Translator Attribution for Arabic Using Machine Learning

Emad Mohamed[1], Raheem Sarwar *[2], and Sayed Mostafa[3]

[1]Research Group in Computational Linguistics, University of Wolverhampton, United Kingdom

[2]Department of Operations, Technology, Events and Hospitality Management, Manchester Metropolitan University, Manchester, M15 6BH, United Kingdom

[3]Department of Mathematics & Statistics, North Carolina A&T State University, USA

October 19, 2022

Abstract

Given a set of target language documents and their translators, the translator attribution task aims at identifying which translator translated which documents. The attribution and the identification of the translator's style could contribute to fields including translation studies, digital humanities, and forensic linguistics. To conduct this investigation, firstly, we develop a new corpus containing the translations of world-famous books into Arabic. We then pre-process the books in our corpus which mainly involves cleaning irrelevant material, morphological segmentation analysis of words and devocalisation. After pre-processing the books, we propose to use 100 most frequent words and/or morphologically segmented function words as writing style markers of the translators (i.e., stylometric features) to differentiate between translations of different translators. After the completion of features extraction process, we applied several supervised and unsupervised machine learning algorithms along with our novel cluster-to-author index to perform this task. We found that the translators are not invisible, and morphological analysis

*Corresponding Author (raheem.bwl@gmail.com)

may not be more useful than just using the 100 most frequent words as features. The SVM Linear Kernel algorithm reported 99% classification accuracy. Similar findings were reported by the unsupervised machine learning methods, namely, K-Mean Clustering and Hierarchical Clustering.

1 Introduction

Authorship Attribution is a sub-discipline of Computational Linguistics that tries to determine whether an anonymous document was written by one of several candidate authors. The field maintains that each author has a specific linguistic fingerprint (or author print) and that computational tools can discover this unique style. Authorship attribution is usually performed by extracting writing style features from the true writing samples of the candidate authors and learning a classifier on them to identify the true author of the anonymous text (Mohamed et al. 2021). The most frequent function words as stylometric features have been extensively used for authorship attribution tasks. As for the machine learning classifiers, support vector machines, k-nearest neighbors, logistic regression and random forests have been extensively used to perform authorship attribution tasks. In this paper, we ask a related question, but one which involves translators rather than authors: *Is there also a translator fingerprint?* Can Author Attribution methods work for Translation Attribution? In other words, given a set of target language documents and their translators, can we know which translator translated which documents? The question of *translator print* is different from that of *author print* since a translator is not conveying her/his own thoughts, and is thus under the influence of the source language. In fact, for some time, it was thought that the best translator is the one who is hidden, and *Translator Invisibility* was taken for granted. In fact, William Shapiro is quoted (in Venuti's The Translator's Invisibility) to have said:

> "I see translation as the attempt to produce a text so transparent that it does not seem to be translated. A good translation is like a pane of glass. You only notice that it's there when there are little imperfections – scratches, bubbles. Ideally, there shouldn't be any. It should never call attention to itself" (Venuti 1986).

This idea that the only good translators are invisible ones may seem to suggest that translators are indeed invisible. While many translators exert effort to make their translation read as *untranslated* as possible, it is worth investigating how far they succeed in this act of self-denial. *Translator*

Attribution is our way of investigating this question. Another aspect that may make translator attribution a more challenging task than authorship attribution is that translated books may also have revisers. In the corpus introduced in this investigation, most books have a translator and a reviser, and some of the translators of a book are revisers of other books. This will give us the chance to examine, through supervised learning and the confusion matrix, whether the misclassified examples are those involving authors-revisers or whether misclassification is non-conditioned on the double-roles played by translators.

To the best of our knowledge, this is the first study to investigate the translator attribution task for the Arabic language. Specifically, we investigate the Arabic translation of world-famous books. There are approximately 280 million native speakers of Arabic (Brown 2005), and it is the official language for 26 countries, and third most official language in the world followed by English and French. Arabic is gaining interest by people all around the world due to its socio-political importance, millions of Muslims (both Arab and non-Arab) are studying Arabic as the Holy Qur'an (the main religious text of Islam) is written in Arabic (El-Fiqi et al. 2011). Therefore, the interpretation of Arabic has wider significance. We formulate the following research questions to answer in this investigation:

Research Questions:

- Is is possible to differentiate between translations based on the translators' writing style? Are the translators invisible?
- What type of machine learning algorithms are the best performer for translator attribution task ?
- How many features are sufficient to perform translator attribution?
- Arabic is morphological complex language. Does morphological pre-processing such as identifying specialised function words help improve the accuracy of translator attribution task?
- Can authorship attribution methods be used for translator attribution?

In this paper, we propose a translation attribution solution which is consists of 3 main steps including preprocessing the books to convert them into a suitable content for analysis , extracting features from books, and apply machine learning algorithms on those feature vectors. Our contributions can be summarized as follows.

1.1 Summary of Our Contributions

- We present the first translator attribution study for Arabic translations of world-famous books.
- We introduce a new dataset to perform the translator attribution task for Arabic language.
- For translator attribution task, we propose to remove punctuation and perform devocalisation of the text book before extraction the stylometric features. We also propose to use function words as writing style marker for the translators (features) and the use of supervised and unsupervised machine learning algorithms to perform extensive experimental studies to solve this task.
- We propose a novel index, cluster-to-author ratio to sense translator invisibility.
- We will make our dataset and the scripts publicly available.

The rest of the paper is organised as follows. Section 2 reviews the existing works on translator attribution on English and provide a comparison between our work and existing studies on this topic. Section 3 describes our dataset and methodology for the translator attribution task. Section 4 provides the experimental results and discussion. Section 5 contains the conclusions and the future works directions.

2 Literature Review

Machine learning is a branch of computer science which focuses on the use of data and algorithms to imitate the way humans learn and gradually improving its accuracy. Translator attribution can also be considered as a machine learning problem. To the best of our knowledge, this is the first study on translator attribution for Arabic. The translator attribution task has been investigated for books translated into English (Caballero et al. 2021; Lynch et al. 2018; El-Fiqi et al. 2019). Baker (2000) conducted the first investigation on translator attribution using features such as type-token ratio and average sentence lengths as the writing style markers of the translators. Baker suggested the existence of translator fingerprints, and tried to identify some translators signatures in their translations. However, her study was limited in terms of computational linguistics analysis. Moroever, Baker used translations of different languages and for different texts. Furthermore, these translations were not for the same original texts. Such analysis left several open questions in the context of the differences in translators' styles.

Mikhailov et al. (2001) performed an analysis to identify whether there were style fingerprints left by translators. A parallel corpus of Finnish translations of Russian fiction literature was used in the study. The features of the translators' style were word frequencies, favourite terms, and vocabulary richness. According to their findings, the language of different translations of the same text done by different people is closer than the language of different translations done by the same translator. Despite the existence of some translation preference patterns, existing authorship methodologies failed to identify translator styles, according to their findings. Recently, several studies were performed to explore the problem of translator attribution (Rybicki 2012; Rybicki et al. 2010; Rybicki and Heydel 2013) using Burrows's Delta (Burrows 2002), a well-known technique for authorship attribution based on the z-score of the word frequencies. Rybicki et al. (2010) used this strategy to look into the contributions of Jeremiah Curtin, a translator, and his wife Alma Cardell to his translations. Rybicki proved that his wife wrote Memoirs of Jeremiah Curtin (1940). Rybicki (2012) used the same method to see if it could distinguish between translator collaborations on a single literary work. They looked at Virginia Woolf's novel Night and Day, which has 36 chapters. Anna Kolyszko, the first translator, died after completing the first 26 chapters, and Heydel, the second translator, completed the remaining chapters. The translations were clustered according to the translators using their proposed approach.

Despite the success of these investigations, the observed translator signature may be lost if analysed in the context of alternative corpora. Rybicki and Heydel (2013) later utilised the Delta metric to see if translations could be traced back to translators. Every attempt to do so failed, according to the author, because texts clustered based on the author rather than the translator. Several recent studies have found that translators leave traces on the texts that can be utilised to distinguish various translators (Forsyth et al. 2014; Hedegaard et al. 2011; Rybicki 2012; Lee 2018). Existing research on translation style, in particular, are contradictory (Lee 2018). Some people notice the stylistic presence of translators, while others don't. The lack of agreement in the literature leads one to believe that there are certain contextual elements influencing the visibility of translators. Lee's investigation validated the notion that the greater the structural distance between the two languages involved, the more probable the translator's style will become visible, as the higher distance will provide the translator more opportunity to be creative with his or her choices.

Covington et al. (2015) used quantitative stylometry to compare ten English translations of the same Bible chapter, then use clustering to create a dendrogram that reflects the translations' well-known history and intent. They used features like vocabulary richness, average sentence length, and average type-token ratio to make their decisions. They came to the conclusion that combining quantitative stylometry with clustering is a viable method for reconstructing literary history. There are two studies that are roughly linked to our work, however they are both English-focused (Lynch et al. 2018; El-Fiqi et al. 2019).

For the translation attribution task, Lynch et al. (2018) decided to employ two corpora. The first is a collection of seven works by Norwegian playwright Henrik Ibsen, which two translators have translated into English. The researchers used a parallel corpus in their first experiment, which consisted of training several Support Vector Machine (SVM) classifiers with the stylometric features such as ten most distinctive words, ten most distinctive word bigrams, and ten most distinctive Part-of-Speech (POS) bigrams. They then used this strategy to the remaining six plays (a nonparallel corpus) and used 18 document-level features such as average sentence length, type/token ratio, and average word length to train other machine learning classifiers (Naïve Bayes, Logistic Regression, and Decision Trees). The second corpus contains ten works by Russian author Anton Chekhov, which were translated into English by two female translators. Six of the ten were translated by both parties. On this corpus, they used the same methods as before. Finally, they demonstrate a clustering analysis on both corpora using a modified Burrows' Delta with the 100 most common words and the ten most distinctive terms. They attain excellent grouping by translation instead of work with only the 10 most distinctive terms, confirming the efficacy of said set of unusual words in the categorization task. El-Fiqi et al. (2019) applied network motifs in their second paper, arguing that they capture some grammatical patterns in each translator's writing. They cleaned and lemmatized the text as part of their data preprocessing, leaving just alphanumeric characters (thus removing punctuation).

3 Data and Methods

Our solution consists of three main steps: (i) collecting data and performing pre-processing, (ii) extracting features from the translated books and (iii) applying machine learning algorithms, both the supervised and unsupervised on the extracted features from the translated texts. Each of the steps of our solution is explained in the following subsections.

Translator	Books	Authors	Words
Abbas Hafiz	4	3	442391
Abulfattah Abdalla	8	2	136283
Hiba Ghanim	7	6	260104
Mustafa Fuaad	6	5	388263
Sara Taha Allam	15	9	346211
Sara Yaqut	9	5	303630
Total	49	30	1876882

Table 1: Data

3.1 Data Collection and Pre-processing

The data for this paper come from the Hindawi website, a website that publishes translations of world-famous books and classics into Arabic and makes them available in the public domain. We have chosen six translators, three males and three females, and we have selected their books as our corpus. This has resulted in 49 books, described in Table 1. The books go through a pipeline of text conversion, cleaning, and morphological segmentation.

Text conversion is a necessary step since the books are distributed in the *EPUB* format. EPUB is an XML-based ebook format that is widely supported, and it consists of XHTML files along with pictures and other material in a compressed format. We use Calibre for uncompressing the files and converting them into text.

Text cleaning comes next and it consists mainly of (1) separating punctuation from text, and (2) devocalisation. While separating punctuation from text is probably a universal text processing methods, devocalisation is probably specific to Arabic. Arabic is often written without the short vowel, so a word like *ktb* could be read as *katab*, *kutiba*, *kattaba* and *kutub* among other things. Furthermore, we have examined the way native speakers vocalise texts, in the not so many case in which they do, and we have found that it is inconsistent, and even sometimes wrong. We have found that the best way to maintain data integrity is to remove the short vowels completely from the words. An average of 16% of the words in each book are at least partially vocalised, with a minimum of 4%, a maximum of 39%, and a standard deviation of 8.5. The removal of punctuation and vocalisation helps reduce an otherwise artificially inflated type token ratio.

3.2 The feature set

We use two kinds of features in our experiments: (1) the top 100 most frequent words and (2) the top 100 function words. We notice that 34 of the top 100 most frequent words are morphologically complex, in most cases comprising a preposition and a pronoun.

These features have also been widely used in the literature. Function terms, not lexical ones, have been the focus of this field since the pioneering study on the disputed federalist texts(Mosteller et al. 1963) . The function words we wish to utilise, are non-contextual, that is, terms whose rate of use is nearly unchanging as the topic changes. As a result, the small filler words known as function words are particularly appealing for discriminatory reasons. The attractiveness of function words is summed up as follows (Kestemont 2014): (1) people use the same function words, making them easily comparable, (2) there is always enough of them due to their high frequency, (3) they are less affected by the genre or theme of the text, and (4) they are not as under the conscious control of the author as are lexical words. Other than authorship attribution, function terms have been utilised in numerous research. Emotion detection and analysis, lie detection and analysis, gender and age studies, and many more psychologically relevant investigations are all documented in the popular book The Secret Life of Pronouns.

3.3 Machine Learning Methods

There are mainly two types of machine learning methods namely supervised and unsupervised. more details about them is provided in the following subsections.

3.3.1 Supervised Machine Learning

For supervised Machine Learning, we adopt a practical approach by using the PyCaret[1] Library, which "is essentially a Python wrapper around several machine learning libraries and frameworks such as scikit-learn, XGBoost, LightGBM, CatBoost, spaCy, Optuna, Hyperopt, Ray, and many more." We have tried PyCaret against manually tuned algorithms and have found no major differences in quality. PyCaret achieves the same results as those algorithms carefully crafted by the team, but it has the edge when it comes to speed as it compares over 17 algorithms in a matter of minutes.

3.3.2 Unsupervised Machine Learning

We also utilize unsupervised learning techniques in the form clustering algorithms to further explore translators visibility. Specifically, we apply K-Means and hierarchical clustering algorithms

[1] `https://pycaret.readthedocs.io/en/latest/api/classification.html`

to the morphological processed function words data for each translator. The goal is to assess the translator visibility by comparing the number of clusters emerging from the clustering algorithms with the number of authors for the translator. The assumption is that the text of invisible translators should cluster into a number of clusters that is close to or significantly higher than the number of authors that wrote the text. Conversely, we assume that the translator has a detectable role when the translated text clusters into way fewer clusters than the number of authors of the text. For both clustering algorithms, the optimal number of clusters K_{opt} is determined, via grid search, as the smallest number K such that the percentage of between-clusters sum of squares (BSS) is at least 60%. For more details about K-means and hierarchical clustering, we refer the reader to Rodriguez et al. (2019).

4 Results

In this section we present the findings of our experimental studies based on supervised and unsupervised methods.

4.1 Supervised Learning Results

The results for supervised learning are in Table 2 and Table 3, and we can see that in both tables the CatBoost classifier is the best performing classifier and that the classification accuracy is 87% for the six classes without a major difference between morphologically processed and non-morphologically processed data. The following three results may be explicitly derived from this classification:

- *Translators are not invisible.* With a classification accuracy of 87% for six classes, this is extremely higher than a random result of 16.7%, which means each translator has his/her own style, and that **Translator Invisibility** is truly a myth. In the corpus introduced in this investigation, most books have a translator and a reviser, and some of the translators of a book are revisers of other books. Using confusion matrix as shown in Figure 1 we found that the misclassified examples are those involving authors-revisers.

- *The best machine learning algorithm.* Tree-based algorithms are the best performers in this experiment, especially Gradient Boosting Algorithms. The top 5 performing algorithms are various implementations of Gradient Boosting. This also indicates that implementations

Algorithm	Recall	Prec.	F1
CatBoost Classifier	0.8486	0.8716	0.8666
Extreme Gradient Boosting	0.8421	0.8668	0.8625
Light Gradient Boosting Machine	0.8338	0.8614	0.8566
Gradient Boosting Classifier	0.8249	0.8541	0.8478
Logistic Regression	0.8143	0.8375	0.8336
SVM - Linear Kernel	0.7906	0.8289	0.8143
Ridge Classifier	0.7852	0.8150	0.8108
Random Forest Classifier	0.7556	0.8148	0.7964
Linear Discriminant Analysis	0.7846	0.8130	0.8042
Extra Trees Classifier	0.7535	0.8125	0.7938

Table 2: Classification Results based on 10-fold cross validation and evaluation by F1 macro Score. The feature set comprises the most frequent 100 words without any morphological processing.

Algorithm	Recall	Prec.	F1
CatBoost Classifier	0.8418	0.8757	0.8700
Light Gradient Boosting Machine	0.8322	0.8667	0.8591
Extreme Gradient Boosting	0.8087	0.8479	0.8381
Gradient Boosting Classifier	0.7567	0.8097	0.7999
Random Forest Classifier	0.7365	0.8085	0.7854
Logistic Regression	0.7739	0.8048	0.7961
Ridge Classifier	0.7633	0.7952	0.7875
Extra Trees Classifier	0.7219	0.7893	0.7673

Table 3: Classification Results based on 10-fold cross validation and evaluation by F1 macro Score. The feature set comprises the most frequent 100 function words after morphological processing.

may make a difference in Machine Learning algorithms and may draw our attention to Programmer Invisibility, and that not all implementations are a like.

- *Morphological analysis may not be more useful than just using the top 100 words.* While Arabic is a morphologically rich language and we assumed that untangling this complexity may give us an edge, it turned out that there is almost no difference between the sophisticated function word extraction approach and the simple approach of picking the top 100 most frequent words as features in the classification experiments. The experimental results are given in Tables 2 and 3.

Let's peek a little deeper into these results. We will do so by examining the best classifier (CatBoost) for the word-based experiments (no morphological analysis). As can be seen from Table 4, that performance improves when more features are used.

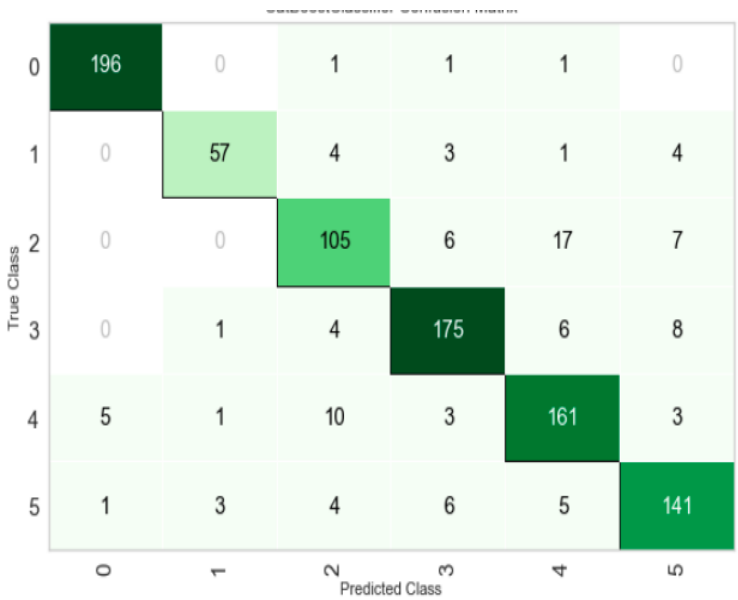

Figure 1: Confusion matrix for the CatBoost Classifier. AbbasHafiz: 0, AbdulfattahAbdalla: 1, HibaGhanim: 2, MustafaFuaad: 3, SaraTahaAllam: 4, SaraYakut: 5

4.2 How many features do we actually need?

We have so far used the top 100 words in our experiments, but this is an arbitrary number. We have also run experiments in which we examined the number of features, from 10 to 300, with an increment of 10 in an effort to determine the value of N in the top N experiments. In these experiments, we decided to accept the highest F1 score regardless of the classification algorithm. Figure 2 shows the performance, as measured in the F1 score, versus the number of features. In all the experiments, the CatBoost classifier was the best performer, with the exception of one experiment, the one with 200 features, where CatBoost came second to XGBoost. The experiments show clearly that the more the features, the better the performance. The straight line also indicates that adding more features will probably improve the results further. While this is true, more features could simply show the effect of the theme, and may not be easily ascribed to author/translator specific style. As we add more vocabulary, more and more thematic lexical items creep into the feature set, and if one translator specialises in scientific texts while another in geography, discipline-specific words may be the discriminator, and not the use of function words, which we know from the literature are more characteristic of individual style. For this reason, we will focus our discussion more on the top 100 words experiments given their stronger alignment with theme-generic and author-specific contexts.

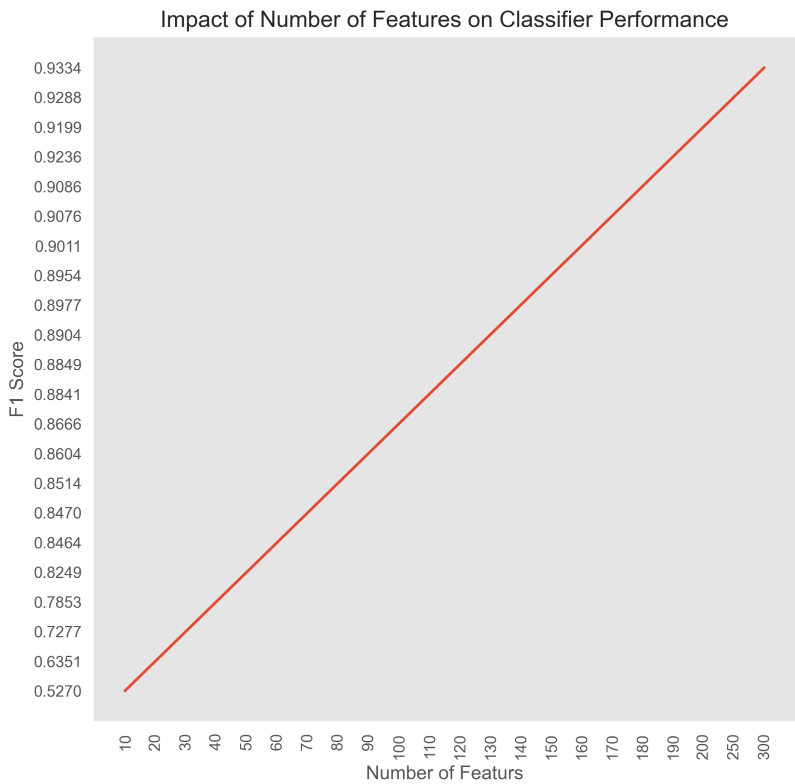

Figure 2: The impact of the number of features on classification accuracy. The higher the number of features, the higher the F1 score, which indicates that adding more features may still lead to better performance.

4.3 Unsupervised Learning Results

Table 5 summarizes the results of the cluster analysis of the top 50 function words for each translator. Since the cluster analysis is applied to the data from each translator separately, we had to restrict the analysis to top 50, instead of top 100, function words to avoid having zero variance features, i.e., columns with all zero entries corresponding to function words never used by translator.

The results in Table 5 compare the number of significant clusters, K, emerging from the top 50 function words for the translator with the number of authors of the translator's text. Clusters with a minimum of 10% of cases were considered significant clusters. Under K-Means clustering, this threshold led to omitting one cluster (size = 0.90%) for Abaas Hafiz, one cluster (6.25%) for Abudlfattah Abdalla, and one cluster (2.36%) for Sara Yaqut. Under Hierarchical clustering, the threshold let to omitting two clusters for Abbas Hafiz (0.90% and 9.82%), one cluster (6.25%)

N	Classifier	F1
10	CatBoost	0.5270
20	CatBoost	0.6351
30	CatBoost	0.7277
40	CatBoost	0.7853
50	CatBoost	0.8249
60	CatBoost	0.8464
70	CatBoost	0.8470
80	CatBoost	0.8514
90	CatBoost	0.8604
100	CatBoost	0.8666

Table 4: Varying the number of N from 10 to 100

for Abudlfattah Abdalla, one cluster (1.55%) for Hiba Ghanim, one cluster (2.79%) for Sara Taha Allam, and two clusters (2.36% and 7.87%) for Sara Yaqut. Under the working assumption that the translator has a detectable style when their translated text groups into fewer clusters than the number of authors of the text and taking the cut off to be 0.50 ratio, two of the six translators, namely, Hiba Ghanim and Sara Taha Allam, appear to have detectable style, i.e., they do not seem to be invisible translators, as indicated by their small clusters-to-authors ratios (0.50 and 0.22 respectively). Translator Mustafa Fuaad also has low clusters-to-authors ratio (0.60). All other three translators have clusters-to-authors ratio of 1.00 or higher. These results are consistent across the two clustering algorithms and are supportive of the supervised learning results described above.

Translator	Authors	K-Means Clustering		Hierarchical Clustering	
	(Books)	K	K/Authors	K	K/Authors
Abbas Hafiz	3 (4)	5	1.67	5	1.67
Abulfattah Abdalla	2 (8)	2	1.00	2	1.00
Hiba Ghanim	6 (7)	3	**0.50**	3	**0.50**
Mustafa Fuaad	5 (6)	3	0.60	3	0.60
Sara Taha Allam	9 (15)	2	**0.22**	2	**0.22**
Sara Yaqut	5 (9)	6	1.20	5	1.00

Table 5: Clustering results using the most frequent 50 function words for each translator after morphological processing.

4.4 Effect of Feature Types

In this study we investigate the effectiveness of different features for the translator attribution task using SVM Linear Kernal algorithm. As can be seen from Table 6, using all word unigrams as features achieved 99% accuracy. On the other hand, concatenating all character-based unigrams, bigrams, trigrams and using them as features achieved 97% accuracy. These results show that word-based features are better than character-based features.

Feature Type	Precision	Recall	F1
words (unigrams)	0.99	0.99	0.99
characters (unigrams)	0.75	0.73	0.73
characters (unigrams + bigrams)	0.93	0.93	0.93
characters (unigrams + bigrams + trigrams)	0.97	0.97	0.97

Table 6: Effectiveness of different types of features for the translator attribution task using SVM Linear Kernel Algorithm.

5 Conclusion

This paper aims at performing translator attribution. We found that translators do leave traces behind on the texts which can be used to differentiate between translators. With a classification accuracy of 87% for six classes, this is extremely higher than a random result of 16.7%, which means each translator has his/her own style, and that Translator Invisibility is truly a myth. Moreover, morphological analysis may not be more useful than just using the top 100 words and that using more features improves the performance of the translator attribution task. Furthermore, tree-based machine learning algorithms are the best performers for the translator attribution task.

It is noteworthy that the clusters-to-authors ratio reported in Table 5 can be used to develop a translator visibility index. However, the development of such index requires larger corpus of translated text and larger number of translators. We plan to pursue this point in our future research and the results shall be reported in a future publication.

ref.bib

References

Baker, Mona (2000). “Towards a methodology for investigating the style of a literary translator.” In: *Target. International Journal of Translation Studies* 12.2, pp. 241–266.

Brown, Keith (2005). *Encyclopedia of language and linguistics*. Vol. 1. Elsevier.

Burrows, John (2002). “‘Delta’: a measure of stylistic difference and a guide to likely authorship.” In: *Literary and linguistic computing* 17.3, pp. 267–287.

Caballero, Christian, Hiram Calvo, and Ildar Batyrshin (2021). “On Explainable Features for Translatorship Attribution: Unveiling the Translator’s Style With Causality.” In: *IEEE Access* 9, pp. 93195–93208.

Covington, Michael A, Iris Potter, and Tony Snodgrass (2015). “Stylometric classification of different translations of the same text into the same language.” In: *Digital Scholarship in the Humanities* 30.3, pp. 322–325.

El-Fiqi, Heba, Eleni Petraki, and Hussein A Abbass (2011). “A computational linguistic approach for the identification of translator stylometry using Arabic-English text.” In: *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*. IEEE, pp. 2039–2045.

— (2019). “Network motifs for translator stylometry identification.” In: *PloS one* 14.2, e0211809.

Forsyth, Richard S and Phoenix WY Lam (2014). “Found in translation: To what extent is authorial discriminability preserved by translators?” In: *Literary and Linguistic Computing* 29.2, pp. 199–217.

Hedegaard, Steffen and Jakob Grue Simonsen (2011). “Lost in translation: Authorship attribution using frame semantics.” In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 65–70.

Kestemont, Mike (Apr. 2014). “Function Words in Authorship Attribution. From Black Magic to Theory?” In: *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 59–66. DOI: `10.3115/v1/W14-0908`. URL: `https://aclanthology.org/W14-0908`.

Lee, Changsoo (2018). “Do language combinations affect translators’ stylistic visibility in translated texts?” In: *Digital Scholarship in the Humanities* 33.3, pp. 592–603.

Lynch, Gerard and Carl Vogel (2018). “The translator’s visibility: Detecting translatorial fingerprints in contemporaneous parallel translations.” In: *Computer Speech & Language* 52, pp. 79–104.

Mikhailov, Mikhail and Miia Villikka (2001). “Is there such a thing as a translator’s style.” In: *Proceedings of Corpus Linguistics*. Citeseer, pp. 378–385.

Mohamed, Emad and Raheem Sarwar (2021). “Linguistic features evaluation for hadith authenticity through automatic machine learning.” In: *Digital Scholarship in the Humanities*.

Mosteller, Frederick and David L. Wallace (1963). “Inference in an Authorship Problem.” In: *Journal of the American Statistical Association* 58.302, pp. 275–309. ISSN: 01621459. URL: `http://www.jstor.org/stable/2283270`.

Rodriguez, M. Z. et al. (2019). “Clustering algorithms: A comparative approach.” In: *PLoS ONE* 14.1, pp. 1–34.

Rybicki, Jan et al. (2010). “The Translator’s Wife’s Traces: Alma Cardell Curtin and Jeremiah Curtin.” In: *Przekładaniec* 2, pp. 89–109.

Rybicki, Jan (2012). “The great mystery of the (almost) invisible translator.” In: *Quantitative Methods in Corpus-Based Translation Studies: A practical guide to descriptive translation research* 231.

Rybicki, Jan and Magda Heydel (2013). “The stylistics and stylometry of collaborative translation: Woolf’s Night and Day in Polish.” In: *Literary and Linguistic Computing* 28.4, pp. 708–717.

Venuti, Lawrence (1986). “The translator’s invisibility.” In: *Criticism* 28.2, pp. 179–212.