as shown in Section 5.2.3. Users deviating from proper grammatical rules of the English language in social contexts and how this can be further analyzed to optimize computational linguistic applications is part of future research.

3.4 Syntactic Similarity between Tweets

The syntactic similarity between two tweets is a combination of various syntactical features as discussed in Section 3.3. As a tweet enters the syntactic decomposition module, it is lexically parsed, tokenized, and tagged according to the POS it contains. The tweet is then represented by a syntactic feature vector, which transforms the syntactic information held in the tweet into a numeric vectorized representation. Consider a pair of tweets, T_1 and T_2, and their corresponding syntactic feature vectors, $v_{syn}(T_1)$ and $v_{syn}(T_2)$ as follows:

T_1: An absolute disgrace! & again British kids get nothing!! #Brexit

T_2: Is @David_Cameron secretly taking us into another war while eyes are on #Brexit?

$v_{syn}(T_1)$: [3, 4, 1, 2, 0, 0, 0, 3, 1, 0, 0, 0]

$v_{syn}(T_2)$: [4, 3, 2, 0, 1, 0, 1, 0, 1, 1, 0, 0]

The syntactic feature vector, $v_{syn}(T_1)$, is derived by calculating the syntactic features (as shown in Table 1) in T_1, and similarly for T_2. The syntactic similarity between $v_{syn}(T_1)$ and $v_{syn}(T_2)$ is therefore a function of POS tags and lexical parsing of common Twitter convention. It is derived by computing the cosine coefficient between the syntactical feature vectors $v_{syn}(T_1)$ and $v_{syn}(T_2)$ as follows:

$$S_{syn}(T_1, T_2) = \frac{v_{syn}(T_1) \,.\, v_{syn}(T_2)}{\|v_{syn}(T_1)\| \, \|v_{syn}(T_2)\|}, \tag{8}$$

The overall similarity between a pair of tweets is a combination of semantic and syntactical similarity at variable contributions, which are determined by empirical experiments.

3.5 Overall Tweet Similarity

As discussed earlier, the syntactical analogy between tweets plays a role in conveying the meaning of tweets. Therefore, the latter is a combination of semantic and syntactic similarity, each contributes according to its significance to the overall similarity score. The semantic similarity represents the potential meaning between words constructing a tweet, while the syntactic similarity provides information about the morphological structure of the words and common Twitter conventions used. Hence, the overall tweet similarity is defined as a combination of semantic similarity and syntactic similarity as follows:

$$S(T_1, T_2) = \delta S_{sem} + (1 - \delta) S_{syn}$$
$$= \delta \frac{v_{sem(T_1)} \cdot v_{sem(T_2)}}{\|v_{sem(T_1)}\| \|v_{sem(T_2)}\|} + (1 - \delta) \frac{v_{syn(T_1)} \cdot v_{syn(T_2)}}{\|v_{syn(T_1)}\| \|v_{syn(T_2)}\|} \quad (9)$$

Where $\delta \leq 1$ determines the relative contributions of semantic and syntactic information to the overall similarity score. However, is has been reported that syntactic information carry subordinate value for semantic processing of text [45], δ should therefore be a value larger than 0.5, i.e., $\delta \in (0.5, 1]$ [12].

4 Implementation Using Word Embedding and Statistical Analysis

Two Twitter-based annotated datasets were used for the evaluation of the proposed tweet similarity computation measure. This section briefly describes these two datasets and presents an illustrative example of deriving the overall tweet similarity demonstrating the model's processing stages.

4.1 SemEval Tweet-News Dataset

SemEval is a collection of online computational semantic analysis shared tasks intended to explore the natural meaning in different languages. Part of SemEval-2014 shared task comprises of a published STS.tweet_news training and testing dataset [46] that are labelled with human similarity judgments. The dataset consists of 750 tweet-news pairs harvested during the period of 11th of Jan to the 27th of Jan, 2013. This benchmark adopted a 6-point Likert scale to assign the degree of similarity score between pairs as defined by Agirre [47]:

(0) On different topics.
(1) Not equivalent, but are on the same topic.
(2) Not equivalent, but share some details.
(3) Roughly equivalent, but some important information differs/missing.
(4) Mostly equivalent, but some unimportant details differ.
(5) Completely equivalent, as they mean the same thing.

This dataset provides human similarity labels on pairs of tweets and news headlines, which are both short text in a general domain. However, the lack of domain specific tweet pairs that reflect the actual noise and challenges in Twitter posts necessitate constructing a new dataset. The subsequent section describes the procedures undertaken from data collection to benchmark production of a rich controversial domain dataset. Nevertheless, the former dataset is used to evaluate TREASURE along with the domain-specific dataset in order to validate the proposed approach's generalizability to different domains.

4.2 Political Tweets Dataset

This section describes the methodologies used in this study for collecting and building an annotated dataset from Twitter microblog. It provides a description of the dataset in terms of size, attributes, and tweets harvesting and cleaning methodology.

4.2.1 Data Collection

In this study, the political domain of the EU Referendum is considered, as it has been an active trend in OSNs and a rich source of controversial views. The United Kingdom European Union Membership

(known as EU Referendum) took place on the 23rd of June 2016 in the UK. Based on a voting criteria, the voters were exposed to two opposing campaigns supporting remaining or leaving the EU. Three months prior to the day of the referendum, the data collection process has commenced through the use of Twitter Application Programming Interface (API), and went on until one month past that day. The Twitter streaming API allows for establishing a connection and continuously streaming real time tweets according to a specified set of search terms. Communicating with the Twitter platform was made possible via the Open Authentication (OAuth) mechanism. This mechanism requires an application registration on the Twitter platform beforehand. Kumar, Morstatter [48] provide a comprehensive overview of the authentication process required by the Twitter API. Amongst various programming languages that interface with the API, Python has been used for its flexibility and prebuilt selection of Twitter software packages and NLP libraries.

Twitter streamed instances are returned as JavaScript Object Notations (JSON) data structures, which are composed of multiple metadata per tweet. These JSON objects were stored in a NoSQL database called MongoDB [49]. MongoDB is used as it is a fully scalable non-relational database, intended for storing unstructured data, such as text, as documents instead of tuples in tables. It has been trusted by several web 2.0 big data sites such as Foursquare, Disney Interactive Media Group, The Guardian, GitHub, and Forbes [49]. The entire 1.2TB text corpus of Wordnik online social dictionary [50] is also stored in over 5 billion MongoDB records. In this study, the documents inserted into MongoDB are the tweets JSON objects that were retrieved by Twitter API.

A dataset of 4 million tweets was collected and stored in MongoDB. Each instance in the dataset is a tweet associated with multiple metadata. These metadata contain information relating to the tweet, users, and entities.

4.2.2 Data Preprocessing

An important part in creating meaningful dataset for accurately evaluating the proposed method is data preprocessing and filtering. This research follows a 2-stage preprocessing methodology in order to generate a semantic-rich set of tweets in the political domain under consideration.

1. In the first part, the proposed algorithm follows the pre-processing heuristic proposed by Alnajran *et al.* [51] for cleaning short text in OSN. This preprocessing methodology takes into consideration the characteristics of the text and common conventions (e.g. retweets, hashtags, mentions, etc.) in Twitter. It eliminates redundant tweets as well as features that can be misleading for the similarity computation algorithm, yet preserves important information. For example, words containing repeated letters to emphasize sentiment are standardized to their original form. This enables the Part-of-Speech (POS) tracking and word analogy sub components to trigger them and perform similarity computations. The application of the first preprocessing stage on the raw tweets dataset generated 4 million candidate tweets.
2. As the aim of this study is to develop a tweet semantic similarity computation method, a structure-based filtering is performed [52]. This filtering process eliminates tweets that do not carry sufficient clean textual features for semantic processing. Tweets containing more than 2 user mentions or more than 3 hashtags, or less than 5 text tokens are filtered out. This is based on the idea that tweets containing too many twitter-based user conventions such as hashtags and mentions, and less semantic content are generally very noisy [52]. Even though the study mentioned that hashtags are considered as noisy as user mentions, we believe that hashtags contribute to the meaning of a tweet. Therefore, we favor hashtags and set their acceptable occurrence threshold to three instead of two. Consequently, the tweet length restriction is increased by 1 in order to accompany the additional allowance of hashtags. Due to the nature of the political tweets of the active EU Referendum event, this stage filters out most of the tweets and keeps only 137K semantic-rich tweets that satisfy the invasive preprocessing criteria.

4.2.3 Unsupervised Sampling Methodology

A benchmark is ideally generated by human judges with a good level of inter-rater agreement [53]. However, the production of similarity judgments for the whole dataset of collected tweets is a labor intensive process. Furthermore, manually generating pairs of tweets from the whole dataset to cover the Likert scale of six similarity scores described in Section 4.1 is extremely expensive, if not impossible. Therefore, an unsupervised approach is required to derive a representative sample set of the political tweets in order to reduce judges' recruitment expensive process for generating the benchmark dataset. A recursion-based incremental clustering is implemented using the proposed similarity measure. The goal of this cluster analysis application on the tweets dataset is twofold:

1. Generating pairs of tweets using the resulting centroids and tweets (i.e. observations) at different distances to the cluster center to form pairs of tweets. The produced pairs of tweets are used for building the benchmark of human judgments on similarity. This benchmark is then used for intrinsic evaluation.
2. Analysis of the generated clusters provides an extrinsic evaluation of the proposed tweet similarity method as it has been used in allocating tweets to the most similar cluster (i.e. clustering distance measure).

The clustering algorithm is implemented following a divisive approach such that all observations in the dataset start in one cluster. The cluster analysis commences by assigning a random observation, T, as a cluster center. A recursive series of splits are subsequently performed based on comparing each observation with the derived centroids. An observation, T, is assigned to a predefined cluster if it satisfies a certain threshold, δ. Otherwise, a new cluster is generated and T is assigned as the new cluster's centroid. This process recursively carries on until all observations in the dataset are assigned in clusters. Unlike most clustering algorithms that require the number of clusters to be determined beforehand, such as *k*-means, this approach does not apply this condition. Instead, the number of clusters in the dataset is directly proportional to the specified similarity threshold. This linear relationship implies that as the value of the threshold increases, more clusters are generated and vice versa. Based on an experiment conducted on a similarity-labelled Twitter dataset [46], it has been empirically determined that a value of $\delta = 3$ yields the best set of clusters in terms of cohesion and separation.

However, a cluster analysis of the whole dataset would be a complex and time consuming process. Therefore, a subset of the whole corpus of collected tweets is derived, such that the complete timeframe for the data collection process is spanned. Although it has been reported that a 10% of a dataset is considered a representative sample set [54], collecting a random 10% of the whole dataset may introduce bias in the resulting tweets and miss out on important events.

Thus, the methodology for building as representative subset is conducted as follows:

1. The corpus of pre-processed tweets is divided into 4 groups according to the month a tweet has been streamed.
2. For each month during the data collection, the group of corresponding tweets is further split into 4 groups according to the week of tweet streaming.
3. The result is a corpus of tweets organized into 4 main groups corresponding to the 4 months of data collection and each group contains 4 subgroups according to the week a tweet has been streamed.
4. The representative subset is created by retrieving a random sample of 10% from each of the 16 subgroups in order to span the entire data collection period.

This sampling methodology resulting in 13.7k tweets, not only ensures a representative set is collected in terms of size, but in content as well. The clustering algorithm is applied on the representative sample of tweets using the proposed similarity measure at a similarity threshold, $\delta = 3$. The unsupervised approach generated 11 non-overlapping clusters as summarized in table 2.

Table 2. Cluster analysis of political tweets

Cluster id	Representative tweet (centroid)	Cluster size
1	*Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today*	2731
2	*EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats*	1840
3	*Sterling slides on renewed Brexit worries*	1719
4	*Brexit Emerges As Threat to TTIP Deal*	1682
5	*It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union*	1524
6	*Should the United Kingdom remain a member of the EU or leave the EU?. Opinium poll: Remain: 49% (-3) Leave: 51% (+3)*	1243
7	*Erdogan is an Islamic extremist who will flood the EU w #jihadists. Kick Turkey out of NATO and no admission to the EU. #Brexit*	987
8	*Both #HillaryClinton and #Obama continue to call on UK not to leave EU? If not EU #terror movement limited!*	688
9	*Brexit introduce controlled immigration system, deport those who support extremism*	604
10	*Terrorism is the scariest think. And it's ways more scarier if it's in the EU, in your home. Stay strong Brussels! #prayersforBrussels*	421
11	*It's just utterly stupid. Thank god UKIP will never get in power and Brexit will fucking fail.*	295

4.2.3.1 *Annotation Experiment Design*

In psychology, the capacity of information, i, that can be received, processed, and remembered in immediate memory of a typical human cognitive system is seven plus or minus two [55], that is $i \in r$, where $r = \{5, 6, 7, 8, 9\}$. The methodology of producing the benchmark of similarity judgments on the political tweets dataset is based on this psychological theory. In order to make the annotation task as simple as possible for participants to complete, the experiment has been designed according to the results of the cluster analysis described in Section 4.2.3.

1. Each representative tweet, T, which correspond to the five biggest generated clusters are used to form one part in the pairs of tweets.
2. For each representative tweet, 6 tweets are randomly selected from the dataset and assigned to make up a pair.
3. This subsampling process is performed for each representative tweet in the biggest five generated clusters.
4. The resulting 30 pairs of tweets are used to form the benchmark dataset as shown in table 3.

This sampling methodology is performed to prevent any bias being introduced by selecting the pairs included in the test data and also to avoid reliance on the TREASURE to perform the selection, which has not been evaluated by human experts yet.

Table 3. Tweet pairs used in the annotation experiment

Pair id	Tweets	Representative tweets
1	Brussels attacks may sway Brexit vote: Strategists	*Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today*
2	On one hand, there are decent human beings that send their sympathies to the Brussels victims and their families. And then there's Brexit.	
3	#Brussels attacks: Terrorism could break the EU and lead to Brexit	
4	Terrorism is the scariest think. And it's ways more scarier if it's in the EU, in your home. Stay strong Brussels! #prayersforBrussels	
5	Brussels Attacks Spur Brexit Campaign: Anti-Immigration Parties Link Terror To EU Open Borders	
6	The world is seriously fucked up right now.	

7	@caddenlimos connecting low paid workers doing non skilled jobs from Poland with terrorism in Belgium	*EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats*
8	Has anyone from the Brexit Brigade addressed the issue of what will happen to existing EU citizens living and working in the UK	
9	@thebobevans Today's atrocity foreseable under EU policy. Trust UK security services to protect UK citizens. Brexit	
10	#Brexit supporters claim EU needs UK more than we need it. 45% of UK exports go to EU, 10% of EU exports come here	
11	Could 2m+ 18-34 Year Old Workers Emigrating After a Brexit Cause a Recruitment Nightmare?	
12	We must stay in #EU to protect jobs	
13	Sterling slides on renewed Brexit worries - A decline of 1% marks the 25th day this year the pound has moved	*Sterling slides on renewed Brexit worries*
14	London-based crowdfunding platform Seedrs poll on the EU referendum finds 47% of investors and 43% of entrepreneurs	
15	Brussels bombing rose Brexit risk. How to trade Pound in case of Brexit: GBP $USD #FX	
16	Sterling has dipped cause markets believe Brexit will happen-GOOD-spivs in the city will adjust after playing their gambling games	
17	In most scenarios #Brexit will impose a significant long-term cost on the UK economy #OEBrexit	
18	it's not just an economic argument	
19	Jobs rely on trade NOT a political union. Security relies on sharing information NOT a political union. #Brexit #StrongerIn #VoteLeave	*#Brexit Emerges As Threat To TTIP Deal*
20	#Brexit, a new threat to TTIP transatlantic trade talks	
21	Naive to think Brexit would solve the #ttip problem...only way to protect #NHS is for govt to exclude it from TTIP	
22	Benign Brexit would require accepting high levels of immigration and deep trade agreement with EU	
23	Brexit Risks Rising	
24	Negotiating trade agreements after #Brexit would be complicated for UK as there's no @wto for #services: @angusarmstrong8 at @FedTrust event	
25	UK's NHS will NOT survive staying in the EU	*It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union | via @Telegraph*
26	What would #Brexit mean for the #pharma industry?	
27	To the "expats" in spain who are moaning about immigration can i just say this to you? Jog the fuck on you UTTER hypocrites	
28	How can we save NHS inside EU	
29	We send £350 million to Brussels every week - enough to build a new NHS hospital every week. Let's #VoteLeave and #TakeControl	
30	The EU referendum is a vote for the EU or the NHS, we can't have both	

4.2.3.2 Adaptation of the similarity scale

O'shea et. al. [56] provided evidence that the semantic scale descriptors contribute to more consistent human judgments. The definitions in the similarity scale present in [47] are set for general sentences pairs, in which similarities are easier interpreted and distinguished than tweets. Therefore, a set of descriptors need to be identified to give the best approximation to intervals in a Likert scale for tweets. The 4-point scale validated semantic anchors defined by Charles [57] show a very close agreement between the actual score and desired scores. Agirre, Diab [47], on the other hand, used intuitively chosen scale point definitions for a 6-point scale, but were not validated. The latter is mapped in the annotation experiment to use Charles' validated semantic anchor descriptors in a 6-Likert decimal scale.

Table 4. Adapted semantic anchors for tweets

Scale point	Semantic anchor
0.0	The overall meaning of the sentences is unrelated (on different topics).
1.0	The overall meaning of the sentences is vaguely similar (on the same topic).
2.0	The overall meaning of the sentences is clearly similar (share some details).
3.0	The overall meaning of the sentences is very much alike (missing/different important information).
4.0	The overall meaning of the sentences is strongly related (unimportant details differ).
5.0	The overall meaning of the sentences is identical (equivalent).

The similarity scale points and definitions adaptation is performed in order to come up with semantic anchors that can better interpret the broader semantics in tweet and produce a reliable inter-judge agreement. The adapted 6-point similarity scale for tweets is shown in table 4. The first decimal point is used to introduce finer degrees of similarity [56].

4.2.3.3 Population and Sampling

The aspiration to represent the general population is restricted due to two issues:

1. Participants would be performing the similarity judgement task without supervision.
2. The tweet pairs are rich in political interrelated information and thus require adequate political background to be able to interpret the latent semantics. Younger population, although maybe more familiar with Twitter terminology, they generally have less political background to qualify them in judging such rich semantic pairs.

Thus, it was decided to restrict the sample to adults with graduate-level education. The sample was also restricted to include only Native English speakers to ensure that the language used in the experiment is totally comprehensible and thus similarity judgments would not be influenced by anticipating text meaning or false interpretations. The 32 total participants volunteered without compensation. The similarity judgment experiment does not require collecting any personal information from any participant, such as age or gender, and therefore no sensitive personal data is held.

4.2.3.4 Inter-rater Reliability

The similarity judgments used to produce the benchmark dataset are generated by human observers instructed to rate 30 pairs of tweets for semantic similarity following the 6-point Likert scale described in Section 4.2.4.1. The average of raters' judgments can only be trusted after demonstrating reliability. Inter-rater reliability (IRR) is the level of consensus among raters. Statistical measures are used to provide a logistical evidence that the agreement among raters' subjective assessments is beyond a simple chance [58]. That is, evaluating whether common instructions given to different observers of equivalent set of phenomena, yields the same readings within a tolerable margin of error. The agreement observed among independent observers is the key to reliability [59]. According to [60], the more agreement among observers on the data they generate, the more comfortable we can be that their produced data can be exchangeable with data produced by other observers, reproducible, and trustworthy.

A variety of measures are employed in existing academic research to compute inter-rater reliability. The lack of uniformity among studies is unlikely due to technical disagreement between researchers, but rather due to less sufficient information on how this test is calculated and how the results should be interpreted [61]. In this research, Krippendorff's alpha [59] (KALPHA) is used as it has been suggested to be the standard reliability measure [59]. It handles different sample sizes, generalizes across scales of measurement; can be used with any number of coders, and satisfies the important criteria for a good measure of reliability. Thanks to the work of Hayes and Krippendorff [59], who made computing this test easily accessible by developing a macro to make KALPHA calculation possible in SPSS. A good inter-rater agreement was obtained at $\alpha = 0.8$ for the production of the benchmark described in Section 5.2. Therefore a subjective evaluation of the proposed similarity measure can be

conducted against the expert judgments with a relatively good confidence that the gold standards are reliable enough to make conclusions towards the measure's performance.

4.3 Word Embedding Models

In this research, two word embedding models are used in computing word analogies.

4.3.1 Google's Pre-trained Model

Mikolov, Sutskever [32] trained a Skip-gram Word2vec model on a large dataset of general news articles. The model consists of 3 million vocabulary words. The generated word embeddings are used to calculate word similarities in the developed sematic similarity method. This model is used for evaluation on the labelled twitter-news dataset. The model's corpus metadata and training hyper-parameters are shown in table 5.

Table 5. Corpus metadata and model hyper-parameters for Google News pre-trained model

Metadata and hyper-parameters	Google News Embedding Model
Words in the corpus	100 billion words
Unique tokens in the trained embedding model	V = 3M
Training algorithm	Skip-gram/negative sub-sampling
Vector dimension	$d = 300$
Negative samples	$k = 5$
Minimum frequency threshold	min_count = 5
Learning context window	$w' = 5$
Training time	1 day
Trained model size	3G

4.3.2 Political Tweets Pre-trained Model

Alnajran *et al.* [43] trained Skip-gram Word2Vec neural embedding model on the dataset of political tweets (described in Section 4.2). The architecture of the implemented artificial neural network model is shown in Figure 2. The authors implemented negative sub-sampling of frequently occurring words to decrease the number of training examples (examples that has less information content), and consequently, reduce the computational burden of the training process. The learning process is unsupervised, in which the goal is to learn the weights between the input layer and the hidden layer, which are actually the embedding vector representations of words. The model's time complexity is $O(log2(|V|))$, where V is the vocabulary size. Training the Skip-gram model on the political tweets dataset has taken 27 minutes on Intel core i7 CPU and 16GB RAM. The statistical information on the learning corpus, training hyper-parameters, and processor and memory specifications are shown in Table 6.

Table 6. Corpus metadata and model hyper-parameters for the political tweets dataset

Metadata and hyper-parameters	Political Tweets Embedding Model
Raw tweets	4 million
Words in the corpus	12.3 million
Unique tokens in the trained embedding model (min_count < 3 omitted)	V = 86K
Training algorithm	Skip-gram/negative sub-sampling
Negative samples	$k = 5$
Vector dimension	$d = 300$
Minimum frequency threshold	min_count = 3
Learning context window	$w' = 5$
Training time	27 minutes
Training complexity	O(log2(|V|))
Trained model size	136MB

4.3 Illustrative Example: Similarities for a Selected Tweet Pair

To illustrate how to compute the overall tweet similarity for a pair of tweets using the word embedding model, we provide below a detailed description of our method for two example tweets:

T_1: Sterling falls substantially on #Brexit concerns!
$r_{sem}(T_1)$ = [sterling, falls, substantially, on, #brexit, concerns]
T_2: Is the pound falling on renewed Brexit worries?
$r_{sem}(T_2)$ = [is, the, pound, falling, on, renewed, brexit, worries]

The joint word set is:

T = {sterling falls substantially on brexit concerns is the pound falling renewed worries}.

Table 7. Process for deriving the weighted semantic vector, W(š)

i	*T*(*w*$_i$)	*sterling*	*falls*	*substantially*	*on*	*brexit*	*concerns*		š	Weight (W(*T*(*w*$_i$)))	W(š)
1	sterling	1							**1**	0.5452	0.5452
2	falls		1						**1**	0.6166	0.6166
3	substantially			1					**1**	0.7859	0.7859
4	on				1				**1**	0.279	0.279
5	brexit					1			**1**	0.2426	0.2426
6	concerns						1		**1**	0.5664	0.5664
7	is								**0**	0.2693	0
8	the				0.4765				**0.4765**	0.1967	0.1
9	pound	0. 6455							**0.6455**	0.5184	0.3346
10	falling		1						**1**	0.6001	0.6001
11	renewed								**0**	0.7301	0
12	worries						0. 5059		**0.5059**	0.5930	0.3

The semantic features for T_1 and T_2 can be extracted to from the joint word set, T. The process of deriving the semantic vector for T_1, using the proposed method, is shown in Table 7. In the first raw, the words in tweet T_1 are listed, whereas the first column contains the words, w_i, where $i \in \{1, 2, 3, \ldots, 12\}$, in the joint word set T. The words are sorted according to the order they appear originally. For each word in the joint word set, T, the values in the semantic vector are derived as follows:

1. If the same word exists in T_1, the corresponding cell at the cross point is set to 1.
2. If the root of the word exist in T1, such as 'falls' and 'falling', the corresponding cell at the cross point is set to 1.
3. If the similarities between the word and every word in T1 are computed. The cell at the cross point of the word with the highest similarity is set to their resulting similarity value, if the value exceeds the predefined threshold which is set to 0.3[1].
4. The word is assigned 0 if the highest similar word in T_1 is below 0.3.

For example, the word 'pound' is not in T_1, but the most similar word is 'sterling', with a similarity of 0.65. Thus, the cell at the cross point of 'pound' and 'sterling' is set to 0.65. In the same manner, the word 'on' does not exist in T_1 and the most similar word to it holds a similarity value of less than 0.3, and therefore 0 is assigned. Other column cells are left empty as there values are not required in demonstrating the similarity computation process. The semantic vector $š$ is obtained by selecting the largest value in each column. The resulting values are multiplied by the weight of the corresponding word in T, to account for the significance of the term. As a result, the semantic vectors for T_1, and similarly, T_2, are:

$v_{sem}(T_1)$ = {0.5452 0.6166 0.7859 0.279 0.2598 0.5664 0 0.1 0.3346 0.6001 0 0.3}
$v_{sem}(T_2)$ = {0.3519 0.6166 0 0.279 0.2598 0.2865 0.2693 0.1967 0.5184 0.6001 0 0.593}

Table 8. Process for deriving the syntactic vectors

Syntactic features	T_1	T_2
Function word	1	2
Noun	2	3
Verb	1	3
Adjective	0	0
Adverb	1	0
Digit	0	0
Hashtag	1	0
Mention	0	0
Interrogation	0	1
Exclamation	1	0
Currency	0	0
Ratio	0	0

From $v_{sem}(T_1)$ and $v_{sem}(T_2)$, the semantic similarity between the two tweets is $S_{sem} = 0.781$.

The syntactic vectors $v_{syn}(T_1)$ and $v_{syn}(T_2)$ are derived from the syntactical features that correspond to each tweet. The process of deriving the syntactic vectors, $v_{syn}(T_1)$ and $v_{syn}(T_2)$, as per the feature set shown in table 1, is shown in Table 8. Unlike semantic vectors, these are count-based vectors that record the number of occurrences for the different morphological structures and syntactical features in a tweet.

$v_{syn}(T_1) = \{1\ 2\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\}$
$v_{syn}(T_2) = \{2\ 3\ 3\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\}$

and, thus, $S_{syn} = 0.7646$.

Finally, the similarity between tweets *"Sterling falls substantially on #Brexit concerns!"* and *"Is the pound falling on renewed Brexit worries?"* is 0.78, using 0.8 for δ[2].
Although T_1 and T_2 do only share words *on* and *Brexit*, the algorithm is still aware of the similarity between the tweet pair. Traditional BOW methods would result in a similarity of 0.2887, which is very low similarity measure, while the proposed measure computes a relatively high similarity. Thus, this example demonstrates that the proposed method can capture the meaning of the tweet regardless of the common words.

5 Experimental Methodology and Results

Although multiple benchmark datasets have been published for evaluating short-text similarity measures, there are not much benchmarks produced on raw tweets. SemEval-2015 shared paraphrase and semantic similarity task published a test dataset of similarity labelled tweet pairs. However, this dataset is considered irrelevant for evaluating the proposed measure due to four reasons: 1) the data is developed for the task of paraphrase identification, and each pair consists of a tweet and a synthetic tweet to determine whether they imply the same meaning, thus, are considered paraphrases. 2) The tweets are provided in a pre-processed format and thus important structural and syntactical information are missing. 3) The frequency distribution of the data exhibits a strong bias, with 59% of the data falling in the lower quarters of the similarity range and only 17% of the data falling in the upper range. 4) The benchmark is produced using Amazon Mechanical Turks (AMT) [62] instead of human experts. It is worth noting that the maximum correlation achieved among all participating groups for this task on the full test data is 0.61. Instead, SemEval-2014 STS-750 tweet-news benchmark is used with Google News pre-trained embedding model for evaluation. In addition, a preliminary dataset of 30 raw tweet pairs was constructed with human similarity scores provided using 32 participants as described in Section 4.2.3, and used with the Political tweets pre-trained model described in section 4.3.2. Both datasets are used to evaluate the performance of the proposed similarity algorithm, which requires two parameters to be determined at the outset:

1. A threshold for deriving the semantic vector.

[2] Empirically derived value through experiments on tweet pairs.

2. A weighting factor, δ, for determining the significance between semantic information and syntactic information.

The parameters in the following experiments where empirically found using the benchmark datasets, evidence and methodology of previous publications [12, 45] and intuitive consideration as follows: since syntax plays a relatively small role for semantic processing of text, the semantic computation is weighted higher, 0.8 for δ. With regard to the semantic vector threshold, it has been determined considering two aspects: 1) detecting and utilizing similar words semantic characteristics to the greatest extent, and 2) keeping the noise low. This implies the use of a small semantic threshold, but not too small. A small threshold allows the model to capture sufficient sematic information of words distributed vector representations obtained by the neural embedding model. However, as the word embedding model represents word co-occurrence relationships, a too small threshold will introduce excessive noise to the model causing a deterioration of the overall performance. Based on these considerations, different parameter values were experimentally observed and the appropriate values were identified using the tweets pairs' benchmark datasets. In this way, we empirically found 0.3 for semantic vector threshold works well for both word embedding pre-trained models (Section 3.1.3). Similarly, 0.8 for δ works well for weighting the contribution of semantic and syntactic information to the overall similarity in both Twitter-based datasets used in this research and thus, both thresholds should be extended to different application domains in microblogging OSN.

5.2 Experiment with Human Similarities of Tweet Pairs

The referenced benchmark datasets use an adapted 6-point Likert scale for similarity ratings (Section 4.2.3.2, table 4). The production of the political tweets benchmark involved asking participants to complete a questionnaire, rating the similarity of meaning of the tweet pairs on the scale from 0.0 (minimum similarity) to 5.0 (maximum similarity), as in Charles [57] and Agirre, Diab [47]. Tweets are listed according to their corresponding cluster centers as discussed in Section 4.2.3 to make up tweet pairs. These pairs are listed in a randomized order within each cluster. The two tweets making up each pair are the cluster representative tweet and the randomly selected tweet to prevent introducing any bias to the benchmark data (Section 4.2.3.1, table 3). The participants were asked to complete the similarity annotation task in their own time and to work through from start to end according to the given instructions. As discussed in Section 4.2.3.2, these instructions contain linguistic anchors for the 6 main scale points 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, adapted using [47, 57]. The use of these anchors allows the application of similarity statistical measurements as they yield psychometric properties analogous to an interval scale [57]. Each of the 30 tweet pairs was assigned a semantic similarity score calculated as the mean of the judgments obtained by the participants. These can be seen in Table 9, where all human similarity scores are provided as the mean score for each pair.

Table 9. Political tweets dataset results

Pair Id	Tweet Pair	Human Similarity (Mean)	Algorithm Similarity Measure
1	Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today Brussels attacks may sway Brexit vote: Strategists	3.6	3.71
2	Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today On one hand, there are decent human beings that send their sympathies to the Brussels victims and their families. And then there's Brexit.	3.85	3.78
3	Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today #Brussels attacks: Terrorism could break the EU and lead to Brexit	3.53	3.62
4	Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today Terrorism is the scariest think. And it's ways more scarier if it's in the EU, in your home. Stay strong Brussels! #prayersforBrussels	3.51	3.67

5	Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today Brussels Attacks Spur Brexit Campaign: Anti-Immigration Parties Link Terror To EU Open Borders	2.83	3.73
6	Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today The world is seriously fucked up right now.	0.45	2.73
7	EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats @caddenlimos connecting low paid workers doing non skilled jobs from Poland with terrorism in Belgium	1.93	2.54
8	EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats Has anyone from the Brexit Brigade addressed the issue of what will happen to existing EU citizens living and working in the UK	3.54	3.43
9	EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats @thebobevans Today's atrocity foreseable under EU policy. Trust UK security services to protect UK citizens. Brexit	0.53	2.28
10	EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats #Brexit supporters claim EU needs UK more than we need it. 45% of UK exports go to EU, 10% of EU exports come here	0.49	2.39
11	EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats Could 2m+ 18-34 Year Old Workers Emigrating After a Brexit Cause a Recruitment Nightmare?	2	2.46
12	EU Referendum Briefing on Living and Working in the UK #ProtectJobs #Expats We must stay in #EU to protect jobs	3.52	2.99
13	Sterling slides on renewed Brexit worries Sterling slides on renewed Brexit worries - A decline of 1% marks the 25th day this year the pound has moved	4.77	4.44
14	Sterling slides on renewed Brexit worries London-based crowdfunding platform Seedrs poll on the EU referendum finds 47% of investors and 43% of entrepreneurs	0.83	2.79
15	Sterling slides on renewed Brexit worries Brussels bombing rose Brexit risk. How to trade Pound in case of Brexit: GBP $USD #FX	2.63	3.59
16	Sterling slides on renewed Brexit worries Sterling has dipped cause markets believe Brexit will happen-GOOD-spivs in the city will adjust after playing their gambling games	3.94	3.52
17	Sterling slides on renewed Brexit worries In most scenarios #Brexit will impose a significant long-term cost on the UK economy #OEBrexit	2.27	2.63
18	Sterling slides on renewed Brexit worries it's not just an economic argument	0.7	1.56
19	#Brexit Emerges As Threat To TTIP Deal Jobs rely on trade NOT a political union. Security relies on sharing information NOT a political union. #Brexit #StrongerIn #VoteLeave	0.99	2.84
20	#Brexit Emerges As Threat To TTIP Deal #Brexit, a new threat to TTIP transatlantic trade talks	4.92	3.98
21	#Brexit Emerges As Threat To TTIP Deal Naive to think Brexit would solve the #ttip problem...only way to protect #NHS is for govt to exclude it from TTIP	3.32	3.8
22	#Brexit Emerges As Threat To TTIP Deal Benign Brexit would require accepting high levels of immigration and deep trade agreement with EU	1.96	3.15
23	#Brexit Emerges As Threat To TTIP Deal Brexit Risks Rising	0.9	2.55
24	#Brexit Emerges As Threat To TTIP Deal Negotiating trade agreements after #Brexit would be complicated for UK as there's no @wto for #services: @angusarmstrong8 at @FedTrust event	2.93	3.31
25	It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union | via @Telegraph UK's NHS will NOT survive staying in the EU	4.74	4.45
26	It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union | via @Telegraph What would #Brexit mean for the #pharma industry?	0.93	2.97

27	It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union ! via @Telegraph To the "expats" in spain who are moaning about immigration can i just say this to you? Jog the fuck on you UTTER hypocrites	0.3	3.31
28	It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union ! via @Telegraph How can we save NHS inside EU	3.67	3.9
29	It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union ! via @Telegraph We send £350 million to Brussels every week - enough to build a new NHS hospital every week. Let's #VoteLeave and #TakeControl	3.05	3.15
30	It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union ! via @Telegraph The EU referendum is a vote for the EU or the NHS, we can't have both	3.91	4.06

5.2.3 Experiment Analysis and Discussion

The proposed algorithm's similarity measure demonstrated a good Pearson correlation coefficient compared to human judgments with relatively good inter-rater agreement, significant at the 0.01 level. A correlation of 0.825 and 0.776 was achieved on EU_Referendum and STS.tweet_news benchmarks respectively. The average performance of TREASURE is 0.8, which is the best correlation among state-of-the-art measure for tweet similarity. The test dataset of SemEval-2015 semantic similarity shared task is irrelevant for evaluation due to the reasons discussed in Section 5 and reproducing the systems that participated in this task is labor and time intensive process. Nevertheless, the maximum achieved correlation on this task is 0.61, which is significantly lower than TREASURE's performance. The inconsistent experiment constraints between SemEval-2015 shared task and TREASURE in order to make such judgment is worth acknowledgement, however, both measures aim to compute semantic similarity for tweets. TREASURE is compared against different levels of textual semantic similarity computation; concepts-based measures [63-69], formal English sentences measure (STASIS) [12], the top performing measure on SemEval-2014 semantic similarity task (DLS@CU) on informal microblogging posts (i.e. tweets) [70] in order to provide a thorough insight on the performance of TREASURE. Table 10 and Figure 3 show the correlation coefficient, mean, and standard deviation for the nine measures on two benchmark datasets against TREASURE.

Table 10. Correlations achieved by different semantic similarity measures

	PATH	WUP	RES	JCN	LCH	LIN	STASIS	DLS@CU	WPATH	TREASURE
STS.tweet_news	0.740	0.54	0.313	0.75	0.319	0.656	0.683	0.764	0.699	0.775
EU_Referendum	0.653	0.579	0.004	0.636	0.087	0.589	0.744	-	0.605	0.825
Mean	0.697	0.56	0.159	0.693	0.203	0.623	0.714	-	0.652	0.8
Standard Deviation	0.062	0.028	0.218	0.081	0.164	0.047	0.043	-	0.066	0.035

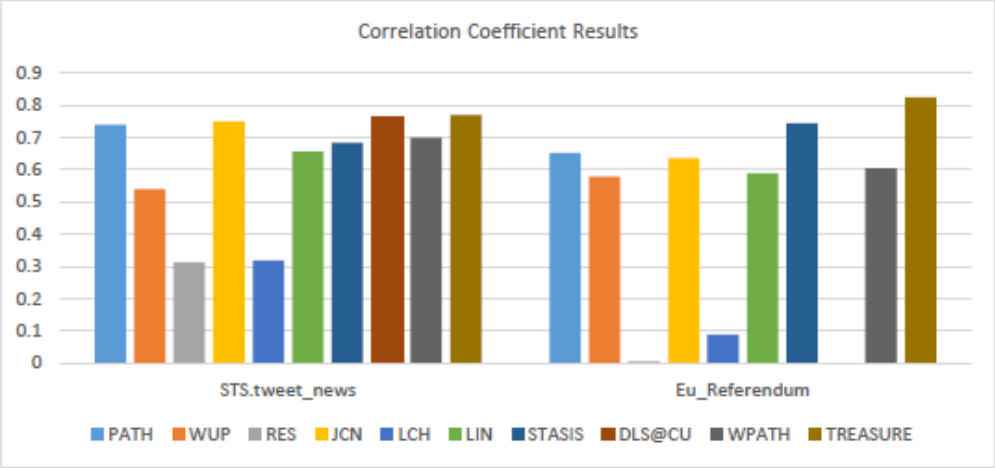


Figure 3. Correlations achieved by different semantic similarity measure

TREASURE achieved the best correlation compared to the other measures for both benchmarks used. Using a uniform experiment settings and constant threshold parameter values, it can be observed that TREASURE performed better on the EU_Referendum benchmark than the STS.tweet_news dataset. This can be attributed to three reasons:

1. **Characteristics of the test dataset** –the architecture of the developed algorithm is composed of semantic-based modules and syntactic-based modules. The latter is designed to extract syntactic features from raw tweets while the former generates semantic feature vectors upon performing certain steps of preprocessing. All tweet pairs in the Eu_Referendum political dataset retain Twitter-based user conventions and share relatively similar level of noise. This means that the syntactical feature vector is not biased with data in one tweet that make up a pair. This is not the case in the STS.tweet_news dataset, where each pair is formed of a typical tweet, which may contain hashtags and special symbols, and a corresponding news headline that is a typical sentence composed of formal English text. The lack of uniformity of the tweet pairs in the STS.tweet_news dataset results in a performance deterioration of the syntactical similarity computation module, which consequently causes the accuracy of the overall similarity score to degrade.

2. **Word embedding pre-trained model** –the core of the semantic processing is the word analogy module, which calculates the semantic relationships between words. This module computes the semantic relationship between word vectors generated by a neural embedding model. The effectiveness of this model depends on two factors: 1) *quality* (positive examples such as "cloudy sky" are more informative than negative examples such as "cloudy book") and 2) *quantity* (i.e. vocabulary coverage) of the learning text corpus. The Google News pre-trained model was used in the evaluation of the similarity algorithm on STS.tweet_news, whereas the political pre-trained model was used in the evaluation of the measure on EU_Referendum dataset. While the former model features a higher vocabulary coverage from a large corpus of Google news, it misses on some of the OOV words such as hashtags and slangs (e.g. *uhhhh, yummie, hmmm, WTF, damn, aww, ouch,* etc.) event-specific vocabulary occurring in incredible velocity in tweets. This is due to the fact that the training corpora contain news articles, which are generally written in a formal structured language, in which words can be mapped to English dictionaries. Thus, the model learns distributed representations for words used in such documentation and misses out of vocabulary (OOV) words that are commonly used in tweets due to the character length restriction. Therefore, although the model exhibits a large set of examples and vocabulary size, it does not provide a vectorized modelling for OOV words. This means that an embedding model, which is learned from tweets data is required in order to

cater for the informal language used in social media contexts [71]. Therefore, the evaluation of the developed measure with reference to the EU_Referendum benchmark was performed using a word embedding model that was pre-trained with a corpus of political tweets instead of the Google News model. The results shown in table 10 demonstrate that, under the given experimental setting, a correlation enhancement of 5% when a Twitter-based neural embedding model is used to predict the semantic equivalence between tweets, rather than using a model trained on general data.

3. **Production of the gold standard labels** – similarity judgments are highly subjective between humans and are linked to psychological and mental behaviors. Thus, in order to perform statistical tests and derive accurate conclusions on a measure that predicts human typical cognitive system, it is imperative to compare it against a benchmark produced by human experts with a good level of inter-judge agreement. The first benchmark test data was assembled using AMT crowdsourcing, gathering 5 scores per sentence pair. The similarity label score is represented as the mean of those five scores. It is worth noting that five annotators is relatively low number of raters in order to generate a reliable benchmark [72]. This can be observed through example pairs where the similarity prediction measure produces a score that is intuitively more logical than the gold standard. For example, the pair *This is interesting: "What We Don't Know Is Killing Us"* and *Editorial: What We Don't Know Is Killing Us* is assigned a similarity score of 3.6, while the measure predicted score is 4.85. Such cases contribute to the decrease of correlation even though the measure intuitively seems to perfume better than the gold standard. The non-logical labelled similarities observed can be attributed to a benchmark reliability problem of low inter-judge agreement. In contrast, the Eu_Referendumn benchmark was produced by 32 human observers who share a certain set of characteristics (nativeness, age, and education level). The generated benchmark features a good degree of reliability, at α = 0.8. That is, the similarity measure can be statistically evaluated against relatively uniform human psychometric properties that can be reproducible using other set of observers.

Table 10 shows that TREASURE achieves the best correlation among other measures in predicting the semantic similarity of tweets. For the SemEval-2014 semantic similarity shared task, the algorithm developed in [70] achieved the best correlation coefficient on STS.tweet_news test data among 38 other participating systems, at r = 0.764. The comparison of our tweet similarity computation algorithm with the top scoring competitor shows that TREASURE performed better when tested on the same dataset, at r = 0.775. Compared to STASIS, TREASURE achieved 9.2% better correlation on STS.tweet_news and 8.1% on EU_Referendum test data. Comparing with concept similarity algorithms, JCN provides the closest performance to our measure, at r = 0.75 while RES recorded the least correlation for STS.tweet_news, r = 0.313. For EU_Referendum, Path comes after STASIS with 17.2% less correlation compared to our algorithm. Again, RES's results demonstrate a non-significant correlation for this test data, at r = 0.004. The average of the measures correlation coefficient indicates that TREASURE outperforms the three type of measures under comparison; concept-based, formal, and informal short-text similarity measurements for two Twitter-based benchmark test data. STASIS (which uses WordNet) achieved a very good correlation when evaluated on sentences composed with dictionary word definitions and DLS@CU (uses PPDB [73]) performed as well on image descriptions, at r = 0.816 and r = 0.821 respectively. However, their performance has deteriorated when applied to the context of social data. It can be observed from the analysis results that such measures, which are based on lexical taxonomies achieved less correlation to human judgements when used for informal short text analysis. This is mainly attributed to the high proportion of OOV words present in microblogging posts. These words are more prevalent in the EU_Referendum test data, which is the reason behind the decrease in the correlations obtained by evaluation on this benchmark. TREASURE, unlike these algorithms, obtains its semantic calculations by learning word distributed representations from co-occurrences in large corpora of formal and informal text. This way, it is able to derive semantic relationships for the

nature of modern language used in social media user generated context, which is absent in traditional English knowledge bases such as WordNet.

6 Conclusion and Future Work

This paper presented TREASURE, a novel statistical semantic approach for measuring the semantic similarity between microblog posts, particularly tweets, based on hybrid semantic and syntactic feature set. The novelty of TREASURE builds upon the success of distributed word representation obtained by training a word embedding model on relevant corpora. The resulting pre-trained model represent corpus vocabulary as dense word vectors in a high dimensional space. Thus, our tweet similarity novel approach not only captures common human knowledge, but it is also able to adapt to an application areas using different domain specific corpora. The computation of the semantic similarity between two tweets commences by deriving semantic similarity between words from the pre-trained word embedding model. The whole corpus of tweets under consideration is assembled to construct a domain specific corpus, from which statistical calculations are performed to derive the information content carried by words. The proposed measure not only considers the semantic information contained in a tweet, but also captures the syntactic features that contribute to the overall meaning. The overall tweet similarity is then defined as a combination of semantic similarity and syntactic similarity. Considering the view that syntactic information play a subordinate role for interpreting tweet meaning, it has been weighted less in defining the overall tweet similarity. The evaluation methodology was conducted using two Twitter-based benchmark datasets. The first one consists of 750-pairs on tweet_news data that is labelled with similarity judgements. The second benchmark was produced using 30-pairs on EU_Referendum tweets and rated by 32 human participants with a good level of inter-judge agreement. The average of raters' judgments was considered the gold standard for comparison with our measure's estimated similarity scores. The achieved correlation was compared to the state-of-art tweet similarity measure, a sentence similarity measure, and concept measures. The experimental results on both test datasets illustrates that the proposed measure achieves a similarity scores that are consistent with human typical cognitive knowledge, significant at the 0.01 level.

Tweets have always been restricted to 140 characters and users tend to squeeze as much information as possible in a single tweet, which impose lots of noise. This limit has recently increased to 280 characters providing more room for context. Further work will investigate the consequences of this increase on the performance of the proposed measure and how the defined parameters can be adapted accordingly. In the meanwhile, there is no published benchmark dataset that includes tweets with more than 140 characters. Therefore, future work will involve the following stages:

1. Building a reliable benchmark dataset for tweets containing larger context.
2. Evaluation and analysis of the proposed tweet similarity computation approach.
3. Optimization of the architecture's semantic and syntactic components in order to capture a representative set of features for the expanded tweets.
4. Adaptation of the predefined threshold parameters and recursively repeat processes 2-4 until the highest correlation with reference to the benchmark is achieved.

References

1. Jungherr, A., *Twitter use in election campaigns: A systematic literature review*. Journal of information technology & politics, 2016. 13(1): p. 72-91.
2. Kim, J., et al. *Leveraging the crowd to detect and reduce the spread of fake news and misinformation*. in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 2018. ACM.
3. Xu, W., C. Callison-Burch, and B. Dolan. *SemEval-2015 Task 1: Paraphrase and semantic similarity in Twitter (PIT)*. in *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*. 2015.
4. Zhu, G. and C.A. Iglesias, *Exploiting semantic similarity for named entity disambiguation in knowledge graphs*. Expert Systems with Applications, 2018. 101: p. 8-24.
5. Alnajran, N., et al. *Cluster Analysis of Twitter Data: A Review of Algorithms*. in *Proceedings of the 9th International Conference on Agents and Artificial Intelligence*. 2017. Science and Technology Publications (SCITEPRESS)/Springer Books.
6. Lippi, M. and P. Torroni, *Argumentation mining: State of the art and emerging trends*. ACM Transactions on Internet Technology (TOIT), 2016. 16(2): p. 10.

7. Lin, Y.-S., J.-Y. Jiang, and S.-J. Lee, *A similarity measure for text classification and clustering*. IEEE transactions on knowledge and data engineering, 2014. **26**(7): p. 1575-1590.
8. Zou, X., J. Yang, and J. Zhang, *Microblog sentiment analysis using social and topic context*. PloS one, 2018. 13(2): p. e0191163.
9. Jehl, L., F. Hieber, and S. Riezler. *Twitter translation using translation-based cross-lingual retrieval*. in *Proceedings of the seventh workshop on statistical machine translation*. 2012. Association for Computational Linguistics.
10. Miller, G.A., et al., *Introduction to WordNet: An on-line lexical database*. International journal of lexicography, 1990. 3(4): p. 235-244.
11. Pawar, A. and V. Mago, *Calculating the similarity between words and sentences using a lexical database and corpus statistics*. arXiv preprint arXiv:1802.05667, 2018.
12. Li, Y., et al., *Sentence similarity based on semantic nets and corpus statistics*. IEEE transactions on knowledge and data engineering, 2006. **18**(8): p. 1138-1150.
13. Soğancıoğlu, G., H. Öztürk, and A. Özgür, *BIOSSES: a semantic sentence similarity estimation system for the biomedical domain*. Bioinformatics, 2017. 33(14): p. i49-i58.
14. Barry, C., et al., *Text Information Retrieval Systems*. 2007: Academic Press.
15. Okazaki, N., et al., *Sentence extraction by spreading activation through sentence similarity*. IEICE TRANSACTIONS on Information and Systems, 2003. **86**(9): p. 1686-1694.
16. Tian, Y., et al. *Measuring the similarity of short texts by word similarity and tree kernels*. in *Information Computing and Telecommunications (YC-ICT), 2010 IEEE Youth Conference on*. 2010. IEEE.
17. Rudrapal, D., A. Das, and B. Bhattacharya. *Measuring Semantic Similarity for Bengali Tweets Using WordNet*. in *Proceedings of the International Conference Recent Advances in Natural Language Processing*. 2015.
18. Das, D. and S. Bandyopadhyay. *Developing Bengali WordNet Affect for Analyzing Emotion*. in *International Conference on the Computer Processing of Oriental Languages*. 2010.
19. Salton, G. and C. Buckley, *Term weighting approaches in automatic text retrieval*. 1987, Cornell University.
20. Allan, J., C. Wade, and A. Bolivar. *Retrieval and novelty detection at the sentence level*. in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. 2003. ACM.
21. Akkaya, C., J. Wiebe, and R. Mihalcea. *Subjectivity word sense disambiguation*. in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. 2009. Association for Computational Linguistics.
22. Atoum, I., A. Otoom, and N. Kulathuramaiyer, *A comprehensive comparative study of word and sentence similarity measures*. arXiv preprint arXiv:1610.04533, 2016.
23. Foltz, P.W., W. Kintsch, and T.K. Landauer, *The measurement of textual coherence with latent semantic analysis*. Discourse processes, 1998. **25**(2-3): p. 285-307.
24. Landauer, T.K., et al. *How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans*. in *Proceedings of the 19th annual meeting of the Cognitive Science Society*. 1997.
25. Dennis, S., et al. *Introduction to latent semantic analysis*. in *Slides from the tutorial given at the 25th Annual Meeting of the Cognitive Science Society, Boston*. 2003.
26. Blei, D.M., A.Y. Ng, and M.I. Jordan, *Latent dirichlet allocation*. Journal of machine Learning research, 2003. 3(Jan): p. 993-1022.
27. Burgess, C., K. Livesay, and K. Lund, *Explorations in context space: Words, sentences, discourse*. Discourse Processes, 1998. **25**(2-3): p. 211-257.
28. Landauer, T.K., D. Laham, and P.W. Foltz. *Learning human-like knowledge by singular value decomposition: A progress report*. in *Advances in neural information processing systems*. 1998.
29. Hong, L. and B.D. Davison. *Empirical study of topic modeling in twitter*. in *Proceedings of the first workshop on social media analytics*. 2010. ACM.
30. Mehrotra, R., et al. *Improving lda topic models for microblogs via tweet pooling and automatic labeling*. in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 2013. ACM.
31. Collobert, R. and J. Weston. *A unified architecture for natural language processing: Deep neural networks with multitask learning*. in *Proceedings of the 25th international conference on Machine learning*. 2008. ACM.
32. Mikolov, T., et al. *Distributed representations of words and phrases and their compositionality*. in *Advances in neural information processing systems*. 2013.
33. Beam, A.L., et al., *Clinical Concept Embeddings Learned from Massive Sources of Medical Data*. arXiv preprint arXiv:1804.01486, 2018.
34. Mikolov, T., et al., *Efficient estimation of word representations in vector space*. arXiv preprint arXiv:1301.3781, 2013.
35. Pennington, J., R. Socher, and C. Manning. *Glove: Global vectors for word representation*. in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.
36. Naili, M., A.H. Chaibi, and H.H.B. Ghezala, *Comparative study of word embedding methods in topic segmentation*. Procedia Computer Science, 2017. **112**: p. 340-349.
37. De Boom, C., et al. *Learning semantic similarity for very short texts*. in *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*. 2015. IEEE.
38. Dey, K., et al., *Emtagger: a word embedding based novel method for hashtag recommendation on twitter*. arXiv preprint arXiv:1712.01562, 2017.

39. Alnajran, N., et al., *An Empirical Performance Evaluation of Semantic-Based Similarity Measures in Microblogging Social Media*, in *In Proceedings of the Fifth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT 2018)*. 2018, IEEE/ACM: Zurich.
40. Budanitsky, A. and G. Hirst. *Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures*. in *Workshop on WordNet and other lexical resources*. 2001.
41. Hale, M.M., *A comparison of WordNet and Roget's taxonomy for measuring semantic similarity*. arXiv preprint cmp-lg/9809003, 1998.
42. Bojanowski, P., et al., *Enriching word vectors with subword information*. arXiv preprint arXiv:1607.04606, 2016.
43. Alnajran, N., et al., *A Word Embedding Model Learned from Political Tweets*, in *In Computer Engineering & Systems (ICCES), 2018 13th International Conference on*. 2018, IEEE.
44. Aggarwal, C.C. and C. Zhai, *Mining text data*. 2012: Springer Science & Business Media.
45. Wiemer-Hastings, P. *Adding syntactic information to LSA*. in *Proceedings of the Annual Meeting of the Cognitive Science Society*. 2000.
46. Guo, W., et al. *Linking tweets to news: A framework to enrich short text data in social media*. in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2013.
47. Agirre, E., et al. *Semeval-2012 task 6: A pilot on semantic textual similarity*. in *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. 2012. Association for Computational Linguistics.
48. Kumar, S., F. Morstatter, and H. Liu, *Twitter data analytics*. 2014: Springer.
49. Banker, K., *MongoDB in action*. 2011: Manning Publications Co.
50. Davidson, S., *Wordnik*. The Charleston Advisor, 2013. **15**(2): p. 54-58.
51. Alnajran, N., et al. *A Heuristic Based Pre-processing Methodology for Short Text Similarity Measures in Microblogs*. in *High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2018 IEEE 20th International Conference on*. 2018. IEEE.
52. Ifrim, G., B. Shi, and I. Brigadir. *Event detection in twitter using aggressive filtering and hierarchical tweet clustering*. in *Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014*. 2014. ACM.
53. Schütze, H., C.D. Manning, and P. Raghavan, *Introduction to information retrieval*. Vol. 39. 2008: Cambridge University Press.
54. Severino, R., *Getting Your Random Sample in Proc SQL*.
55. Miller, G.A., *The magical number seven, plus or minus two: Some limits on our capacity for processing information*. Psychological review, 1956. **63**(2): p. 81.
56. O'Shea, J., et al., *Benchmarking short text semantic similarity*. International Journal of Intelligent Information and Database Systems, 2010. **4**(2): p. 103-120.
57. Charles, W.G., *Contextual correlates of meaning*. Applied Psycholinguistics, 2000. **21**(4): p. 505-524.
58. Klaus, K., *Content analysis: An introduction to its methodology*. 1980, Sage Publications.
59. Hayes, A.F. and K. Krippendorff, *Answering the call for a standard reliability measure for coding data*. Communication methods and measures, 2007. **1**(1): p. 77-89.
60. Hayes, A.F., *Statistical methods for communication science*. 2009: Routledge.
61. De Swert, K., *Calculating inter-coder reliability in media content analysis using Krippendorff's Alpha*. Center for Politics and Communication, 2012: p. 1-15.
62. Buhrmester, M., T. Kwang, and S.D. Gosling, *Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data?* Perspectives on psychological science, 2011. **6**(1): p. 3-5.
63. Rada, R., et al., *Development and application of a metric on semantic nets*. IEEE transactions on systems, man, and cybernetics, 1989. **19**(1): p. 17-30.
64. Wu, Z. and M. Palmer. *Verbs semantics and lexical selection*. in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. 1994. Association for Computational Linguistics.
65. Lin, D. *An information-theoretic definition of similarity*. in *Icml*. 1998. Citeseer.
66. Leacock, C. and M. Chodorow, *Combining local context and WordNet similarity for word sense identification*. WordNet: An electronic lexical database, 1998. **49**(2): p. 265-283.
67. Jiang, J.J. and D.W. Conrath, *Semantic similarity based on corpus statistics and lexical taxonomy*. arXiv preprint cmp-lg/9709008, 1997.
68. Zhu, G. and C.A. Iglesias, *Computing semantic similarity of concepts in knowledge graphs*. IEEE Transactions on Knowledge and Data Engineering, 2017. **29**(1): p. 72-85.
69. Resnik, P., *Using information content to evaluate semantic similarity in a taxonomy*. arXiv preprint cmp-lg/9511007, 1995.
70. Sultan, M.A., S. Bethard, and T. Sumner. *DLS $@ $ CU: Sentence Similarity from Word Alignment*. in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. 2014.
71. Li, Q., et al., *Data sets: Word embeddings learned from tweets and general data*. arXiv preprint arXiv:1708.03994, 2017.
72. O'shea, J., Z. Bandar, and K. Crockett, *A new benchmark dataset with production methodology for short text semantic similarity algorithms*. ACM Transactions on Speech and Language Processing (TSLP), 2013. **10**(4): p. 19.
73. Ganitkevitch, J., B. Van Durme, and C. Callison-Burch. *PPDB: The paraphrase database*. in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013.

A Semantic Based Cluster Analysis Algorithm for Detecting Semantic Themes within Microblogging Posts

Noufa Alnajran, Keeley Crockett, *Senior Member, IEEE*, David McLean, and Annabel Latham, *Member, IEEE*

Abstract—Unsupervised machine learning has been a problem of intense discussion due to its potential in knowledge extraction for various applications and domains. Much research have been conducted to tackle this problem for Information Retrieval (IR) systems by clustering context-rich documents. The problem is more complex in microblogging Online Social Networks (OSN), where users generate highly unstructured content, such as tweets, which are short text posts often composed of informal English language. Due to the special characteristics of these tweets, traditional cluster analysis algorithms may not produce optimal results. Little research has been undertaken towards clustering Twitter posts however; these methods feature one or more of three weaknesses. 1) Require the number of clusters to be determined beforehand, 2) perform keyword-based clustering, which ignores the semantic relations between tweets, and 3) Model the text in a high dimensional vector space model (VSM) and use Euclidean Distance to calculate similar documents. In this paper, we propose a novel approach towards the problem of semantic cluster analysis for microblogging posts. Unlike existing research, our new algorithm performs recursive iterations over the dataset to produce the optimal number of clusters with semantic themes. The proposed approach tackles the problem from a natural language processing (NLP) perspective, and uses TREASURE as the distance measure to computer the semantic similarities between tweets. The evaluation experiment was conducted with reference to a reliable benchmark produced by human raters on a political dataset of tweets. Results show that the peoposed algorithm outperforms k-means baseline and achieved significant accuracy compared to human jugdements.

Index Terms— Cluster Analysis, Unsupervised Learning, Semantic Textual Analysis, Microblogging, Social Network Analysis, Twitte

1 Introduction

MICROBLOGGING social networks has rapidly gained interest among different societies. Twitter, in particular, provides an informal platform where people can easily publish posts and broadcast messages on various spacial and temporal events. Its role in spreading real-time awareness during natural disasters such as Hurricane Sandy and socio-political events such as the Arab Spring [1] has made it a significant source of information for both businesses and decision makers.

Several studies have aimed at analyzing Twitter posts through machine learning techniques such as supervised and unsupervised learning [2]. However, classification techniques could be considered to have limited capabilities due to: 1) the unpredictable nature of the dataset and 2) the massive amount of data required for training, which is too large to produce manual labelling. The exponential amount of user-generated content on this site is too vast for manual analysis. More than 500 million short-text messages, referred to as "tweets", are published every day [3]. This requires an automated and scalable mining process to discover patterns in the unstructured data.Unsupervised applications of cluster analysis have been reported to be particularly suitable for the user generated content in microblogging [4]. This can be attributed to the nature of the data, which implies the existence of unforeseen groups. These groups may carry important nuggets of information, which can only be revealed by unsupervised learning algorithms.

Among the research conducted around clustering microblogging posts, researchers aim to analyse social behaviours find relevant information and achieve different tasks, such as tailoring advertisements for groups with similar interests [5], event detection [6], trending issues extraction [7], and prediction of micro-populations [8]. However, several natural challenges of the data prevent standard clustering algorithms being applied with their full potentials: 1) Sparseness –unlike traditional clustering of documents, which are rich in context, tweets are restricted to 140 characters. 2) Non-standardization – people invented many ways to expand the semantics that are carried out by the tweet. This implies the usage of slangs, misspelled, and connected words. Users also use self-defined hashtags to identify topics or events. 3) Volume –the rapid generatio

Due to the textual length restriction of microblogging posts, the content in tweets is limited; however, they may still contain rich meanings. Therefore, tweets require intelligent techniques, such as incorporating semantic technologies that can analyse datasets with such complex characteristics and convey meanings and correlations.

1.1 Problem Statement

The ability to identify fine-grained granularities in high volum text corpora has proven useful for a wide variety of machine learning applications. Previously, many researchers have investigated ways of automatically detecting themes in a collection of text documents. Nevertheless, most previous work focused on traditional documents containing grammatical text.

• *N. Alnajran, K. Crockett, D. McLean, and A. Latham are with the School of Computing, Mathematics, and Digital Technologyology, Manchester Metropolitan University, Manchester, UK. E-mail: noufa.alnajran@stu.mmu.ac.uk, {k.crockett, [illegible], a.latham}@mmu.ac.uk.*

The state-of-the-art literature review shows that there are few recent studies, which have highlighted the potential and importance of developing semantic-based clustering algorithms specifically for microblogging posts, i.e. tweets. Alnajran, Crockett [9] provides a comprehensive review on applications of unsupervised learning for microblogging posts. They also indicated that the very informal language, especially the high degree of lexical variation, used in social media has posed serious challenges. Due to these challenges, they concluded that the current Twitter-based clustering approaches feature several weaknesses. 1) Require the number of clusters to be determined beforehand, 2) perform keyword-based clustering, which ignores the semantic relations between tweets, and 3) Model the text in a high dimensional vector space model (VSM) and use Euclidean Distance to calculate similar documents.

1.2 Contributions and Outline

The main purpose of this paper is to tackle the problem of clustering microblogging posts into semantic themes to discover hidden patterns. Towards achieving this aim, this paper proposes a novel approach towards the problem of semantic cluster analysis for microblogging posts. Unlike existing research, our new algorithm performs recursive iterations over the dataset to produce the optimal number of clusters with semantic themes. Thus, The number of clusters, *k*, is not required to be determined a priori. The weaknesses of previous research discussed in Section 1.1 are tackled from a natural language processing (NLP) perspective, using TREASURE [10]–tweet similarity measure for computing the semantic distance between tweets.

The contributions of this paper are as follows:

- Designing a novel unsupervised algorithm for discovering semantic themes in microblogging posts that can be adapted to a wide range of informal short text applications.
- Designing an experimental methodology for producing a benchmark dataset with a good level of inter-judge agreement for a subjective evaluation of the proposed algorithm.
- Evaluation of the proposed cluster analysis algorithm with reference to human judgements.

In this paper, the authors present relevant literature on existing methods for unsupervised ML application for microblogging posts (2). The authors present the data collection and preprocessing methodology (3) in which a large volume of microblogging posts have been collected (3.1) and preprocessed (3.2). The authors present the proposed cluster analysis approach (4) including the distance measure (4.1) and algorithm (4.2). In (5), the authors present the experiment methodology, design (5.1), benchmarks (5.2), and evaluation metrics (5.3). The authors present results from validating the microblogging cluster analyser (6), conclusions and future work (7).

2 Relater Research

In this section, the author reviews literature to highlight and discuss the advantages and disadvantages of the previous technical approaches taken to cluster microblogging posts, particularly tweets, by analysis of textual features.

In Section 1.1, the authors discussed that determining the number of clusters, *k*, a priori affects the final clustering quality. For example, Li, Ye [11] propose a manual selected based dynamic incremental clustering algorithm for clustering microblogging posts using *k*-means. Unlike traditional *k*-means, which randomly selects the initial cluster seeds, their proposed algorithm involve manually selecting these initial centers. The author of this paper argues that this semi-supervised approach is inefficient for the large-scale microblog posts and may introduce bias in the resulting clusters. Hachaj and Ogiela [12] propose a graph-based community detection approach through clustering popular hashtags. Their unsupervised method analyses trending hashtags usage, assuming that Twitter users who use those popular tags are interested in similar topics. Nevertheless, this approach do not detect semantic themes in tweets as only hashtags are considered rather that the full text. Furthermore, hashtags are processed as distinct keywords, where the semantic relationships between them are ignored.

A correlation-driven clustering approach is prposed in [13]. The correlation is used in clustering tweets keywords into groups of words, such that the average intra-cluster correlation is higher than a given threshold. A high threshold is expected to generate two or more clusters referring to the same event. Whereas a smaller threshold may result in clusters that match separate events however, has the risk of clustering keywords that are loosely correlated, yielding higher level of noise. Hence, each cluster represents an event mentioned by a large number of posts from the input stream. While the proposed approach does not require *k* to be predetermined, the keyword-based analysis approach tends to ignore semantic relationsships in a stream and consequently generate clusters that are less meaningful. A hybrid hierarchical approach of agglomerative and divisive clustering was proposed to dynamically create broad categories of similar tweets based on the appearance of nouns [14]. In this study, only nouns have been utilized as features as the authors claim they are the most meaningful entities among other part of speech tags, such as verbs, adjectives, and adverbs. Therefore, their approach tends to discard all sentence tokens but nouns. The adopted bottom-up technique merges similar clusters together to reduce their redundancy, in which a recursive and incremental process of dividing and conquering clusters has been applied in order to produce more meaningful sorted clusters. The divisive stage works by dividing clusters down the hierarchy to arrange most similar tweets in different clusters. Afterwards, the bottom-up procedure is applied to remove or merge redundant information, if any. This proposed combinatorial approach showed increase in clustering effectiveness and quality compared to standard hierarchical algorithms. However, due to tweets' challenges discussed in Section 1, some tweets might lack the presence of nouns to form a rich nouns foundation in the clustering dataset. Therefore, it should useful to consider other textual features in addition to nouns to enhance the system's performance.

Review of related work has highlighted a specific gap in the literature. While affect studies have utilized clustering for the analysis of microblogging posts and extending to user behaviours and community detection, most of these studies use traditional clustering algorithms such as k-means, which requires the number of clusters to be determined beforehand. Due to the nature of microblogging platforms, the content generated by users is unpredictable and may contain interrelated conrovesial discussions [9]. Therefore, it is infeasible to prededict the optimal number of clusters for such data. Further-

more, statistical tests performed to detect the optimal number of clusters, such as the so-called 'elbow' phenomenon [15] or the 'gap statistics' [16] may provide insights on the optimal value for k in producing clusters with maximum inter-cluster and minimum intre-cluster distances. However, as Twitter posts where converted into sparse bog-of-word (BOW) vectors, each produced clusters may contain posts, in which their vector representations are close in a high dimensional model space, while their actual semantic equivalence is marginal.

Another related issue is concerned with the distance measure used in cluster analysis of microblog posts –the distance measure. Eclidean distance is often used in classical clustering algorithms, such as *k*-means. While this measure demonstrates large range of successful numerical-based clustering applications, they may not be the optimal choice for analyzing textual data such as tweets. Microblogging posts are modelled in a high dimensional space and are uneven in length. Thus, the magnitude of the vector representing each word does not matter. Euclidean distance is susceptible to documents being clustered by their L2-norm (i.e. magnitude, in the 2 dimensional case) instead of direction, where in high dimensional space of text, it could give rise to a large L2 norm. Therefore, a distance measure that keeps the "phase" (i.e. direction) information such as cosine similarity, which is missed in Euclidean distance, is important to capture the semantic similarity despite the high dimensionality of text.

Another important factor in clustering microblogging posts is the feature extraction and vector representation. Current state-of-the-art research in this area often provide poor vector representations for microblogging posts. These posts often contain rich meaning and less context. Therefore, features that are stripped away such as function words in traditional document analysis may still carry structural and syntactical information and cannot be ignored in short text posts [17]. Previous work utilized either single features such as hashtags [18] or particular part of speech (POS) tags [14], or represent microblogging posts as a sparse bag of words vector model [11].

In this research, the authors develop and validate a novel recursive-based cluster analysis (RBCA) algorithm that uses TREASURE –tweet similarity measure to approach the problem of detecting semantic themes in microblogging posts. The proposed approach aims to fill the gap of meaningless clusters and overcome the weaknesses observed in previous applications for clustering microblog posts.

3 Datasets

This section provides a description on the datasets the authors used to evaluate the proposed algorithm. Due to the lack of available multiclass-labelled benchmark on microblogging posts, SemEval-2014 STS.tweet_news similarity-labelled dataset is used [19]. SemEval-2014 is a collection of computational semantic analysis tasks intended to explore the nature of meaning in language. Part of the published trial datasets is a tweet-news dataset containing 750 annotated pairs. The gold standard implements a 6-point Likert scale to interpret the degree of similarity between pairs, as defined by Agirre [20]. The focus of this study is concerned with detecting semantic themes in microblogging posts. Using a similarity-annotated dataset not only provides ground truth on pairs' belongengness, but also determines the degree of semantic belongingness, from which the clustering similarity threshold, δ, can be derived. Therefore, SemEval-2014 STS.tweet_news benchmark is considered relevant to this research.

3.1 Data Acquisition

Another dataset was constructed on the political domain of the EU Referendum. This has been an active trend in microblogs and a rich source of controversial views. The United Kingdom European Union Membership (known as EU Referendum) took place on the 23rd of June 2016 in the UK. Based on a voting criteria, the voters were exposed to two opposing campaigns supporting remaining or leaving the EU. Three months prior to the day of the referendum, the data collection process has commenced through the use of Twitter Application Programming Interface (API), and went on until one month past that day. The Twitter streaming API allows for establishing a connection and continuously streaming real time tweets according to a specified set of search terms. Twitter streamed instances are returned as JavaScript Object Notations (JSON) data structures, which are composed of multiple metadata per tweet. A total of 4 million JSON objects were collected and stored in a NoSQL database called MongoDB [21]. Each instance in the dataset is a tweet associated with multiple metadata. These metadata contain information relating to the tweet, users, and entities. Quiries are designed to retrieve the tweets' text while accompanying attributes are kept stored for future research.

3.2 Data Preprocessing

An important part in creating a quality dataset for evaluating the proposed approach is data filtering and preprocessing. This research follows a composite preprocessing methodology in order to generate a semantic-rich set of tweets in the political domain under consideration.

1) The first stage follows the pre-processing heuristic proposed in [22] for cleaning short text in OSN. This pre-processing methodology takes into consideration the characteristics of the text and common conventions (e.g. retweets, hashtags, mentions, etc.) in Twitter. It eliminates redundant tweets as well as features that can be misleading for the similarity computation algorithm, yet preserves important information. For example, words containing repeated letters to emphasize sentiment are standardized to their original form. This enables both semantic and syntactic components of TREASURE to extract corresponding features and perform similarity computations.
2) A structure-based filtering is performed [46] to eliminate tweets that do not carry sufficient clean textual features for semantic processing. Tweets containing more than 2 user mentions or more than 3 hashtags, or less than 5 text tokens are filtered out. This is based on the hypothesis that tweets containing too many twitter-based user conventions such as hashtags and mentions, and less semantic content are generally very noisy [23]. Athough hashtags are considered as noisy as user mentions, we argue that the former contribute to the meaning of a tweet to a further extent. Therefore, we favor hashtags and empirically set their acceptable occurrence threshold to 3 instead of 2. Consequently, the tweet length restriction is increased by 1 in order to accompany the additional allowance of hashtags. Due to the nature of political tweets on the active EU Referendum event, this stage filters out most

of the tweets and keeps only 137K semantic-rich tweets that satisfy the invasive preprocessing criteria.

4 Recursive-Based Cluster Analysis

In this section, the authors describe the novel recursive-based unsupervised algorithm peoposed for clustering microblogging posts.

4.1 Distance Measure

The proposed cluster analysis approach incorporates TRASURE as the algorithm's distance measure upon which tweets are either grouped or separated according to a similarity threshold, $\delta = 0.6$[1]. TREASURE is selected as the distance measure for clustering microblogs posts due to two reasons:

1) It is particularly designed to capture the similarities between Twitter posts, the most popular microblogging platform, and can be extended to other kinds of microblogging social networks.
2) Its architecture is composed of both semantic and syntactic components to capture a comprehensive set of features from the text. The semantic modules compute the semantic relationships between words based on an artificial neural network embedding model learned from a large corpus of tweet examples. Whereas the syntactical modules capture structural and syntactical features that are common in microblogs, which contributes to the overall similarity score.

TREASURE produces a similarity score following a 6-point Likert scale, $S \in [0,5]$, such that a score of 0 demonstrates no similarity (i.e largest distance) and 5 indicates maximum similarity and thus vectors are represented in the same point in a space model (i.e no distance) between two tweets. In order to convert the similarity measure to a distance measure, S is normalized to [0, 1] using the following equation:

$$S_{norm} = \frac{S(T_1,T_2)}{S_{max}(T_1,T_2)} \quad (1)$$

the corresponding distance measure is then obtained using the following equation:

$$d(T_1,T_2) = 1 - S_{norm}(T_1,T_2) \quad (2)$$

similarly, the similarity threshold is converted according to the distance measure, $\delta = 0.4$.

4.2 Clustroid Re-Computation Methodology

Clustering data points in a Euclidean space represents a cluster by its *centroid*, which is the center of gravity or the average of the points in the cluster [24]. However, when the space in non-Euclidean, which is common in clustering unstructured text, distances cannot be based on *location* of points. In such case, a problem arise when each cluster requires a representative data point, but a collection of points cannot be represented by their centroid because the space is non-Euclidean. A solution may be to select a point from the cluster data to represent that cluster. The selected data point, in some sense, lies in the center by picking up the one that is ideally close to all the points of the cluster. The cluster representative point is called the *clustroid*.

The clustroid can be selected in various ways, each aiming to minimize the distances from the clustroid and every point in the cluster. A common choice is selecting the clustroid to be the point with minimum sum of the distances to the other points in the cluster [24]. However, this method is insufficient for large n as it involves multiple iterations over the cluster points to compute pairwise distances, a complexity of $O(n^2)$.

The proposed algorithm implements a local optimal solution to derive the representative clustroids at O(1), which does not require traversing over the cluster points. This is computed based on the geometry of triangles [25] as described in Section 4.2.1.

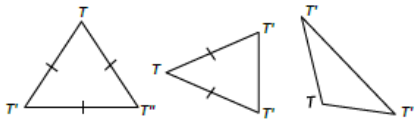
Deriving the clusteriod is determined based on two interrelated constraints, *cluster size* and *distance*. The proposed algorithm assigns instances to clusters as a linked list in a stack, such that last items inserted are first out (LIFO) to perform recomputations for determining the new clustroid [26]. The distance is calculated depending on the cluster size, which is identified by the instances contained in that cluster. The cluster size would be one of four categories:

1) *Singleton cluster* –when a new cluster is initialized, it contains only one post. In this case, this post is set as the clustroid.
2) *Doubleton cluster* –for clusters consisting of two instances, the last inserted post is assigned as the clustroid.
3) *Tripleton cluster* –for clusters consisting of three instances, the clustroid is determined based on the distances between these instances. The cases are further elaborated in the subsequent section (4.2.1).
4) *Multiple cluster* –these clusters contain quadruple or more instances. In this case, the candidate instances that are taken into consideration when determining the new clustroid are: 1) the new post that will be assigned to the cluster, 2) the previous clusteroid, and 3) the closest point to the previous clusteroid. The new clustroid is then nominated based on calculating the distances betweem the three points following the same heuristic as in a tripleton cluster.

4.2.1 Deriving Clustroids Based on Triangle Geometry

The authors discussed in Section 4.2 that, unlike continuous numerical data, microblogging posts are unstructured text that are not represented in a Euclidean space. This implies that the instances does not point to locations where average distances can be calculated to produce a cluster centroid. Although, multiple studies represent short text in a VSM [27, 28], which impose the curse of dimensionality problem [24]. These approaches generate very sparse vectors that require intensive computational resources in order to compute the centroids in a high dimensional space.

The algorithm proposed in this research identifies clusters' representative point (i.e. clustroid) based on modelling the three candidate instances based on a triangle geometric analysis in order to cover all possible cases. Each instance is assigned to an angle according to their pair wise distances calculated by inverting TREASURE similarity to a distance measure.



[1] Empirically derived threshold by experiments on labelled tweet pairs.

Figure. 1. Deriving clustroids based on triangle geometry

Figure 1 shows cases of the three candidate clustroids positions based on their distances. The pair-wise distances between the candidate clustroids are modelled according to the three main types of triangles, where T, T'', and T' denote the three last queued instances in the cluster, which are candidate clustroids. A triangle is a figure enclosed by three straight lines, where the sum of its three angles, $\angle ABC = 180°$ [25], where A, B, and C are the interior angles of the triangle. An angle degree refers to the direction of a triangle side, whereas the magnitude of the sides demonstrate the distance between two angles. In this research, the authors focus on the distance between instances rather than the direction (i.e. angle degree). Towards determining the new clustroid for tripleton and multiple clusters, the distances between candidate clustroids represent triangle straight lines, which can be one of the following cases:

Case 1. *Equilateral triangle* –figure 1.a represents $\Delta T'TT''$, a triangle in which all sides are equal. This means that the distances, $d(T,T')$, $d(T, T'')$, and $d(T', T'')$ are equal. In this case, the last assigned instance, T'', is defined as the new clustroid, C.

$$\because\ d(T,T') = d(T,T'') = d(T',T'')\ \therefore C = T'' \qquad (3)$$

Case 2. *Isosceles triangle* – figure 1.b represents $\Delta TT'T''$, a triangle in which only two sides are equal. This case represent one of two sub cases: 1) the size of the equal sides is less than the size of the third side such that,

$$d(T',T'') > \frac{d(T,T')+d(T,T'')}{2}$$

$$\because\ (TT' < T'T'') \wedge (TT' = TT'')\ \therefore C = T'' \qquad (4)$$

The clustroid, C, is set as the point that minimizes the sum of distances to other points, which is T in this case [24]. 2) The size of the equal sides is greater than the size of the third side, equations (4) and (5) become

$$d(T',T'') < \frac{d(T,T')+d(T,T'')}{2}$$

$$\because\ (TT' > T'T'') \wedge (TT' = TT'')\ \therefore C = T' \qquad (5)$$

In this sub case, even though T resides at an equally distant point to T' and T'', it does not represent the majority of the cluster's instances. Therefore, T' instead is assigned as the new clustroid.

Case 3. *Scalene triangle* –the most common case where candidate clustroid instances have different pair-wise distances, such as Figure 1.c., which shows $\Delta TT'T''$, a triangle with unequal sides. In this case, the sum of distances is computed for each instance and the one with the minimum value is considered the representative instance, C [24].

$$\exists C\ \in \Delta TT'T'',\ C := \arg\min_d \sum f(x) \qquad (6)$$

Where *f(x)* is the distance function *d*, between each instance, *x*, and other candidate instances, such that the point that satisfies the minimum sum of distances is set as the new clustroid. In the case present in Figure 1.c, $C = T$.

4.3 Clustering Algorithm

The proposed algorithm performs iterates over the collection of data points (i.e. microblogging posts) and generates non-overlapping clusters. It implements a crisp partitioning methodology where each data point belongs to one and only one cluster. Table 8.1 presents a pseudocode of the implemented SBCA algorithm. It demonstrates the recursive iterations performed from initiating a new cluster to the stage where all data points are assigned to clusters.

Table 1 The SBCA algorithm pseudocode

Algoritm 2 SBCA for microblogging posts using TREASURE

```
1   function SBCA(E, τ):
Input: Let A_k be the array of cluster's dictionaries, k, A_c be the
array of clustroids, C, and E be the dataset of microblogging
posts, T_i, where i = {1, 2, 3, 4, ..., n}, len(E) = n, considered for
cluster analysis, the distance threshold τ_dis.
Output: assignment of T to the relevant cluster dictionary, k, satisfying d(T, C) < τ_dis, where C is the clustroid.
2   T ← first(E)
3   k_1 ← T
4   c_1 ← T
5   A_k ← k_1
6   A_c ← c_1
7   while not at end of E do:
8   loop through each cluster center, A_c, where c_i ∈ k_i, i = {1, 2,
    3, ..., len(A_c)}.
9       T ← next(E)
10      distance ← 1 – (S(T, C_i)/ S_max(T, C_i))
11      if distance[2] < τ_dis then
12          assign T to k_i
13          k_i, c_i = UpdateSums(T, k_i)
14      else
15          initialize new 'k
16          'k ← T, d(t,c ) > τ_dis
17          'c ← T
18          A_k ← 'k
19          A_c ← 'c
20  end function SBCA(E, τ)
```

```
1   function UpdateSums(T, k):
Input: T is the new instance that will be assigned to the dictionary, corresponding to cluster k, the distance threshold τ_dis.
Output: k updated with new sums of distances for each instance
after the insertion of T, and the new clustroid with the minimum
sum.
2   min = 0
3   C ← T
4   foreach j, sum in k:
5   j is an instance in k where j ∈ {1, 2, 3, ..., len(k)}
6       sum ← sum + (1 – (S(j,T)/ S_max(j,T)))
7       if min = 0
8         min = sum
9       else
10        if sum < min
11          min = sum
12          C ← j
13  return k , C
14  end function UpdateSums(T, k)
```

4.4 RBCA Complexity

In this section, the authors analyze the complexity of the proposed algorithm in order to figure out how well it scales to larger datasts in relation to other solutions.

In terms of complexity, the SBCA algorithm shares the same time complexity as *k*-means partition-based clustering (worst case is $O(n^2)$), which is generally considered a low computa-

[2] Where *distance* τ_{dis} = 0.4 was derived from empirically determined similarity threshold.

tional cost algorithm [29]. The space requirements for the SBCA algorithm are modest because only the data points are stored. Therefore, the specific storage requirements are

$$\text{Space complexity} = O((K+f)n) \text{ , hence } O(n)$$

Where K is the number of clusters, f is the number of features (i.e. attributes), and n is the number of data points. The run time requirement of SBCA is linear to the number of data points. In particular, the time complexity is

$$\text{Time complexity} = O(I*K*f*n), \text{ worst case would be } O(n^2)$$

Where I is the number of iterations required to update the sum of pairwise distances in each cluster. Therefore, SBCA is basically linear in the number of data points. This makes the SBCA algorithm quite efficient for clustering microblogging posts.
Compared to hierarchical approaches, the agglomerative (bottom-up) algorithm has a time complexity of $O(n^3)$, whereas the divisive (top-down) algorithm runs in even more time at $O(2^n)$ [30], which means that the SBCA algorithm scales better to large datasets such as microblogging posts.

5 Experiment Methodology

The subjective evaluation methodology is carried out through undertaking three experiments designed to evaluate the SBCA algorithm as follows:
Experiment (1) – this experiment was conducted using the STS.tweet_news benchmark dataset (described in Chapter 5), which consists of similarity ratings for tweet pairs. This experiment was performed in order to determine the optimal value of TREASURE similarity threshold, τ_{sim}, which will determine if an instance will be assigned to an existing cluster or to a new cluster.
Experiment (2) – this experiment was conducted with human participants to generate a benchmark of tweets classifications into semantic categories utilising the EU_Referendum dataset, which is a rich source of controversial views (described in Section 3).
Experiment (3) –this experiment used the threshold determined by experiment (1) in order to detect semantic themes in the EU_Referendum dataset. The resulting clusters were evaluated using the benchmark generated from experiment (2).

5.1 Experiment (1) Evaluation Methodology using the STS.tweet_news Benchmark

The STS.tweet_news benchmark dataset consists of tweet pairs that are annotated with similarity ratings. The lack of Twitter-based benchmarks that are annotated with actual multi-class classification of tweets that can be used to evaluate an unsupervised clustering algorithm has led to running the SBCA algorithm on the STS.tweet_news similarity benchmark dataset. The application of the evaluation metrics discussed in the subsequent section for different values of τ_{sim} is carried out to determine the optimal value for detecting semantic themes in Twitter feeds, which can be extended to different microblogging posts.

5.1.1 *Rationale for the Selection of Evaluation Criteria*

The STS.tweet_news benchmark dataset does not consist of classes from which each instance belongs. Therefore, it is imperative to design an evaluation methodology such that a similarity labelled benchmark can be utilised for the purpose of cluster analysis evaluation. The evaluation of the proposed clustering algorithm on the STS.tweet_news benchmark in order to determine the optimal value of the similarity threshold, τ_{sim}, is performed through four external evaluation criteria as follows:

- **Rand Index** – considers the assignment of tweets to clusters according to a series of decisions.

$$RI = \frac{TP+TN}{TP+FP+FN+TN} \quad (7)$$

- **Precision (*P*) and Recall (*R*)** – *P*/*R* are the most common measurements for evaluating classifiers, which can be used to evaluate the grouping decisions determined by a clustering algorithm.

$$P = \frac{TP}{TP+FP} \quad R = \frac{TP}{TP+FN} \quad (8)$$

- **F-measure** – this metric is defined as the weighted harmonic mean of precision and recall.

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P+R} \quad (9)$$

For each of the aforementioned evaluation metrics, the SBCA algorithm is executed for six consecutive cases. Each case uses a different value of τ_{sim} in order to determine the optimal parameter threshold value for the proximity measure (TREASURE). The proportion of correctly clustered observations determines the accuracy of the clustering algorithm. The higher this proportion, the better the algorithm.
Thus, the SBCA algorithm is evaluated on six different similarity thresholds τ_{sim}, spanning the three similarity ranges used in [31], which are:

- The *lower* bound, [0 – 2]
- The *neutral* bound, (2 – 3]
- The *upper* bound, (3 – 5]

From each range, two threshold values are used in the evaluation of the SBCA algorithm, such that, if a tweet, T, and a clustroid, C, has a similarity, $S(T, C) > \tau_{sim}$, T is assigned to the cluster where C is the representative tweet for. Otherwise, T is assigned to a new cluster.
The next section describes the SBCA results for each value of τ_{sim} using the aforementioned evaluation metrics along with a discussion on the value that provided the most accurate clusters according to the STS.tweet_news similarity labelled benchmark.

5.1.2 *Experiment (1) Results: The Optimal τ_{sim} Value*

The results of the evaluation metrics described in Section 5.1.1 can be derived using a contingency matrix of the decisions undertaken by the SBCA algorithm against the actual decisions.
Table 2 shows an ensemble of the evaluation results for different τ_{sim} values. From these results, it can be observed that the higher thresholds τ_{sim} (3.5 and 4.0) have higher recalls, but in-

crease false positives (FP) (the number of dissimilar tweets that were grouped in the same cluster), therefore, precision goes down. In contrast, the lower thresholds τ_{sim} (1.5 and 2.0) recorded higher precisions, but decrease false negatives (FN) (the number of similar tweets that were grouped in different clusters).

Table 2 Evaluation of the SBCA algorithm using different τsim values

τ	Precision	Recall	F-measure	Accuracy (RI)	Clusters (K)
1.5	97.3%	51%	66.9%	56.8%	6
2.0	97.4%	57.9%	72.6%	65.6%	15
2.5	94.4%	64.3%	76.5%	71.2%	37
3.0	91.3%	83.2%	87.1%	84.9%	52
3.5	73.1%	93.1%	81.9%	80.1%	84
4.0	50.4%	98.3%	66.6%	76.5%	131

The SBCA proximity measure (TREASURE) will be assigned the similarity threshold that provides a trade-off between precision (P) and recall (R). Since the F-measure is defined as the weighted harmonic mean of precision and recall, the threshold that demonstrates the highest F-measure is thus determined as the optimal parameter value for the SBCA algorithm. Table 2 shows an excellent performance (F-measure and accuracy) when $\tau_{sim} = 3.0$. Considering the number of clusters, K, it can be observed that there is a linear relationship between τ_{sim} and the number of clusters, such that more clusters are generated as τ_{sim} increases and vice versa. Hence, a low value of τ_{sim} generates a coarse grained clusters, whereas higher values generate finer-grained clusters. Moreover, it can be observed that the number of clusters generated for τ_{sim} at 3.0 is the closest to the mean number of clusters, which is:

$$\mu(K) = (6+15+37+52+84+131)/6 = 54, \text{ which is} \approx 52.$$

The SBCA algorithm generating large number of clusters is attributed to two interrelated factors:

1. The STS.tweet_news dataset consists of 1500 tweets in the general domain of news, which contains tweets related to different events and topics.
2. TREASURE uses the Google News pre-trained word embedding model (described in Chapter 6), which may not contain specific words used in the STS.tweet_news dataset and thus tend to generate lower similarity values causing the SBCA algorithm to generate new clusters.

Experiment (1) provided results that demonstrate an optimal value of τ_{sim} at 3.0 for clustering microblogging posts utilising the STS.tweet_news similarity labelled benchmark. That is, the SBCA algorithm will assign tweets to the same cluster if and only if they share a similarity score > 3.0 ($S > \tau_{sim}$), according to TREASURE STSS measure integrated in the SBCA algorithm. The next section describes the experiment carried out to detect semantic themes within the EU_Referendum dataset using the similarity threshold determined in experiment (1), which is $\tau_{sim} = 3.0$, for the SBCA proximity measure.

5.2 Experiment (2) Detecting Semantic Themes within the EU Referendum Dataset

This section describes the experimental methodology and the detected semantic themes (i.e. generated clusters) in the EU Referendum dataset. Experiment (3) will provide a subjective evaluation of the generated clusters through running a human experiment to gather judgements on the belongingness of a subset of the results to their relevant clustroids.

The SBCA algorithm follows a divisive approach such that all observations in the dataset start in one cluster. The cluster analysis commences by assigning a random observation, T_r, as a cluster center (i.e. clustroid). A recursive series of splits are subsequently performed based on comparing each observation with the derived clustroids. An observation, T_r, is assigned to an existing cluster if it satisfies a certain threshold, τ_{sim}, which is determined to be 3.0 (Experiment 1). Otherwise, a new cluster is generated and T_r is assigned as the new cluster's clustroid, T_c. This process recursively carries on until all observations in the dataset are assigned in clusters. Unlike most clustering algorithms that require the number of clusters to be determined beforehand, such as k-means, the SBCA algorithm does not apply this condition. Instead, the number of clusters in the dataset is dynamically determined according to the specified similarity threshold, τ_{sim}. This linear relationship implies that as the value of τ_{sim} increases, more clusters are generated and vice versa, as shown in Table 2, Section 5.1.2.

5.2.1 *The EU Referendum Dataset Sampling Methodology*

A cluster analysis of the entire EU Referendum dataset would be a complex and time consuming process (given the dataset size as discussed in Section 3 and algorithm complexity as discussed in Section 4.4). Therefore, a subset of the whole corpus of collected tweets is derived, such that the complete timeframe for the data collection process is spanned. Although it has been reported that 10% of a dataset is considered a representative sample set [32], collecting a random 10% of the whole dataset may introduce bias in the resulting tweets and miss out on important events.

Thus, the methodology for constructing a representative sample is conducted as follows:

1. The corpus of pre-processed tweets is divided into four groups according to the month a tweet has been streamed.
2. For each month during the data collection, the group of corresponding tweets is further split into four groups according to the week of tweet streaming.
3. The result is a corpus of tweets organized into four main groups corresponding to the four months of data collection and each group contains four subgroups according to the week a tweet has been streamed.
4. The representative subset is created by retrieving a random sample of 10% from each of the sixteen subgroups in order to span the entire data collection period.

This sampling methodology resulting in 13.7K tweets, not only ensures a representative subset is constructed in terms of size, but in content as well. The SBCA algorithm is applied on the

sampled subset of tweets using TREASURE at the similarity threshold, τ_{sim} = 3.0. For clustering tweets on the EU_Referendum, TREASRE uses the corresponding EU_Referendum pre-trained word embedding [33]. The eleven themes generated by the SBCA algorithm are shown in Figure 2 along with each theme cluster size.

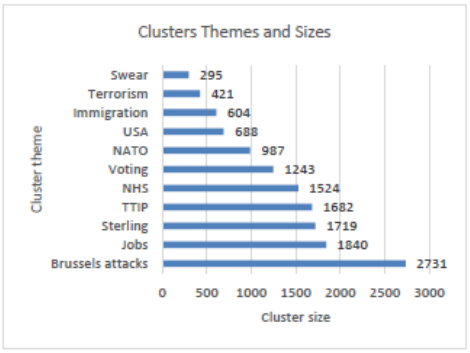


Figure 2 The EU Referendum themes detected by the SBCA algorithm

The next section provides Experiment (3), which describes the subjective evaluation of the generated clusters through running an experiment with humans to gather classifications of random tweets from the sampled subset to their relevant clustroids

5.3 Experiment (3) Evaluating the SBCA Detected Themes through a Multi-Class Benchmark

This section describes the third experiment, which is divided into two stages. Firstly, a human experiment is conducted to generate a reliable multi-class labelled benchmark from the EU Referendum sampled tweets. Secondly, the generated clusters of semantic themes described in Experiment (2) are subjectively evaluated using the multi-class benchmark produced in the first stage.

5.3.1 *Producing the EU_Referendum Multi-Class Benchmark*

The experimental design and instruments used for collecting human classifications of tweets from the EU Referendum dataset is illustrated in this section. The majority of the gathered EU Referendum class annotations will be used as a benchmark for a subjective evaluation of the SBCA and an extrinsic evaluation of TREASURE. The human subjective judgements on mapping tweets to the most relevant class was gathered using a closed-ended questionnaire. These judgements form a subjective qualitative control that is used to assess the quality of the SBCA algorithm in detecting semantic themes within microblogging posts.

This section describes the methodology undertaken in constructing the following elements related to the human experiment:

1. The tweets and clustroids – includes obtaining random tweets from the SBCA generated clusters in which humans will be asked to assign them to their most appropriate category (through mapping a tweet to a clustroid).
2. The questionnaire design – includes the design of the task instructions such that less confusion is introduced to attain consistency between judges in order to produce a reliable benchmark.

5.3.2 *Deriving Random Tweets from Clusters*

In psychology, the capacity of information, i, that can be received, processed, and remembered in immediate memory of a typical human cognitive system is seven plus or minus two [34], that is $i \in r$, where r = {5, 6, 7, 8, 9}. The methodology of producing the benchmark of classification judgments on the RBCA generated clusters for the EU_Referendum sample is based on this psychological theory. In order to make the classification task as simple as possible for participants to complete, the experiment has been designed according to the results of the SBCA algorithm.

1. Each clustroid, C, which is essentially the clustroid corresponding to each of the five largest generated clusters (shown in Table 3) are used to form the categories, which has the themes, *Brussels attacks, Jobs, Sterling. TTIP, NHS*. Only these five clusters are used in the experiment in order to avoid complexity and keep it simple for the participants to follow according to the Miller [34] psychological study.
2. For each C, three tweets are randomly selected to avoid bias and included in the experiment.
3. This subsampling process is performed for each representative tweet in the largest five generated clusters.
4. The resulting 15 tweets are used to form the human semantic classification benchmark on the EU Referendum dataset.

Table 3 Clustroids of the five largest tweets used in the experiment

Category	Clustroids (C)
A	Brussels terror attacks increased Brexit risk. Prayers go out to all families touched by the Brussels bombings today
B	EU Referendum Briefing on Living and Working in the EU #ProtectJobs #Expats
C	Sterling slides on renewed Brexit worries
D	#Brexit Emerges As Threat To TTIP[3] Deal
E	It's the EU or the NHS. I prefer the NHS. Britain's NHS can't survive staying in the European Union

This sampling methodology is performed to prevent any bias being introduced by selecting the tweets included in the experiment. The subsequent section describes the design of the questionnaire and the population sampling for participants.

5.3.3 *Questionnaire Instructions for Participants*

This section describes the design of the questionnaire in terms of the instructions and guidance provided to the participants. The participants were provided with an introduction to the study and the aim of undertaking this research. Due to the nature of the language used in OSN, participants were told that they might find some of the words that are used in tweets of-

[3] Transatlantic Trade and Investment Partnership (TTIP)

fensive and that they can withdraw from the experiment at any time, if they wish. Participants were provided instructions about the assignment process of tweets to their best match from the table of clustroids using the category alphabetical identification based on their interpretation of the meaning of the tweet.

5.3.4 *Sampling the Population for Participants*

The aspiration to represent the general population is restricted due to two issues:

1. Participants would be performing the classification task without supervision.
2. The tweets are rich in political interrelated information and thus require adequate political background to be able to interpret the latent semantics. The younger population, although maybe more familiar with Twitter terminology, generally have less political background to qualify them in judging such rich semantic pairs.

Thus, the sample was restricted to adults with graduate-level education. The sample was also restricted to include only native English speakers to ensure that the language used in the experiment is completely comprehensible and thus semantic-based classifications would not be influenced by anticipating text meaning or false interpretations. The 32 total participants volunteered without compensation. The use of 32 participants is commonly considered a representative population sample in similar studies [35-37]. The semantic classification experiment does not require collecting any personal information from any participant, such as age or gender, and therefore no sensitive personal data is held.

5.3.5 *The Produced EU_Referendum Multi-Class Benchmark*

The production of the EU_Referendum multi-class benchmark involved asking participants to complete a questionnaire, classifying tweets that are listed in a randomized order to their best matching clustroid from the provided list of clustroids. The participants were asked to complete the classification annotation questionnaire in their own time and to work through from start to end according to the given instructions. The 32 participants assigned each of the 15 tweets to their best matching cluster category from Table 3 and the majority of the judgments obtained by the participants was determined as the actual class for each tweet. The resulting benchmark can be seen in Table 4, where all human classifications are provided as the major category score obtained for each tweet alongside the SBCA classifications.

Table 4 The EU_Referendum multi-class benchmark results

Id	Tweets	Human Classifications	SBCA
1	Has anyone from the Brexit Brigade addressed the issue of what will happen to existing EU citizens living and working in the UK	B	B
2	Sterling has dipped cause markets believe Brexit will happen-GOOD-spivs in the city will adjust after playing their gambling games	C	C
3	How can we save NHS inside EU	E	E
4	I'm very sad for the families of the Brussels victims, but not at all surprised it happened! Wake up Europe #Brexit	A	A
5	On one hand, there are decent human beings that send their sympathies to the Brussels victims and their families. And then there's Brexit.	A	A
6	#Brussels attacks: Terrorism could break the EU and lead to Brexit	A	A
7	Sterling slides on renewed Brexit worries - A decline of 1% marks the 25th day this year the pound has moved	C	C
8	@caddenlimos connecting low paid workers doing non skilled jobs from Poland with terrorism in Belgium	B	B
9	Brussels bombing rose Brexit risk. How to trade Pound in case of Brexit: GBP $USD #FX	A	C
10	I did worry about threat to NHS from TTIP - but EU and @EU_TTIP_team have listened to our concerns @HealthierIn	E	E
11	UK's NHS will NOT survive staying in the EU	E	E
12	#Brexit, a new threat to TTIP transatlantic trade talks	D	D
13	We must stay in #EU to protect jobs	B	B
14	Jobs rely on trade NOT a political union. Security relies on sharing information NOT a political union. #Brexit #StrongerIn #VoteLeave	B	D
15	Naive to think Brexit would solve the #ttip problem...only way to protect #NHS is for govt to exclude it from TTIP	D	D

5.3.6 *Evaluating the SBCA Detected Themes using the EU_Referendum Multi-Class Benchmark*

The EU Referendum multi-clas benchmark consists of tweets that are annotated with classes they belong to, which is used in this section to evaluate the SBCA algorithm. The application of the evaluation metrics discussed in the subsequent section for $\tau_{sim} = 3.0$ as determined by Experiment (1) is undertaken to subjectively assess the SBCA generated clusters provided in Experiment (2). The evaluation results will provide insights on the validity of the SBCA algorithm in detecting semantic themes within microblogging posts.

The EU_Referendum multi-class benchmark consists of classes from which each instance (i.e. tweet) belongs. Therefore, the evaluation of the SBCA generated clusters with reference to the EU_Referendum multi-class benchmark will be conducted using the *Purity* external evaluation measure in addition to the criteria described in Section 5.1.1.

Purity is a simple and transparent evaluation measure [38]. To compute *purity*, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly as-

signed tweets instances and dividing by N, which is the total number of clustered instances in the dataset. Purity can be formally defined as:

$$\text{Purity}(\Omega, C) = \frac{1}{N}\sum_k \max_j |k_i \cap c_j| \quad (10)$$

Where $\Omega = \{k_1, k_2, k_3 \ldots k_i\}$ is the set of clusters and $C = \{c_1, c_2, c_3 \ldots c_j\}$ is the set of classes. The k_i is interpreted as the set of tweets determined by the SBCA algorithm as belonging to k_i and c_j as the set of tweets determined in the EU_Referendum multi-class benchmark as belonging to c_j .

The five external evaluation criteria are computed to conduct an in-depth validation of the SBCA algorithm with reference to the EU_Referendum multi-class benchmark, where results are discussed in the subsequent section.

6 Results and Discussion

The SBCA evaluation results using the five external evaluation criteria are provided in Table 5.

Table 5 Evaluation of the SBCA algorithm

	Purity	Precision	Recall	F-measure	Accuracy (RI)
Lower bound	0.0	0.0	0.0	0.0	0.0
Upper bound	100%	100%	100%	100%	100%
SBCA value	90%	80%	75%	77.4%	92.6%

The discussion on the performance of the SBCA algorithm is conducted in terms of the external evaluation criteria as well as the clusters sizes. With regard to the *Purity*, the SBCA is considered to generate 90% pure clusters which is considered a very good level of purity [39]. The F-measure, based on a weighted harmonic mean of precision and recall, recorded 77.4% by the SBCA algorithm on the EU_Referendum dataset. However, because the F-measure does not take into account the true negatives [40], it is generally considered limited in capturing the full story [41]. Therefore, the accuracy (RI) is also computed in interpreting the results of the SBCA algorithm. The evaluation results demonstrated that the SBCA algorithm achieved an accuracy of 90.5%. Based on a similar study, which aimed to perform fuzzy clustering of health surveillance terms in social media, achieved an accuracy of 87.1% [31] that was reported as excellent, SBCA is thus considered to achieve an excellent accuracy at 90.5% as demonstrated in Table 5. Compared to SBCA performance on the STS.tweet_news dataset shown in Table 2, the clustering algorithm achieved an 8% increase in terms of accuracy when applied on the EU_Referendum benchmark. This increase is anticipated to be attributed to the correlation of TREASURE on the EU Referendum dataset being higher than the correlation on the STS.tweet_news general domain dataset, which was originally related to the different word embedding models used for each dataset, from which the semantic relationships between words are computed. In terms of the cluster sizes, a sharp decrease can be observed on the clusters generated from the EU_Referendum dataset compared to the clusters generated from the STS.tweet_news dataset. The SBCA algorithm generated eleven clusters from the EU_Referendum dataset and, at the same similarity threshold $\tau_{sim} = 3.0$, it generated 52 clusters from the EU_Referendum dataset. This difference may be attributed to the following detailed reasons:

1. As the STS.tweet_news dataset was aggregated for the purpose of semantic similarity of tweet pairs, it may not be a good candidate for cluster analysis. This is due to the too many general topics and different news and subjects contained within the 1500 instances. Moreover, there are only few tweets sharing similar meanings compared to the tweets in the EU_Referendum dataset. On the other hand, the EU_Referendum dataset is domain-specific which, due to the controversial views of users concerned with this political event, the dataset is considered to contain different themes that reflect the users' intentions behind their decisions to either leave or remain in the EU. These themes are apparent in the naturally occurring clusters generated by the SBCA algorithm, such as the NHS, drop in the British pound (cause and effect), trade deals with the USA, terrorist attacks, etc. Each of the generated clusters may have controversial views which encourages either the 'stronger in' campaign or the 'Brexit' campaign. Therefore, the EU_Referendum dataset is considered a good candidate for cluster analysis as it provided insights on the intentions, argumentation mining, wider view of different communities that can be detected by posting similar tweets, and other use cases that demonstrate the usefulness of the SBCA algorithm in detecting semantic themes within microblogging posts.
2. A technical and important factor that is considered to have contributed in the difference in cluster sizes is related to the SBCA proximity measure (TREASURE). TREASURE incorporates a word embedding model from which it computes the semantic relationships between words. The pre-trained model used in Experiment (1) is different than the one used for Experiment (2). In the first experiment, TREASURE uses the Google News pre-trained model when applied on the STS.tweet_news dataset. However, using a model trained on traditional text documents for the purpose of social networks linguistic analysis resulted in OOV words and missing terminology from the Google News pre-trained model. Thus, TREASURE tended to assign less similarity scores as a result of not recognising some of the words in a tweet (words that are not present in the pre-trained model). Consequently, new clusters are generated due to a similarity score that is less than the specified threshold causing a false negative by separating the two tweets being assessed for similarity (i.e. false separation decision). This is not the case for the EU_Referendum dataset, where TREASURE uses the corresponding EU_Referendum word embedding model [33]. Therefore, TREASURE is not likely to encounter any OOV or terminology that will not recognized because the model was trained on the four million corpus of tweets collected on the EU_Referendum. Consequently, TREAS-

URE tend to better capture the similarities between tweetsand thus it is less likely to generated generate new clusters as a result of false negatives.

7 Conclusion and Futur Work

The results from the experiments, using the external evaluation criteria with reference to the EU_Referendum multi-class benchmark, show adequate evidence to positively answer the research questions.

The main novel contributions in this chapter are:

- A new reliable benchmark of microblogging posts (tweets) assigned to their best match class, which is denoted by the clustroid of the corresponding cluster, labelled with class judgments by human experts with a good level of inter-rater agreement in the domain of Politics.
- A novel experimental methodology to produce a benchmark with human classifications derived from clusters, which are generated from a large dataset of raw microblogging posts.
- Evidence that the similarity threshold τ_{sim} = 3.0, which corresponds to τ_{dis} = 0.4 (applying Equations 1 and 2 respectively), provides the optimal value for the SBCA proximity measure generating the best set of clusters in terms of accuracy and F-measure compared to different threshold values.
- Evidence that the SBCA algorithm produces pure clusters from microblogging posts, particularly tweets.
- An evidence that the SBCA algorithm demonstrates a high level of accuracy in performing separation and combining decisions, which maximises true positives and true negatives.

References

1. Kumar, S., F. Morstatter, and H. Liu, *Twitter data analytics*. 2013: Springer Science & Business Media.
2. Castillo, C., M. Mendoza, and B. Poblete. *Information credibility on twitter*. in *Proceedings of the 20th international conference on World wide web*. 2011. ACM.
3. Krestel, R., et al. *Tweet-Recommender: Finding Relevant Tweets for News Articles*. in *Proceedings of the 24th International Conference on World Wide Web*. 2015. ACM.
4. Go, A., R. Bhayani, and L. Huang, *Twitter sentiment classification using distant supervision*. CS224N Project Report, Stanford, 2009. **1**(12).
5. Friedemann, V., *Clustering a Customer Base Using Twitter Data*. 2015.
6. De Boom, C., S. Van Canneyt, and B. Dhoedt. *Semantics-driven event clustering in twitter feeds*. in *Making Sense of Microposts*. 2015. CEUR.
7. Purwitasari, D., et al. *K-medoids algorithm on Indonesian Twitter feeds for clustering trending issue as important terms in news summarization*. in *Information & Communication Technology and Systems (ICTS), 2015 International Conference on*. 2015. IEEE.
8. Sinnott, R.O. and W. Wang, *Estimating micro-populations through social media analytics*. Social Network Analysis and Mining, 2017. **7**(1): p. 13.
9. Alnajran, N., et al. *Cluster Analysis of Twitter Data: A Review of Algorithms*. in *9th International Conference on Agents and Artificial Intelligence*. 2017. SCITEPRESS.
10. Alnajran, N., et al., *TREASURE: Tweet Similarity Measure Based on Semantic and Syntactic Computation*. ACM Transactions on Knowledge Discovery from Data (TKDD) 2018((under review)).
11. Li, L., et al., *A comparison study of clustering algorithms for microblog posts*. Cluster Computing, 2016. **19**(3): p. 1333-1345.
12. Hachaj, T. and M.R. Ogiela, *Clustering of trending topics in microblogging posts: A graph-based approach*. Future Generation Computer Systems, 2017. **67**: p. 297-304.
13. Olariu, A. *Clustering to improve microblog stream summarization*. in *2012 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2012)*. 2012. IEEE.
14. Kaur, N., *A combinatorial tweet clustering methodology utilizing inter and intra cosine similarity*. 2015, Faculty of Graduate Studies and Research, University of Regina.
15. Milligan, G.W. and M.C. Cooper, *An examination of procedures for determining the number of clusters in a data set*. Psychometrika, 1985. **50**(2): p. 159-179.
16. Tibshirani, R., G. Walther, and T. Hastie, *Estimating the number of clusters in a data set via the gap statistic*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2001. **63**(2): p. 411-423.
17. Li, Y., et al., *Sentence similarity based on semantic nets and corpus statistics*. IEEE transactions on knowledge and data engineering, 2006. **18**(8): p. 1138-1150.
18. Stilo, G. and P. Velardi, *Hashtag sense clustering based on temporal similarity*. Computational Linguistics, 2017. **43**(1): p. 181-200.
19. Guo, W., et al. *Linking Tweets to News: A Framework to Enrich Short Text Data in Social Media*. in *ACL (1)*. 2013.
20. Agirre, E., et al. *Semeval-2012 task 6: A pilot on semantic textual similarity*. in *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. 2012. Association for Computational Linguistics.
21. Banker, K., *MongoDB in action*. 2011: Manning Publications Co.
22. Alnajran, N., et al. *A Heuristic Based Pre-processing Methodology for Short Text Similarity Measures in Microblogs*. in *High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2018 IEEE 20th International Conference on*. 2018. IEEE.
23. Ifrim, G., B. Shi, and I. Brigadir. *Event detection in twitter using aggressive filtering and hierarchical tweet clustering*. in *Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014*. 2014. ACM.
24. Leskovec, J., A. Rajaraman, and J.D. Ullman, *Mining of massive datasets*. 2014: Cambridge university press.
25. Bird, J., *Basic engineering mathematics*. 2014: Routledge.
26. Aho, A.V. and J.D. Ullman, *Data structures and algorithms*. 1983: Pearson.
27. Laniado, D. and P. Mika. *Making sense of twitter*. in *International Semantic Web Conference*. 2010. Springer.
28. Mozetič, I., et al., *How to evaluate sentiment classifiers for Twitter time-ordered data?* PloS one, 2018. **13**(3): p. e0194317.
29. Salem, S.B., S. Naouali, and M. Sallami, *Clustering Categorical Data Using the K-Means Algorithm and the Attribute's Relative Frequency*. World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2017. **11**(6): p. 709-714.
30. Sharma, A., Y. López, and T. Tsunoda, *Divisive hierarchical maximum likelihood clustering*. BMC bioinformatics, 2017. **18**(16): p. 546.
31. Dai, X., M. Bikdash, and B. Meyer. *From social media to public health surveillance: Word embedding based clustering method for twitter classification*. in *SoutheastCon, 2017*. 2017. IEEE.
32. Severino, R., *Getting Your Random Sample in Proc SQL*. 2006.
33. Alnajran, N., et al., *A Word Embedding Model Learned from Political Tweets*, in *In Computer Engineering & Systems (ICCES), 2018 13th International Conference on*. 2018, IEEE.
34. Miller, G.A., *The magical number seven, plus or minus two: Some limits on our capacity for processing information*. Psychological review, 1956. **63**(2): p. 81.
35. O'Shea, J., et al. *A comparative study of two short text semantic similarity measures*. in *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*. 2008. Springer.
36. O'Shea, J., et al., *Benchmarking short text semantic similarity*. International Journal of Intelligent Information and Database Systems, 2010. **4**(2): p. 103-120.
37. O'shea, J., Z. Bandar, and K. Crockett, *A new benchmark dataset with production methodology for short text semantic similarity algorithms*. ACM Transactions on Speech and Language Processing (TSLP), 2013. **10**(4): p. 19.
38. Schütze, H., C.D. Manning, and P. Raghavan, *Introduction to information retrieval*. Vol. 39. 2008: Cambridge University Press.
39. Vanegas, J. and I. Bonet. *Clustering Algorithm Optimization Applied to Metagenomics Using Big Data*. in *Conference on Information Technologies and Communication of Ecuador*. 2018. Springer.

40. Mihalcea, R., C. Corley, and C. Strapparava. *Corpus-based and knowledge-based measures of text semantic similarity*. in *AAAI*. 2006.
41. Xiong, H., et al. *HICAP: Hierarchical clustering with pattern preservation*. in *Proceedings of the 2004 SIAM International Conference on Data Mining*. 2004. SIAM.

Noufa Alnajran is currently a PhD candidate in the School of Computation, Mathematics, and Digital Technology at Manchester Metropolitan University. Noufa's research interests under the Intelligent Systems Group at Manchester Met include Machine Learning, Social Networks Analysis, text mining, Big Data, and knowledge engineering. She is a System Analyst in the main Department of Information Technology at Princess Nourah bint Abdulrahman University. Noufa received BSc degree in Computer Science from King Saud University in 2010 and the MSc (hons) degree in Software Engineering from Prince Sultan University in 2015. Noufa has multiple publications in the field of Big Data and Machine Learning. She is currently working on joint research within the Intelligent Systems Group.

Dr. Keeley Crockett is a Reader in Computational Intelligence at Manchester Metropolitan University and leader of the Intelligent Systems Group. She has over 20 years' experience of research and development in Computational Intelligence algorithms and applications including the psychological profiling system, Silent Talker and the adaptive conversational agent tutoring systems, Oscar and Hendrix. She is the current Chair of IEEE Women in Engineering UKI, Vice Chair IEEE Women in Computational Intelligence. She has authored/co-authored over 90 peer reviewed papers and is currently one of the principal investigators on the H2020 European project iBorderCtrl. Keeley is a senior member of IEEE.

Dr. David McLean received BSc (hons) degree in Computer Science from the University of Leeds in 1989 and the PhD (neural networks) degree "Generalization in continuous Data Domains," from Manchester Metropolitan University in 1996. From 1996 to 1997, he worked for DERA (Malvern) and Thomson Marconi Sonar and became a lecturer at Manchester Metropolitan University in 1997. He is a member of the Intelligent Systems Group and is a founding member of Convagent Ltd., which conducts research and develops applications in the field of conversational agents. His other main interest is in automatic psychological profiling from nonverbal behavior using artificial intelligence techniques. He is a member of a research group that currently holds a patent for such a system.

Dr. Annabel Latham (FHEA MIEEE MBCS) is a Senior Lecturer in Computer Science and the Information Systems Curriculum Leader in the School of Computing, Mathematics, and Digital Technology at Manchester Metropolitan University. Annabel's research interests under the Intelligent Systems Group at Manchester Met include conversational agents, intelligent tutoring systems, big data, text mining, agent intelligence and knowledge engineering. Annabel leads the School's Athena SWAN project, aimed at changing culture and promoting a gender balance in Computer Science. Annabel is Chair of the IEEE CIS Education Multimedia subcommittee, Vice-Chair of the IEEE UK and Ireland Women in Engineering, and an active committee member of IEEE Women in Computational Intelligence Group, IEEE CIS Student Activities and IEEE CIS Social Media subcommittees.