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.

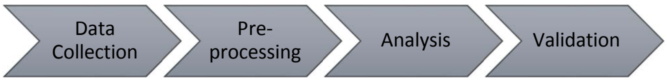

Fig. 1. A typical text mining process

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 &lt; &gt; &amp; 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, &amp; 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 different 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.

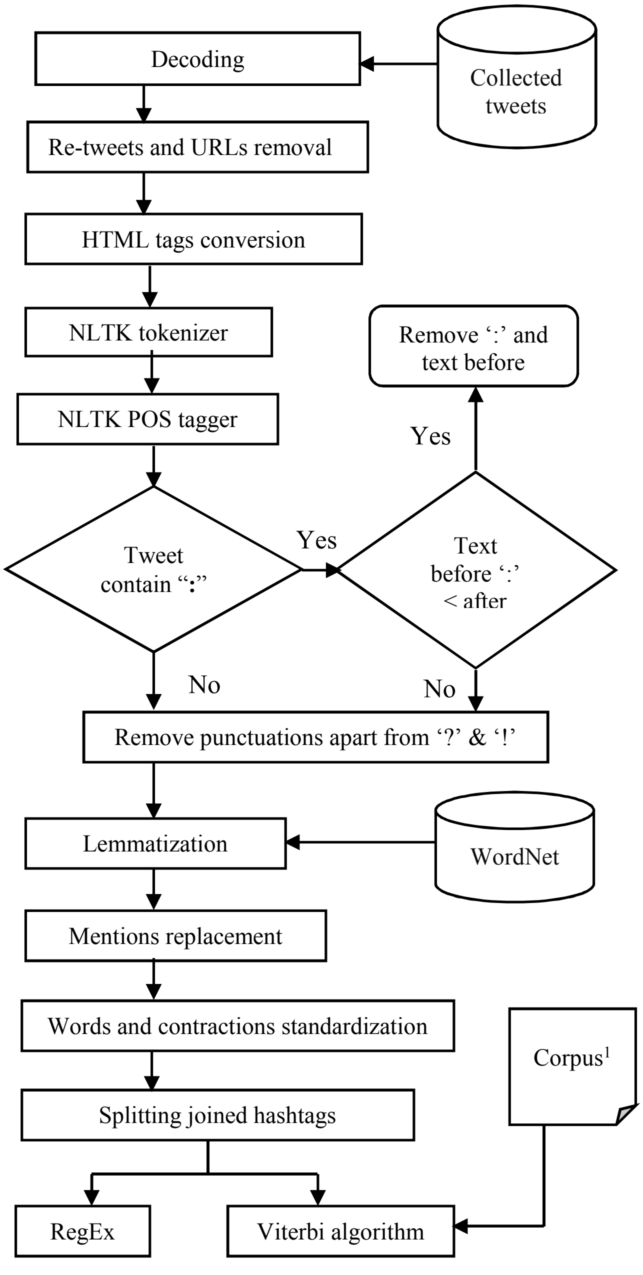

Fig. 2. Proposed heuristic-driven pre-processing components

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.

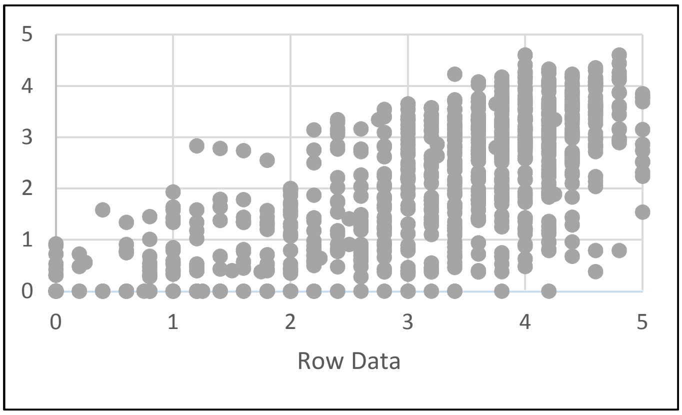

Fig. 3. Row-human data correlations scatterplot

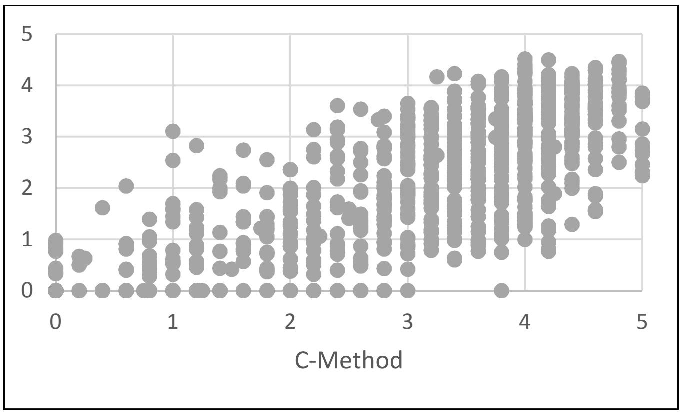

Fig. 4. C Method-Human correlations scatterplot

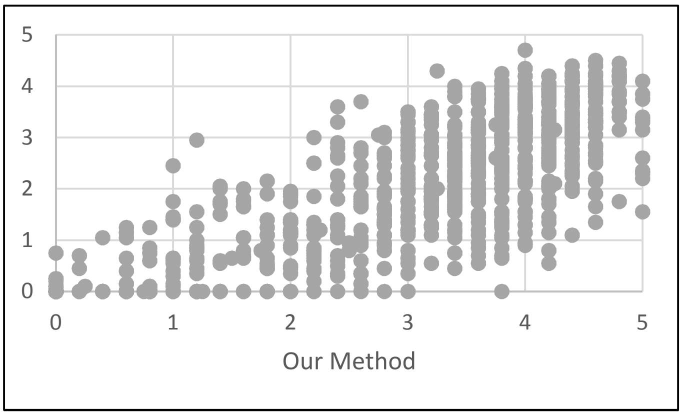

Fig. 5. Our Method-Human correlations scatterplot

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.