Authorship Attribution of Modern Standard Arabic Short Texts

YARA ABUHAMMAD[*], Department of Electrical and Computer Engineering, Birzeit University, Palestine

YARA ADDABE', Department of Electrical and Computer Engineering, Birzeit University, Palestine

NATALY AYYAD, Department of Electrical and Computer Engineering, Birzeit University, Palestine

ADNAN YAHYA, Department of Electrical and Computer Engineering, Birzeit University, Palestine

Text data, including short texts, constitute a major share of web content. The availability of this data to billions of users triggers frequent plagiarism attacks. Authorship Attribution (AA) seeks to identify the most probable author of a given text based on similarity to the writing style of potential authors. In this paper, we approach AA as a writing style profile generation process, where we group text instances for each author into a single profile. We use Twitter as the source for our short Modern Standard Arabic (MSA) texts. Numerous experiments with various training approaches, tools and features allowed us to settle on a text representation method that relies on text concatenation of Arabic tweets to form chunks, which are then duplicated to reach a precalculated length. These chunks are used to train machine learning models for our 45 author profiles. This allowed us to achieve accuracies up to 99%, which compares favorably with the best results reported in the literature.

CCS Concepts: • **Computing methodologies → Machine learning**; **Artificial intelligence**; **Natural language processing**; • **Human-centered computing → Visualization toolkits**; • **Applied computing → Investigation techniques**; • **Information systems → Information retrieval**.

Additional Key Words and Phrases: Arabic Authorship Attribution, Support Vector Classification, Machine Learning, Social Media, Arabic Tweets, Authorship Attribution Datasets

ACM Reference Format:

Yara Abuhammad, Yara Addabe', Nataly Ayyad, and Adnan Yahya. 2021. Authorship Attribution of Modern Standard Arabic Short Texts. In *Proceedings of the 2021 Arab Women In Computing Conference (ArabWIC'21), August 25, 2021, Sharjah, UAE.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Social media in general, and Twitter in particular, have grown massively in the past few years so that the number of tweets per minute has reached 480k in 2020[1]. The influence of public figures with many followers made their content prone to plagiarism and literary theft. Thus, methods to attribute a text to its most probable author acquired added importance and the concept of Authorship Attribution for short texts was brought to life.

This paper focuses on short MSA texts. Arabic is the fourth most used language on the internet[7]. Still, the number of studies on Arabic Authorship Attribution (AAA) is low compared to those for English, more so for short Arabic texts, despite the widespread use of Twitter in Arab countries. Twitter was chosen as our source of data. We limited our attention to MSA so that the results will be applicable to different Arab countries and because Natural Language Processing (NLP) tools are more readily available for MSA.

[*] Corresponding authors: yara.firas.abuhammad@gmail.com

Our focus on short MSA text comes with great challenges. Many techniques developed for English are not applicable for Arabic due to Arabic characteristics like diacritics, inflection, and elongation. Also, collecting data in MSA that varies in topic and for a large enough number for verified authors was not an easy task. In addition, dealing with short texts, such as tweets means having few words/sentences to extract patterns and style elements from.

We experimented with multiple approaches/methods for text preprocessing, data representation and Machine Learning (ML) classifiers and adopted the best-performing candidates in each case.

The rest of the paper is organized as follows: Section 2 presents a review of AA work on non-Arabic and Arabic short texts. Section 3 describes the classification methodology. In Section 4, we describe our experiments along with their results and observations. Finally, Section 5 concludes the paper and points to possible extensions.

2 RELATED WORK

Many studies highly praised using n-grams as a technique for capturing content and stylistic information in English short texts[10, 15]. Ishihara[11] did a study on SMS messages where different datasets for 228 authors were used. Their model achieved 80% accuracy when using word n-grams and chunking. Layton et al.[12], worked on tweets of 140 characters or less. They preprocessed tweets by removing hashtags and mentions. They used the Source Code Authorship Profile (SCAP) methodology, usually used to identify the author of a given source code. They reported an accuracy of over 70% for 50 authors. Saha et al.[14] also worked on English tweets of up to 140 characters for a total of 20 authors and collected 400 tweets per author. They used linguistic and Twitter-specific features. In the latter, the number of hashtags, mentions and URLs over the number of words were calculated. Their Multilayer Perceptron (MLP) model achieved 96.44% accuracy for 4 authors. However, as the number of authors increased, the accuracy gradually decreased, with 67% for 20 authors. They also showed that increasing the number of tweets per author had little effect on accuracy. Belvisi et al.[8] worked in the domain of criminology with 280 character tweets and 40 authors. They removed elements like URLs and trending tags. Stylometric Features (SF) alongside word and character n-grams were used as features. The classification accuracy ranged between 92% and 98.5%.

As for short Arabic texts, Howedi et al.[9] worked with short historical Arabic texts for 10 authors with 3 random pages per author and K-Nearest Neighbors (KNN) classifier. They preprocessed texts by tokenization, punctuation removal and normalization. For features, they used presence of rare words appearing once or twice in the corpus and character n-grams with n = 1,2,3,4. The highest accuracy was 90.42% for 5-NN and 4-grams. Al-Sarem et al.[6] tested different classifiers on 4631 Arabic religious fatwas. For feature extraction, they combined Bag Of Words (BOW) with SF approaches. The highest accuracy reported was 86.39% using NB for 15 authors. Rabab'ah et al.[13], collected 38,386 Arabic tweets for 12 authors. They used features from the morphological analysis plus SF and BOW features (top 100 most common words). The highest accuracy reached was 68.67% using Support Vector Machine (SVM) classifier on all features combined. Al-Falahi et al.[5] worked on 21,929 poems of a total of 114 poets from different eras. Preprocessing included: normalization, stemming, tokenization and stopword removal. Lexical features, character features, structural features, poetry features, syntactic features, semantic features and specific words per authors features were utilized. The maximum accuracy achieved was 99.12% by applying Linear Discriminant Analysis (LDA)[1]. AlTakrori et al.[7] conducted an intensive study in which they investigated the performance of various classification techniques and feature combinations of Arabic tweets for up to 20 authors. The best performance for 20 authors was approximately 52%. Table 1 compares the best results of our work with the results of the previously mentioned works on AAA.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html

Table 1: Comparison of This Work's Authorship Attribution Method to Other Methods on Short Arabic Texts

Reference	Text Type	Attribution Method	No. of Authors	Highest Accuracy
This work	Tweets (280 characters)	Support Vector Classification (SVC)	45	99.24%
Howedi et al.[9]	3 Pages of text	KNN	10	90.42%
Al-Sarem et al.[6]	Religious fatwas	Naïve Bayes (NB)	15	86.39%
Al-Falahi et al.[5]	Poems	LDA	114	99.12%
Altakrori et al.[7]	Tweets (140 characters)	Random Forest (RF)	20	52%
Rabab'ah et al.[13]	Tweets (140 characters)	SVM	12	68.67%

3 METHODOLOGY

In this section, we describe our dataset and the text preprocessing steps. Then, we describe the classification methodology and data representation used for Authorship Attribution of MSA tweets.

3.1 Data collection

To access Twitter and scrape tweets, Tweepy[2] was used. It is an open source Python package that facilitates access to the Twitter API. Tweepy was utilized to retrieve the tweets for each author in a manually created list of authors who tweet in MSA only. For each tweet, its text, Identification (ID) number and publication date were collected.

We first removed retweets (tweets having "RT @"), as they are written by another author. We also removed replies (tweets containing "@"), since they contain information about authors rather than their writing style[12]. Emojis[3], hyperlinks, and English hashtags were filtered out. Next, numbers were normalized to Arabic numbers. Language detection tools[4] were used to ensure that the majority of the tweet is written in Arabic. Finally, we filtered out English and other non-Arabic characters. Any resulting empty tweets were discarded. To avoid text that may reveal author identity, for each author, tweets containing the author name were removed and hashtags were replaced with '#'.

Our final dataset has 45 authors[4]. The tweets varied in topic: 11% were in Religion, 35.5% were in Politics, 28.8% were in Arabic Literature, and 24.7% were in Journalism. 13 Arab countries are represented in the dataset. The number of tweets for the authors ranged from 723 to 3,090 as shown in figure 1.

Table 2: Dataset Statistics

No. of authors	45
Total no. of tweets	71,397
Total no. of words	1,909,081
Total no. of hashtags	43,636
Average no. of tweets per author	1,587
Average no. of words per tweet	26.7
Average no. of hashtags per tweet	0.6

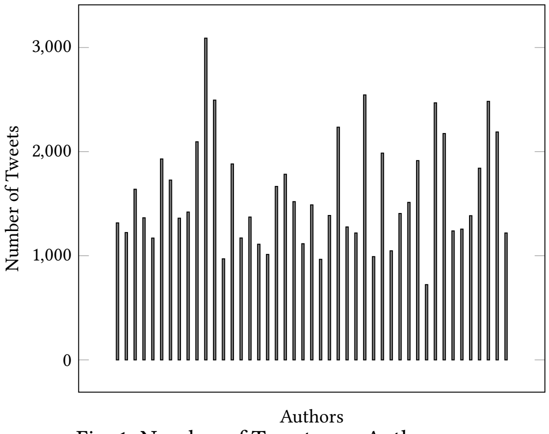


Fig. 1: Number of Tweets per Author

[2] http://docs.tweepy.org/en/latest/

[3] https://pypi.org/project/emoji/

[4] https://pypi.org/project/langdetect/ & https://pypi.org/project/guess_language-spirit/

3.2 Preprocessing

After cleaning as described above, each tweet was processed as follows:

- **Diacritic and Punctuation Marks Removal:** In Authorship Attribution, diacritics have negligible effect[7]. So, they were removed along with punctuation marks.
- **Lemmatization:** Returns the root (i.e. lemma) of a given word. For example, the words, "مدرسة", "دراسة" and "مُدَرِّس", all have the same lemma, "درس". This process tones down the complexity and variety of word forms.
- **Stopword Removal:** Stopwords are words common in a language such as, prepositions, pronouns and conjunction words. They are found in almost every text regardless of the author and are removed to grant more significance to words that define the meaning of the text. We removed about 400 stopwords from our dataset[5].
- **Normalization:** this step transforms text elements to a single canonical form. In our case, hamza in all of its variations "آ، أ ، إ" becomes an alef "ا". Taa marboota "ة ، ـة" becomes "ه ، ـه".
- **NLP Tool of Choice:** We considered several NLP tools, but settled for FARASA[2]. FARASA is a fast Arabic NLP tool with an Application Programmable Interface (API) that can be utilized to create a Java application that suits our needs. FARASA consists of several modules, including a segmentation module, a Part-of-speech (POS) tagger, an Arabic text diacritizer and a dependency parser. We used FARASA segmentation module to tokenize text. The text was then lemmatized, normalized, and diacritics were removed.

4 AUTHORSHIP ATTRIBUTION AND EXPERIMENTS

We employed ML to build the best performing and most accurate Authorship Attribution model for MSA tweets. This was done using scikit-learn[6], an ML library implemented in Python. Scikit-learn is easy to use and provides a plethora of classification, clustering and regression algorithms including RF, NB, KNN and SVM. Scikit-learn website provides a cheat sheet that helps choosing the most appropriate ML classifier[7].

Once data is preprocessed as elaborated previously, we applied vectorization that transforms text into a fixed length vector representation, that ML classification algorithms can operate on. In our work, we used the Term Frequency - Inverse Document Frequency (TF-IDF) Vectorizer[8]: Count Vectorizer (CV) followed by TF-IDF Transformer.

4.1 Feature Selection and Experiments

Multiple experiments were conducted in order to choose the features that produce the model with the highest accuracy for AA for short MSA texts. We experimented with: n-grams, both character and word, POS tagging, removing N most frequent words, removing N least frequent words, frequency replacement where we replaced character n-grams with their frequencies and concatenating series of texts to form chunks of varying lengths. In some experiments we used combinations of text representations. Some of these experiments were conducted using 11 candidate authors (section 4.2). In the process of improving our system, we increased the number of candidate authors to 45 (section 4.3).

4.2 Initial Single Tweet Experiments

In this section, our text unit was a single tweet represented by its character n-grams or word n-grams: n was limited to [1:4] since many studies showed this range to be most effective[10, 15]. More than 120 experiments were conducted for 11 authors and 17,237 tweets. The training to testing split adopted is 80%:20% for 11 authors.

[5] https://www.analyticsvidhya.com/blog/2019/08/how-to-remove-stopwords-text-normalization-nltk-spacy-gensim-python/
[6] https://scikit-learn.org/stable/
[7] https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html
[8] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Most Frequent Character n-grams in the Corpus. Here, the frequency for each character n-grams was calculated. Then a list was made with all the character n-grams and their frequencies. The list is sorted in a descending order based on the frequency. The first M entries in that list (M most frequent character n-grams) were kept in the corpus, and all occurrences of any other character n-gram that wasn't in the top M list were omitted. M ranged from 100 to 500 with an increment of 100. The results obtained were not promising.

Part-of-speech Tagging. In this set of experiments, words were represented by their POS tags. The Stanford Log-linear POS Tagger was used. The resulting accuracies were not significantly higher than the baseline value (1/11).

Term Frequency Based Experiments. Here, our dataset was represented by its word unigrams, bigrams and trigrams. The frequency of each word n-gram in the entire corpus was calculated. Three lists of (word n-gram, frequency) were created for n=1,2,3. The lists were sorted in a descending order based on the frequency. Table 3 can be read as follow, the number of word uni-grams without repetition is 17,260, and the most frequent uni-gram has 2,386 occurrences in the corpus. We conducted three sets of experiments. The reported results are achieved using SVC model.

Table 3: Statistics of Word N-grams Frequencies Across the Corpus

	Count of Unique Word n-grmas	Highest Frequency Value
n = 1	17,260	2,386
n = 2	162,437	461
n = 3	182,370	144

1. **Removing M Most Frequent Word n-grams.** In these experiments, we removed all occurrences of the corpus M most frequent word n-grams. When M = 100, for example, we omitted occurrences of the 100 most frequent word n-grams in the corpus. We experimented with M=100 to 10,000 with 100 increments. 76.21%, 71.66%, 71.31% are the highest accuracies achieved for word unigrams, bigrams and trigrams, respectively.
2. **Removing M Least Frequent Word n-grams.** In these experiments, we removed occurrences of the corpus M least frequent word n-grams. For example when M = 100, n=3 we removed occurrences of the 100 least frequent word 3-grams. We experimented with N=100 to 10,000 with 100 increments. 82.94%, 77.89%, 76.21% were the highest accuracies achieved for unigrams, bigrams and trigrams, respectively.
3. **Removing Word n-grams With a Predefined Frequency Range [1,K].** Here, we omit all occurrences of word n-grams with frequency values less than or equal K. K ranged from 1 to 50 with increment of 1. For example, when K = 3, all word n-grams that occurred one, two or 3 times in the corpus were removed. 82.94%, 64.86%, 28.47% are the highest accuracies achieved for unigrams, bigrams and trigrams, respectively.

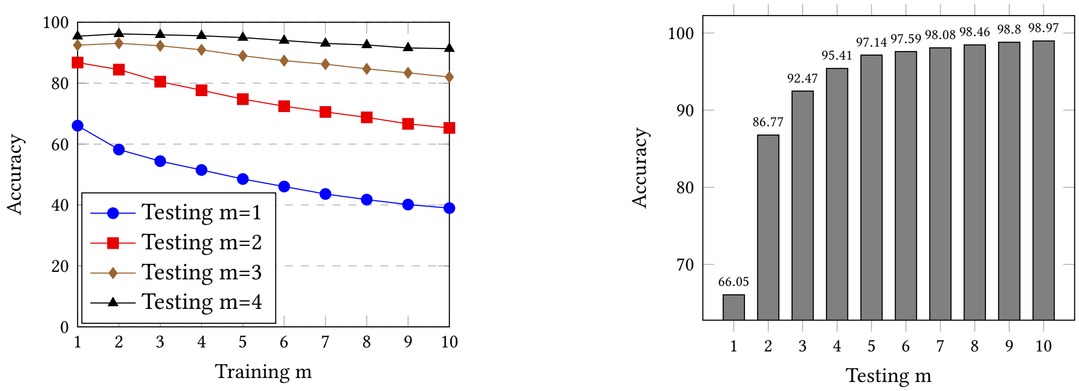
4.3 Non-overlapping Concatenation

Given that single tweet experiments resulted in poor performance and to mitigate the effect of short texts, we decided to introduce the concatenation method and study its effect on the performance of Authorship Attribution models. This method essentially groups m number of tweets into one text unit resulting in a longer unit to conduct experiments on. Our dataset here consists of 71,397 tweets for 45 authors. The concatenation parameter m is an integer used to determine the number of consecutive tweets joined together to form a text chunk, with and without overlapping. First, we consider non-overlapping concatenation. For example, for m = 2 and a corpus consisting of 5 tweets *T = < T1, T2, T3, T4, T5 >, the resulting chunks are Tc = {{T1+T2},{T3+T4},{T5}}, where T1+T2 is the result of appending T2 to T1.* For each resulting file of concatenated tweets for an author, the average length (*L*) of the generated chunks was calculated.

Concatenation: Performance Evaluation. Our dataset was split into two parts, 80% for training and 20% for testing.

1. **Training process:** We trained 10 models corresponding to 10 values of the concatenation parameter m, on 80% of our dataset. The first model was trained using single tweets, the second using chunks of two tweets and so on. Our classifier of choice was SVC with TF-IDF Vectorizer. Each trained model was stored with its TF-IDF Vectorizer instance for later use in external testing using the testing file that had 20% of our dataset.
2. **Testing process:** The testing set was subjected to concatenation for $1 \leq m \leq 10$ to generate 10 test files for a total of 100 tests. While each test file was being fed to each model for evaluation, duplication was done for each test chunk of length less than the average length (L) of the training chunks. The shorter chunks were duplicated to be comparable to (L). So, Concatenation is used for model training and testing. However, the concatenation parameter can be different for training and testing. Thus we can talk about "training m" and "testing m".

Experiments. The model behavior per each value of testing m was interesting in that the accuracy slightly decreased as training m increased. Figure 2a shows the behavior when testing m ranges from 1 to 4 for various training m values. Overall, the best performance (attribution accuracy) was recorded for the model trained using single tweets. Figure 2b depicts the performance of that model on testing m = 1 to 10 for training m=1. We can note that as testing m increases, the performance significantly improves.



(a) Concatenation Model Accuracy for Testing $m \leq 4$

(b) Model Accuracy for Training m=1 vs. Testing m

Fig. 2: Non-overlapping Concatenation Performance Evaluation

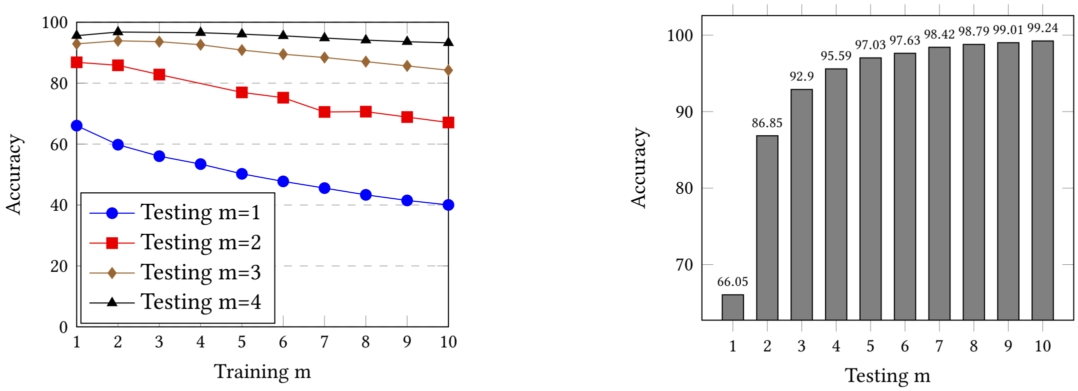
4.4 Shifting Window (Overlapping) Concatenation

In the shifting window method the idea is to concatenate tweets in an overlapping fashion: if the concatenation parameter m = 2 and we have the sequence of tweets T = <T1, T2, T3, T4>, the shifting window output would be the chunks *{{T1+T2}, {T2+T3} and {T3+T4}}*.

We followed the same procedure as in the non-overlapping experiments. Here too, the accuracy slightly dropped as training m increased. The best performance was achieved for training m=1. Figure 3a depicts model behavior when testing m ranges from 1 to 4. In general, the shifting window concept achieved better accuracy as compared to the non-overlapping case. Figure 3b depicts the accuracies recorded for the model that was trained using training m=1.

5 CONCLUSION AND FUTURE WORK

We experimented with various techniques for Authorship Attribution of Arabic tweets. These experiments guided the choice of the best preprocessing tools, data representation and ML algorithms and model parameters. The combination



(a) Accuracy for Testing n = 1 to 4

(b) Model Accuracy for Training m=1 vs. Testing m

Fig. 3: Shifting Window Concatenation Performance Evaluation

of representing tweets by their unigrams, concatenating tweets using the shifting window technique at test time, transforming text using TF-IDF Vectorizer and the SVC model achieved an accuracy of up to 99.24%. This compares favorably with the best results reported in the literature (see table 1). The dataset we used is being made available through our institutional repository[4]. We also created the "Kottabi" mobile application for both Android and iOS[3]. To the best of our knowledge, this is the first attempt to create a mobile application specializing in AAA. Future work plans include experimenting with random chunking and using deep learning algorithms, and investigating the possibility of applying our techniques to cross-lingual texts (Arabic and English).

REFERENCES

[1] 2020. How much data is on the internet? The Big Data Facts Update 2020. https://www.nodegraph.se/how-much-data-is-on-the-internet/

[2] Abdelali, A., Darwish, K., Durrani, N., and Mubarak, H. 2016. Farasa: A Fast and Furious Segmenter for Arabic.. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, 2016.* https://doi.org/10.18653/v1/n16-3003

[3] Abu Hammad, Y., Addabe, A., and Ayyad, N. 2021. Kottabi. https://youtu.be/BGHxAxJGpTc

[4] Addabe, A., Abu Hammad, Y., Ayyad, N., and Yahya, A. 2021. A Dataset for Authorship Analysis of Short Modern Arabic Text. In *FADA: Birzeit University Institutional Repository Dataset Collection.* https://fada.birzeit.edu/handle/20.500.11889/6743

[5] Al-falahi, A., Ramdani, M., and Bellafkih, M. 2019. Arabic Poetry Authorship Attribution using Machine Learning Techniques. *Journal of Computer Science* (07 2019), 1012–1021.

[6] Al-Sarem, M., Emara, A., and Abdel Wahab, A. 2020. Performance of Authorship Attribution Classifiers With Short Texts: Application of Religious Arabic Fatwas. *International Journal of Data Mining, Modelling and Management* 12, 3 (01 2020), 350–364.

[7] Altakrori, M., Iqbal, F., Fung, B., Ding, S., and Tubaishat, A. 2018. Arabic Authorship Attribution: An Extensive Study on Twitter Posts. *ACM Transactions on Asian and Low-Resource Language Information Processing* 18, 1 (11 2018), 1–51.

[8] Belvisi, N., Muhammad, N., and Alonso-Fernandez, F. 2020. Forensic Authorship Analysis of Microblogging Texts Using N-Grams and Stylometric Features. *2020 8th International Workshop on Biometrics and Forensics (IWBF)* (03 2020), 1–6.

[9] Howedi, F., Mohd, M., Aborawi, Z., and Jowan, S. 2020. Authorship Attribution of Short Historical Arabic Texts using Stylometric Features and a KNN Classifier with Limited Training Data. *Journal of Computer Science* 16 (10 2020), 1334–1345.

[10] Ioannis, K. and S. Efstathios. 2011. Author Identification Using Semi-supervised Learning - Notebook for PAN at CLEF 2011., Vol. 1177. 19–22.

[11] Ishihara, S. 2012. A Forensic Authorship Classification in SMS Messages: A Likelihood Ratio Based Approach Using N-gram. *Proceedings of the Australasian Language Technology Workshop* (05 2012).

[12] Layton, R., Watters, P., and Dazeley, R. 2010). Authorship Attribution for Twitter in 140 Characters or Less. In *2010 Second Cybercrime and Trustworthy Computing Workshop.* 1–8.

[13] Rabab'ah, A., Al-Ayyoub, M., Jararweh, Y., and Aldwairi, M. 2016. Authorship Attribution of Arabic tweets. In *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA).* 1–6.

[14] Saha, N., Das, P., and Saha, H. 2017. Authorship Attribution of Short Texts using Multi Layer Perceptron. *International Journal of Applied Pattern Recognition* 5 (09 2017), 251–259.

[15] Schwartz, R., Tsur, O., Rappoport, A., and M. Koppel. 2013. Authorship Attribution of Micro-Messages. *Association for Computational Linguistics* (10 2013), 1880–1891.