Arabic Authorship Attribution: An Extensive Study on Twitter Posts

MALIK H. ALTAKRORI, School of Computer Science, McGill University, Canada
FARKHUND IQBAL, College of Technological Innovation, Zayed University, United Arab Emirates
BENJAMIN C. M. FUNG and STEVEN H. H. DING, School of Information Studies, McGill University, Canada
ABDALLAH TUBAISHAT, College of Technological Innovation, Zayed University, United Arab Emirates

Law enforcement faces problems in tracing the true identity of offenders in cybercrime investigations. Most offenders mask their true identity, impersonate people of high authority, or use identity deception and obfuscation tactics to avoid detection and traceability. To address the problem of anonymity, authorship analysis is used to identify individuals by their writing styles without knowing their actual identities. Most authorship studies are dedicated to English due to its widespread use over the Internet, but recent cyber-attacks such as the distribution of Stuxnet indicate that Internet crimes are not limited to a certain community, language, culture, ideology, or ethnicity. To effectively investigate cybercrime and to address the problem of anonymity in online communication, there is a pressing need to study authorship analysis of languages such as Arabic, Chinese, Turkish, and so on. Arabic, the focus of this study, is the fourth most widely used language on the Internet. This study investigates authorship of Arabic discourse/text, especially tiny text, Twitter posts. We benchmark the performance of a profile-based approach that uses *n*-grams as features and compare it with state-of-the-art instance-based classification techniques. Then we adapt an event-visualization tool that is developed for English to accommodate both Arabic and English languages and visualize the result of the attribution evidence. In addition, we investigate the relative effect of the training set, the length of tweets, and the number of authors on authorship classification accuracy. Finally, we show that diacritics have an insignificant effect on the attribution process and part-of-speech tags are less effective than character-level and word-level *n*-grams.

CCS Concepts: • **Human-centered computing** → **Visualization toolkits**; • **Computing methodologies** → **Language resources**; **Supervised learning by classification**; Classification and regression trees; Support vector machines; • **Information systems** → *Content analysis and feature selection*; *Information extraction*; • **Networks** → *Online social networks*; • **Applied computing** → Investigation techniques; Evidence collection, storage and analysis;

Additional Key Words and Phrases: Authorship attribution, visualization, short text, social media, twitter

5

The article is partially supported by the Canada Research Chairs (CRC) Program (950-230623), the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grants (RGPIN-2018-03872), and the Zayed University Research Incentive Fund (RIF) (R14025 and R13059).
Authors' addresses: M. H. Altakrori, School of Computer Science, McGill University, 3480 Rue University, Montréal, QC, H3A 2A7, Canada; email: malik.altakrori@mail.mcgill.ca; F. Iqbal and A. Tubaishat, College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates; emails: farkhund.Iqbal@zu.ac.ae, abdallah.Tubaishat@zu.ac.ae; B. C. M. Fung (corresponding author) and S. H. H. Ding, School of Information Studies, McGill University, 3661 Peel St, Montréal, QC, H3A 1X1, Canada; emails: ben.fung@mcgill.ca, steven.h.ding@mail.mcgill.ca.

https://doi.org/10.1145/3236391

ACM Reference format:
Malik H. Altakrori, Farkhund Iqbal, Benjamin C. M. Fung, Steven H. H. Ding, and Abdallah Tubaishat. 2018. Arabic Authorship Attribution: An Extensive Study on Twitter Posts. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 18, 1, Article 5 (November 2018), 51 pages.
https://doi.org/10.1145/3236391

1 INTRODUCTION

Criminals are exploiting the anonymous nature of the Internet to covertly commit crimes online. These perpetrators create fake IDs while conducting illegitimate, malicious social media communications or attacking online e-commerce systems such as eBay and Amazon. Authorship analysis techniques have been successful [17, 29] in defending users against such attacks and addressing the issue of anonymity without sacrificing the privacy of Internet users.

Authorship attribution helps identify the original author of a given anonymous text by extracting and analyzing the author-specific writing style features [65]. Initially, authorship attribution was used in the field of literature to identify the original authors of novels, plays, or poems [32, 36, 65]. Later, its applications have been extended to forensics by investigating the true authors of malicious texts and/or using the analysis results as evidence in courts of law [14]. Various computational techniques have been proposed for different languages and types of text such as Twitter posts, Facebook status, Short Message Service (SMS) messages, or chat conversations.

Compared to the large body of authorship attribution research for popular languages such as English [20, 43, 44] and Chinese [33, 70], only around 10 studies are dedicated to Arabic authorship analysis [2, 6, 7, 37, 50, 59]. However, Arabic is the fourth-most popular language used over the Internet after English, Chinese, and Spanish (see Figure 1). It accounts for 4.8% of the total use. The research literature shows that researchers mostly direct their efforts toward English, with few works being done on other languages. This is attributed to the various challenges that face researchers when they analyze an Arabic document; techniques developed for English may not be directly applicable to other languages. As noted in Abbasi and Chen [2], Arabic has three special characteristics, namely word length and elongation, inflection, and diacritics. These characteristics prevent the direct application of English authorship analysis techniques because of their effect on the feature extraction process.

Moreover, among these few relevant works on Arabic, none focus on Arabic short text such as social media posts, chat logs, and emails. Cybercrime investigators must frequently deal with these kinds of short texts. Existing Arabic authorship studies assume that there are plenty of text data for authorship analysis. Specialized methods have been proposed for novels [37], books [6, 50, 59], articles [7], and the combination of forum messages [2]. However, this may not be the case in real-life applications. Forensic investigators may have only a limited set of samples to be analyzed, especially for cybercrime investigation. Authorship attribution over short text is more challenging due to the limited information carried by the text. Even for English, the length of the text sample has a significant impact on the attribution result. Specialized techniques are needed for the short text scenario [29].

Furthermore, little focus was given to visualizing the results beyond providing an accuracy and a confidence value. For English, only three techniques [3, 9, 35] have been proposed for this purpose. For Arabic, no techniques have been adopted or proposed. Classification techniques that are known to produce the best results are mostly black-box methods with complicated computations. As an example, Support Vector Machines (SVM) and Random Forests are known to produce good accuracy, yet the results are difficult to interpret or explain. Law enforcement agents use the skills of an expert witness, such as an authorship attribution expert or a linguistics expert. The role of

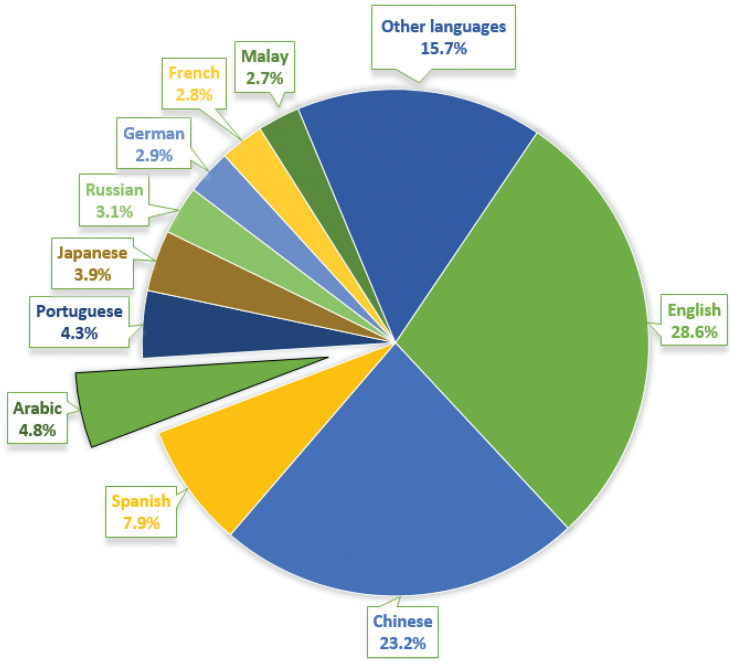


Fig. 1. Top 10 languages used over the Internet according to a study published by Miniwatts Marketing Group [47].

these experts is to help law enforcement officers narrow down the number of potential suspects or provide evidence to justify a conclusion in a court of law. For experts to perform their role properly, they have to find the most plausible author from the candidate authors and show how they reached their conclusion in a clear and presentable way. To do so, they must use a classification model that provides high accuracy as well as being easy to present and explain, as opposed to using a model that is vague or complex, even if they have to sacrifice the performance by a bit.

In this article, we address the aforementioned research gap by focusing on the Arabic short text from the social media platform Twitter.[1] We customize and apply existing Arabic authorship attribution techniques on Twitter data. We also adopt an English visualizable attribution technique based on *n*-gram to Arabic. We compare their performance on Twitter posts and report our findings. To our best knowledge, this is the first work reporting the performance of authorship attribution on short Arabic text in general. Moreover, this is the first work adopting a visualizable approach for Arabic authorship attribution.

1.1 Problem Statement

We first provide an informal description of the authorship attribution problem on short text, followed by a formal problem definition. Given a set of candidate authors of an anonymous, relatively short, text, and a set of sample writings for each one of the candidate authors, an authorship attribution expert analyzes the anonymous text and the sample writings of each author to capture the writing styles of the anonymous text as well as the sample writings of each candidate author. Based on that, the authorship attribution expert identifies the most plausible author of the anonymous

[1] www.twitter.com.

text as the author whose writing style has the highest similarity to the writing style captured from the anonymous text. As mentioned earlier, this work addresses the authorship attribution problem in Arabic text, specifically, Arabic tweets. However, for brevity, we drop the word "Arabic" from "Arabic writing samples" and "Arabic tweets." Furthermore, we use the terms "a writing sample" and "tweet" interchangeably throughout the article to refer to the same unit of text.

Formally, let $C = \{C_1, \ldots, C_n\}$ be the set of candidate authors of an Arabic anonymous text a and W_i be a relatively large collection of sample writings that belong to candidate author $C_i \in C$. Finally, let $f(a, W_i)$ be a function that computes the similarity between the writing styles of the anonymous text a and each sample writing in W_i. The *problem of authorship attribution* is to identify the most plausible author C_i from the set of candidate authors C, where $f(a, W_i) > f(a, W_j)$ $\forall\, C_i, C_j \in C$, and $C_i \neq C_j$.

1.2 Research Questions

This study adopts and benchmarks the profile-based n-gram approach and the instance-based approach to address the problem of authorship attribution in Arabic short text from Twitter. Specifically, we answer the following questions in this article.

(1) *How does the n-gram approach perform compared to state-of-the-art instance-based classification techniques under varying attribution scenarios?* A typical authorship attribution problem has three factors that affect the performance, namely, the number of candidate authors, the number of writing samples per candidate author, and the length of the writing samples and the anonymous text under investigation. We benchmark the state-of-the-art instance-based classification techniques, such as Naïve Bayes (NB), SVM, Decision Trees, and Random Forests (RF), with stylometric features. Then we compare their performance with the n-gram approach under varying attribution factors.
(2) *Which n-gram level (character, word, or syntactic) is the most helpful in distinguishing the authors' writing styles?* The use of the n-gram approach in authorship analysis has not been studied for Arabic short text. In this article, we will investigate the performance of the n-gram approach on the characters, word, and syntactic levels. For the syntactic level, we use part-of-speech (POS) n-grams.
(3) *How important are diacritics to the attribution process when the n-gram approach is used?* Diacritics in Arabic appear on the character level, and they are optional. The fact that their presence with a word could change its meaning is one of the morphological properties that make Arabic different from English. We will compare the performance of the attribution process using n-grams before and after removing the diacritics from the text.
(4) *When using instance-based classification techniques, how important is it to use all three categories of stylometric features?* There are three categories of stylometric features: lexical, structural, and syntactic. They have been intensively studied for authorship analysis, especially in English. For the sake of completeness, we investigate the importance of each category for the attribution process in Arabic.

1.3 Contribution

The contribution of this work is summarized as follows:

- *Providing the first extensive authorship study on Arabic tweets.* To the best of our knowledge, this is the first work that investigates authorship attribution on Arabic short text in general and Arabic tweets in specific. We conduct a systematic evaluation for various attribution

techniques on Arabic tweets and make the dataset publicly available.[2] Not only does this open the door for further investigation of other authorship attribution techniques to be used for Arabic but can also serve as a benchmark of experimental results for future work.

- *Providing an interactive system to visualize the attribution process and results.* Our work on Arabic authorship analysis is a significant extension of Ref. [19], which supports only English. One main application of authorship attribution techniques is to use the results and analysis as evidence in the context of criminal investigation; therefore, the interpretability of the results is as important as high accuracy. We have adopted the original models that were developed specifically for English and proposed using a language detection tool to automate selecting between Arabic and English; we added Stanford's part-of-speech tagger for Arabic and we changed the forms' orientation to show English or Arabic. With these modifications, the tool is able to visualize the authorship attribution evidence for the Arabic language in an intuitive and convincing style.

The rest of the article is organized as follows: Section 2 presents a comprehensive review of authorship attribution work on Arabic in general and short English text. Section 3 describes the two approaches to perform authorship attribution: an instance-based approach and a profile-based approach. In Section 4, we describe our experimental design, followed by the results and discussion in Section 5. Finally, Section 6 concludes the article.

2 LITERATURE REVIEW

This research targets authorship attribution for short Arabic messages, specifically, Arabic tweets. Although Arabic is widely used over the Internet [59], literature shows that authorship attribution research focuses on English, while research on Arabic is scarce. To the best of our knowledge, none of the available work on Arabic targets short text. The term "short" was formerly used to describe one page or a blog post; currently, the term is used to describe much shorter forms of text, e.g., a Facebook status, a Twitter post, a SMS message, or an Instant chat Message (IM).

We start by reviewing the techniques used for authorship attribution on non-Arabic short text and then review the work done on Arabic text, regardless of the text length. While doing this, we keep in mind the difference between Arabic and English as detailed in Reference [2]. According to Abbasi and Chen [2], authorship attribution techniques developed for English cannot be directly applied to Arabic text without modifications, since Arabic and English languages have different language properties.

2.1 Authorship Attribution on Non-Arabic Short Text

Authorship attribution on non-Arabic short text has addressed different social media, e.g., SMS messages, chat logs, and Twitter posts. In reviewing the work on these topics, we look at the features and the classification techniques.

Chat Logs. One area in which authorship attribution on short text is investigated is chat conversations, such as Internet Relay Chat (IRC) and online chat rooms. Examples of the work done on these domains are References [28] and [38].

Inches et al. [28] used word-level n-grams as features and statistical modeling, namely X^2 and Kullback-Leibler divergence, to compute the similarity between a candidate author and a query text. They started by generating an author profile for each author, which they did in two steps.

[2]Due to Twitter regulations for developers [67], we cannot explicitly share the actual text for each tweet. Instead, a list of Ids and a Python script to crawl the tweets are provided via http://dmas.lab.mcgill.ca/data/Arabic_Twitter_dataset.zip. This seems to be a common practice in the research community [66].

The first step is concatenating all the text generated by a single user into one document. The next step divides the text into vectors by using "*non-letter* characters" as tokens or stop marks. When a query text is introduced, a profile is created for it, and its profile is compared to all the authors' profiles to find the most similar one.

Layton et al. [38] used three similar techniques, namely Common *n*-grams (CNG) [33], Source Code Author Profiles (SCAP) [20], and Re-centered Local Profiles (RLP) [40] to collect character-level *n*-grams and create the authors' profiles. In addition, they applied the Inverse Author Frequency (IAF) to weight the *n*-grams, which, as they explained in their article, is merely a trajectory of the Inverse Document Frequency (IDF) weighting approach on profiles, as opposed to documents. To compare the distance between different profiles, they used the relative distance [33] for the CNG-generated profiles, the size of the intersection between two profiles for the SCAP-generated profiles, and a modified version of the cosine similarity presented in Reference [40].

SMS. Ragel et al. [54] and Ishihara [30] addressed the authorship attribution problem in SMS messages. The number of authors in Ragel et al. [54] was 20 authors, while Ishihara [30] reached 228 authors. In their article, Ishihara [30] reported using 38,193 messages for all the authors. This makes the average number of messages per author around 167 messages. However, Ragel et al. [54] used 500 messages for each author; hence, the total number of messages in their dataset was 10,000 SMS messages.

Both Ishihara [30] and Ragel et al. [54] used word *n*-grams as features and combined the SMS messages together, in one document, to increase the text size. This is important, because SMS messages, by nature, are limited in size, containing very few words. In terms of classification and validation, Ragel et al. [54] divided their dataset into training and validation sets, and then they created a profile for each author in both sets. In the following step, they used the Euclidean distance and the cosine similarity to measure the distance between the profiles in the validation and the training sets. They show that grouping 300–400 SMS messages per author increases the uniqueness of an author's profile, leading to a higher accuracy when an author's profile is being classified. However, they do not mention the number of words a profile contained when they combined all these messages, nor the average number of words per message.

In contrast, Ishihara [30] grouped the SMS messages until a certain number of words was reached. For example, if the current total number of words is 197 words, then the maximum is 200 and the next message to be added has four words, they would stop at 197. If it had fewer than four words, then they would add this message and check the next one. In terms of validation, they created two sets of messages: One set contained messages from the same author and the other contained messages from different authors. Then, a word *n*-grams model was built for each set of messages. To measure the similarity between the different models, they used the Log Likelihood Ratio function. The results show that accuracy reaches 80% when 2,200 or more words per set were used. Since the application of our work is mainly for digital forensics, we believe that it would not be possible to collect enough messages from the candidate authors to get this high number of words.

Twitter Posts. Twitter is the most targeted source for short text in the authorship attribution literature. The number of candidate authors in these studies was on a small scale ranging from 10 to 120 authors in References [10] and [63], respectively, and on a large scale up to 1,000 authors in Reference [58]. The most common feature representation approach that was used is the character-level *n*-grams.

Both Cavalcante et al. [12] and Schwartz et al. [58] used character- and word-level *n*-grams as features while using a SVM classification model. In both articles, the number of candidate authors started at 50 and then increased gradually to 500 in Reference [12] and to 1,000 authors in Reference [58]. A major difference between these two articles is that Schwartz et al. [58] proposed the concept

of *K-signature*, which they define as "the author's unique writing style features that appear in at least *K%* of the author's Twitter posts."

Layton et al. [39] used the SCAP methodology to address the authorship attribution problem for Twitter posts. In their article, they divided the tweets into training and validation tweets. Then, they combined all the training tweets that belonged to the same author in one document and extracted only character-level *n*-grams from this document as features. To create a profile for that author, they picked the top *L* most frequent features in his document and ignored the rest. The tweets in the validation set were handled in the same way and a profile was created for each author as well. Finally, the validation profile was compared to every author's profile and the similarity was simply measured by counting the number of common *n*-grams between the test profile and the candidate author's profile, i.e., the intersection between the test profile and the candidate author's profile. This similarity measure is known as the Simplified Profile Intersection (SPI).

Bhargava et al. [10] and Silva et al. [63] chose a different approach to extract features than the common *n*-grams method. Bhargava et al. [10] proposed four categories of features: lexical, syntactic, Twitter-specific, and "other." Examples of lexical features are the total number of words per tweet and the total number of words per sentence. Examples of syntactic features are the number of punctuation marks per sentence and the number of uppercase letters. Examples on Twitter-specific features are the ratio of hashtags to words and whether the tweet is a retweet. Finally, examples of features that belonged to the "other" category are the frequency of emoticons and the number of emoticons per word. As for the classification model, a radial, nonlinear kernel for SVM was used. Similarly, Silva et al. [63] applied a combination of features that they categorized into four groups: quantitative markers, marks of emotion, punctuation, and abbreviations. Two differences were observed in these two articles: The first is that Silva et al. [63] used a linear SVM instead of a nonlinear one. The other difference is that Bhargava et al. [10] experimented with combining the set of Twitter posts into a number of groups to enlarge the text body before the feature extraction step. Bhargava et al. [10] showed that grouping a set of 10 tweets together achieved better accuracy.

Deep Learning-Based Authorship Attribution Techniques. Deep learning techniques have received a lot of attention recently due to their tremendous success in various domains. Particularly, being end-to-end methods where they do not require manual feature engineering makes them very favorable over traditional methods. This is because selecting the right set of features is crucial for achieving high accuracy [22]. However, much of this success is attributed to the availability of a tremendous amount of data to train such models. One way to interpret how a Neural Network works is that it learns an implicit representation for the data in the hidden layers and then perform the classification at the output layer based on the learned abstract representation. Given enough training samples, the learned representation is better than the hand-crafted features in representing the problem to the classifier, hence, the better performance [22]. In this section, we review the works that use deep learning-based methods for authorship attribution of short text.

In a class project, Rhodes [55] explored the direction of authorship attribution using a Convolutional Neural Network (CNN). In their experiments, they used two hidden layers: an embedding layer and a convolution module (convolution, and pooling) on the character level with filters of sizes (3, 4, and 5). The model was applied on a dataset of 8 English books for 6 authors collected by Rhodes [55], and 28 English novels written by 14 authors borrowed from the PAN 2012 Authorship Identification competition. The classification was performed on the sentence level rather than on the document level. Rhodes [55] only reported the number of sentences per author for the PAN2012 dataset that ranged from around 7,000 to around 36,000 sentences. Rhodes [55] reported that their accuracy on the Books dataset was about 76% and compared it to the probability

of picking the correct class randomly, which was 16.3%. For the PAN2012, the reported accuracy was 20.52% while the random baseline was 7.14%. The accuracy was not high, and the comparison with the random baseline is not very meaningful.

Ruder et al. [57] also used various CNN configurations to address the attribution problem on emails, movie reviews, blog posts, tweets, and Reddit forum posts. Their proposed method was compared to traditional techniques, such as SCAP [20] and SVM with *n*-grams at the words' stems level. For experiments on tweets, the authors compared two settings: 10 authors and 50 authors where the average number of tweets per author is 229, and the average tweet size in terms of words is 19. The best reported results were using a character-level CNN where their model achieved an accuracy of 97.5% and 86.8% for 10 authors and 50 authors, respectively. This result might be biased, because it is unclear if the authors had applied any preprocessing to remove usernames and hashtags [39], in addition to collecting a dataset of authors who are prolific on twitter, such as celebrities. Ruder et al. [57] further explained that the network may have learned to classify the samples using subtle hints, such as usernames, and frequent terms that are used as a form of branding.

Ge et al. [21] used a feedforward Neural Network architecture to learn a Language Model instead of using a Neural Network as a classifier. Essentially, the network learned a representation for each word based on its context (a window of 4 grams). The language model was evaluated using perplexity, which measured the model's surprise in seeing a test sentence. Ge et al. [21] optimized their network to minimize[3] the perplexity while training a separate language model per author to predict a set of words in a test sentence. The author whose language model has the lowest perplexity is the most plausible author of the test sentence. Using this approach, Ge et al. [21] tried to classify transcripts of 16 online courses, hence 16 authors, from the Massive Open Online Courses (MOOC) Coursera platform. There are around 8,000 sentences per course and an average of 20 words per sentence. This approach achieved 95% accuracy, which according to Ge et al. [21], is comparable with Reference [16] which uses Naïve Bayes for classification. However, the authors noted that the task might be too easy due to the availability of a huge training set.

Shrestha et al. [62] used a three-layer CNN on the character bi-grams level to address the attribution problem for English tweets and proposed a technique to visualize the outcome of the attribution process. With 50 authors and 1,000 tweets per author, Shrestha et al. [62] reported that their method can achieve an accuracy of 76.1% and compared it with SVM running on *n*-gram, which has an accuracy of 71.2%. It is unclear if the authors had removed hashtags and usernames before performing the attribution process, as per the suggestion in Reference [39]. Additionally, about 30% of the authors behave as "bots," where they always use the same pattern to tweet. For example, a tweet about the weather could have the following format: @<<network name>>: Today <<data>>, the high temp. is << ## >> and the low is << ## >>. It is unclear whether or not these "bot" authors were excluded from the experiments. In terms of visualization, Shrestha et al. [62] calculated the Saliency score [41] for each character and used the color intensity to express its value. Compared to our work, we visualize the similarity on three levels, characters, words, and POS, and we provide an interactive tool where the user can focus on a certain feature and further investigate the candidate authors' profiles.

To summarize, using deep learning techniques for authorship attribution may achieve high accuracy only when a large volume of training data is available. Both Kim [34] and Zhang et al. [69] used Convolutional Neural Networks (CNN) on the word level and the character level, respectively, to perform various text classification tasks. Their results showed that training a CNN requires a

[3]Ge et al. [21] report maximizing the perplexity in their article. We believe that it is just a typo as they define the perplexity and derive it correctly using probability principles.

huge amount of data. In fact, Zhang et al. [69] noted that using a Logistic Regression classifier with *n*-gram features achieves a higher accuracy compared to a character-level CNN when the number of training samples is less than 500,000 samples. If the goal of the attribution process is to use it for forensics applications, then there are a number of restrictions that should be considered in the experimental settings. First, the number of training samples per author that is available for training should be limited [43]. Second, even though Neural Networks are end-to-end systems, preprocessing should be used to ensure that no subtle hints are provided to the classifier [39, 57]. Finally, if the outcome of the attribution process is to be used as evidence in the courts of law, a visualization technique should be provided to present the results in a clear and intuitive way. Perhaps, if the results for a Neural Network and a traditional classifier, such as SVM, are similar, one might use SVM which requires less time to train.

2.2 Arabic Authorship Attribution

In reviewing the work on Arabic, we are specifically interested in three elements: the text source and its size, the writing style features, and the classification techniques.

Kumar and Chaurasia [37] performed authorship attribution on Arabic novels where the corpus consisted of a training and a test set for four authors. The average number of words per author in the training set was 67,635 words, and the test set was 164,673.25 words. In terms of features, they used the initial and final bi-grams and tri-grams [56] that, in the case of bi-grams, are formed from the first two letters and the last two letters of every word. Similarly, tri-grams are formed of three letters. The classification process is based on the dissimilarity measure algorithm [33]. Different *profiles* for each user were tested, and a profile was the most frequent 200, 500, and 700 bi- or tri-grams. For each profile setting, a dissimilarity threshold value was calculated by comparing the author's profile from the training set with the profile from the test set. For a test document, a new profile was built and the dissimilarity value between the author's profile and the unknown document was compared to the author's dissimilarity threshold value. Their results suggest a 100% accuracy when the initial tri-gram is used with any of the profile sizes.

Ouamour and Sayoud [50] built their dataset from ancient Arabic books, where the average number of words per book was around 500 words. They collected three books for each one of the 10 authors in their dataset. Their features set consisted of (one to four)-gram word-level features in addition to "rare words." The datasets were tested using different machine learning classifiers. Each feature was tested alone using Manhattan distance, cosine distance, Stamatatos distance, Canberra distance, Multi-Layer Perceptron (MLP), SVM, and Linear regression. The best results were reported when rare words and 1-word gram features were paired with a SVM classifier. As the value of *n* (in *n*-grams) increased, the reported results showed significant deterioration. They correlated this deterioration in performance to the small-sized text they used.

Shaker and Corne [59] created a dataset composed of 14 books written by six different authors, with the average number of words per book being 23,942 words. Their approach utilized the most frequent function words to discriminate between two textbooks. Motivated by the work in Reference [48], they generated a list of 105 function words and ignored the 40 words that are comparatively less frequent than the rest, to end up with a list of 65 **A**rabic **f**unction **w**ords, denoted by *AFW65*. Further filtering was applied to the *AFW65* set where the 11 words with the least frequency variance were removed. The resulting set of 54 **A**rabic **f**unction **w**ords was denoted by *AFW54*. This created a new set of words that consisted of 54 function words. Each book was divided into chunks of words and two sizes were experimented on 1,000 and 2,000 words. For each chunk, a feature-vector was created using the ratio of each function word, and the author name was assigned to the chunk as a class. This created four experimental settings. They used a hybrid approach of Evolutionary Algorithm and Linear Discriminant Analysis that they developed for

English in Reference [60] to classify a set of test documents. The best reported result was achieved when a chunk of 2,000 words was used along with the *AFW54* set of function words.

Alwajeeh et al. [7] built their text corpus from online articles. They manually selected five authors with 100 articles each. Then they manually annotated the articles and extracted the features from them. The average number of words per article was 470.34 words and features, such as the average number of general articles, characters per word, punctuation per article, and unique roots per article, were extracted. They also investigated the effect of using a Khoja stemmer, which returns the original root of the specified word. They used Naïve Bayes and SVM as their methods of classification and the reported accuracy almost reached 100%. In their discussion of the results, they highlighted the negative effect that the Khoja stemmer introduced. They explained that the reason behind this effect is that root stemming causes two different words with different meanings to become the same word, which in return leads to information loss and, therefore, bad performance.

Abbasi and Chen [2] and Altheneyan and Menai [6] used exactly the same set of features. In fact, Abbasi and Chen [2] proposed these features, and then Altheneyan and Menai [6] adopted their suggestions in their article. There are 418 features divided as follows: 79 lexical features, 262 syntactic features, 62 structural, and 15 content specific. Altheneyan and Menai [6] tested these features on a dataset composed of 10 authors and 30 books, with an average number of words per book ranging between 1,980 and 2,020 words. However, Abbasi and Chen [2] collected 20 web forum messages for each of their 20 authors, and the average number of words per message was 580.69 words. In both these studies, variations of Naïve Bayes, SVM, and C4.5, a famous decision tree classifier, were used, and the reported accuracy ranged from 71.93% to 97.43%.

Rabab'ah et al. [53] collected a dataset of 37,445 tweets for 12 users from the top Arab users on Twitter. On average, they collected around 3,120 tweets per author. Three sets of features were used in this study: stylometric features provided by [4], uni-grams, and morphological features extracted using MADAMIRA tool [51], which is a tool made by combining the functionality of MADA [23] and AMIRA [18] tools for Arabic feature extraction. They used Naïve Bayes, Decision Trees, and SVM for classification and experimented with the sets of features separately and combined. The best result was achieved using SVM with all three sets of features. This study was followed by Reference [5] on the same dataset, where Al-Ayyoub et al. [5] investigated the effect of using feature selection tools such as Principle Component Analysis (PCA) and Information Gain on reducing the running time of the classification process.

Al-Ayyoub et al. [4] investigated the authorship problem for news articles. They collected around 6,000 articles for 22 authors, where each author has 220 articles on average, and the articles' lengths ranged between 202 and 565 words. For their work, they considered two sets of features. In the first set, they compiled a list of stylometric features from References [1, 2, 15, 49, 59], while in the other set they considered all the uni-grams that have more than 1,000 occurrences in the dataset and then applied feature reduction using a correlation-based feature selection technique [24]. The value for a feature is the TF-IDF score. Finally, they used Naïve Bayes, Bayes Networks, and SVM to compare the performance using each feature set separately. The outcome of this study is that using stylometric features yielded a higher accuracy compared to using uni-grams with TF-IDF scores.

As discussed above, all the relevant works either focus on long Arabic text data or use a very large number of writing samples per author. Given the exploding text generated online, there is a pressing need to benchmark the performance of existing techniques on Arabic short text. Moreover, none of the relevant works focus on interpretability, which is a critical factor in real-life investigation scenarios. In this article, we address the knowledge gap by benchmarking various instance-based and *n*-gram baseline on short text data from Twitter. We also adopt a model that can present visualizable attribution results.

2.3 Visualization for Authorship Attribution

State-of-the-art attribution techniques that are known to have good accuracy are complex and/or impossible to visualize. For example, consider a SVM model that maps the input into a new dimension when the input is nonlinearly separable. Such models cannot be visualized beyond three dimensions, where each dimension represents a feature. Since a typical attribution problem has much more than three attributes and hence requires more dimensions, it is impossible to visualize the writing samples of all the authors on a plane and the decision boundary that divides them. In contrast, a decision tree is easy to present, either as a tree or by converting it into a set of rules. However, decision trees do not perform as well as an SVM or a random forest.

We identified the work of References [3, 9, 35] and [19] to be the only work on visualization with authorship attribution. The problem with Reference [35] is that it produces three-dimensional images, one image per author, to be used for author identification; however, these images are difficult to compare holistically. In contrast, Reference [9] visualizes each feature separately and does not aggregate them to find the most plausible author. Instead, it leaves the decision for the user to find the most plausible author. The problem with this approach is that it causes the decision to be dependent on the user's understanding of the visualized figures. Finally, Abbasi and Chen [3] produces one graphical representation per feature, but this representation cannot scale up to a large number of features. Additionally, the authors highlighted a limitation of their approach by saying that its performance is constrained when used for text less than 30–40 words long. This limitation prevents its application to Twitter posts as tweets are naturally much shorter. Ding et al. [19] provides a visualization technique that overcomes all these previous points. The algorithm provides the most plausible author and its confidence for this outcome and then motivates its findings to help the user understand the outcome.

3 AUTHORSHIP ATTRIBUTION

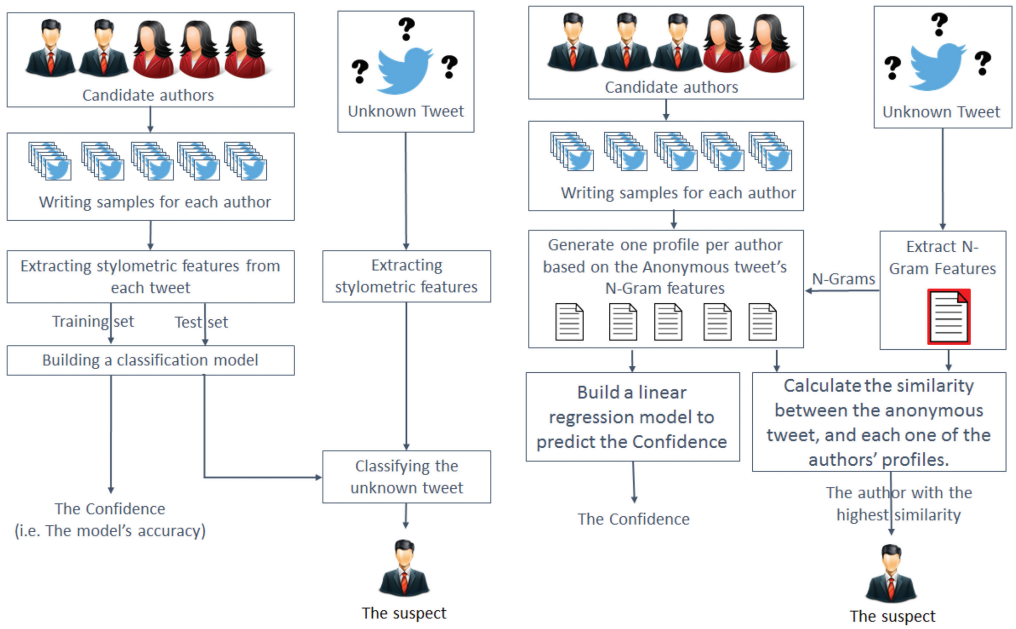
Stamatatos [65] described three approaches to extract the writing style from writing samples. The first approach is instance based, in which a writing style is extracted from every sample separately. By using this approach, the candidate author will have s-styles, where s = number of writing samples per author. The second approach is profile based in which all the writing samples for a particular author are used to generate one writing style for that author. Note that the profile-based approach is different from authorship profiling, where the task is to infer the characteristics of the author such as the age, gender, education level, and so on. The third approach is a hybrid one that starts as an instance-based one, and then the features are aggregated over all the instances to create one profile per author.

Figure 2(a) shows the steps for the attribution process using the instance-based versus the profile-based approach (Figure 2(b)). In Section 3.1, we explain the steps for the instance-based approach, and in Section 3.2 we explain the steps of the profile-based approach.

3.1 Instance-Based Authorship Attribution

Refer to Figure 2(a). The first step in the attribution process is to collect a set of writing samples for each one of the candidate authors. This is explained in detail in Section 4.1. Assuming that the set of candidate authors is identified and a collection of writing samples for each one of them is collected, the next step is to analyze these writing samples to extract the writing-style features. We discuss the various types of features in Section 3.1 below.

Extracting Stylometric Features. In this section, we provide a detailed description of the features that we extracted for each one of the writing samples. As mentioned earlier, Abbasi and Chen [2] highlighted that techniques developed for English cannot be directly applied to Arabic due to the



(a) Instance-based authorship attribution. (b) Profile-based authorship attribution.

Fig. 2. Instance-based vs. profile-based authorship attribution.

different morphological nature of each language, which directly affects the feature extraction process. In fact, the features list is the only language-dependent element in the authorship attribution process, while the choice of the classification model, such as Naïve Bayes, or SVM, is not. This is because the feature extraction process uses the features list to convert text to the feature-vectors, which the classifiers use to build a classification model.

In general, an English-based features list can be used as a starting point for creating a new features list for non-English authorship analysis. First, some basic language-specific features have to be modified in the feature list. For example, the frequencies of alphabets in English have to be replaced with the frequencies of Arabic letters and the ratio of capital letters in English has to be removed, because Arabic letters have only one case. However, the use of elongation "-" can be found in Arabic but not in English. Therefore, it has to be included as a feature for Arabic. Second, the list of features has to be modified when the source of the investigated text changes from news articles, for instance, to tweets. This is because some features are meaningful in one source but not in another. Consider the feature "The greeting line." This feature is only meaningful in e-mail analysis. Looking for a greeting line in a tweet will not yield any results. Finally, there are some features that are common and can work for different languages and in different domains, such as "the ratio of space to characters," but the number of such features is low. If the number of features is low, i.e., the features are not representative, then the accuracy of the classification model will be very low. After that, choosing a classification model is not an issue, because the classifier will use the feature-vectors the same way whether they were generated for an English text or a non-English text.

We adopted a similar structure of Arabic features presented in Reference [2], with two main differences: (1) we removed all the features that are inapplicable to Twitter posts (e.g., font size

and greetings) and (2) we used a larger set of function words. The following steps show how the features were extracted. We have three categories of features: lexical, structural, and syntactic.

Lexical Features. We started by counting the number of characters that included diacritics or "Arabic Tashkil" and special characters and punctuation and excluded white space characters such as spaces, tabs, and newline. Let M be the number of characters in a tweet whether this character occupies a location or not. Examples are alphabets and diacritics. Next, we calculated the ratios of digits (0–9) to M, spaces to M, tabs to M, and spaces to all characters. Last, we generated the ratio of every single alphabet to M. Despite the fact that the collected tweets are in Arabic, we also calculated the ratio of English alphabets. This was important, since we observed some tweets that included both English and Arabic words. Finally, it is important to mention that we considered the Alif "ا" letter and the Alif with Hamza "أ" letter to be two different letters, as opposed to Altheneyan and Menai [6], who combined them under the Alif "ا" letter.

The previous set of features was observed on the characters' level and that is why they are called character-based features. This next set of features, however, was observed on the words level and is therefore called word-based features. The first word-feature is intuitive, which is the word count W. Before the words were counted, we replaced punctuation and white space characters with a single space. Special characters and diacritics were kept when the words were counted, because they are parts of words and will not affect the word counting process. Next, the length of each word was used to find the average word's length. The average word's length feature was calculated by summing the lengths of all the words and dividing the sum by the number of all words. In addition, the words' lengths were used to find the number of short words (1 to 3 characters) and then the ratio of short words to the words count W. Below, we present two lists summarizing the character- and the word-based features.

- Character-based features:
  (1) Character count excluding space characters (M).
  (2) Ratio of digits to M.
  (3) Ratio of letters to M.
  (4) Ratio of spaces to M.
  (5) Ratio of spaces to total characters.
  (6) Ratio of tabs to M.
  (7) Ratio of each alphabet to M (Arabic and English): [a–z] (26 features), [ا–ي] (28 features) and {أ، إ، آ، ء، ة، ؤ، ى، ئ} (8 features). (Total is 62).
  (8) Ratio of each special character to M: <>% |{} [ ] @ # ~ + - * / = \ $ ^ & _ (21 features).
- Word-based features:
  (1) Word count (W).
  (2) Average word length.
  (3) Ratio of short words [1–3] character to W.

Structural Features. The average sentence length, in terms of characters, was calculated. To do so, newline "\n", period ".", question mark "?" and exclamation mark "!" characters were used to divide a tweet into a set of sentences. This feature is also used to find the average sentence length the same way the average word's length was calculated. The last structural feature obtained is the ratio of blank lines to all lines. This can be calculated by looking for two newline characters together, i.e., \n\n. A summary of the previous features is provided below:

- Textual features:
  (1) Sentence count.
  (2) Average sentence length.
  (3) Ratio of blank lines to all lines.
- Technical features. Examples of technical features are font size and color, but these kinds of features are not applicable to Twitter, because users do not have control of them. All tweets are published with the same font size, type, and color.

Syntactic Features. The previous features can be called generic or language independent. This is because these features can be collected from tweets regardless of the language in which they were written. Since we are targeting Arabic tweets, we added a set of Arabic-derived features.

- Diacritics. Arabic scripts have many diacritics and Tashkil "تشكيل", where the latter includes the Harakat "حركات" (vowel marks). Ideally, Tashkil in Modern Standard Arabic is used to represent missing vowels and consonant length, and it helps identify the words' grammatical tags in a sentence. For example, the question من ضرب الرجلُ؟ means: whom did the man hit? ("من = who/whom", "الرجل = the man" and "ضرب = hit"). However, if the Damma (ُ) on the word الرجلُ is replaced with a Fatha (َ), the question will become من ضرب الرجلَ؟, meaning: who hit the man? So, the change on Tashkil on the word الرجل from Damma to Fatha changed it from being the subject to become the object. If Tashkil was not provided, then the context can help to understand the statement. However, if no context was provided, then the reader will not be able to tell which question is being asked. The following diacritics have been observed:
  — Hamza: "أ، ؤ" and stand alone "ء" were converted to "أ" and "ئ" and "إ" were converted to "إ".
  — Tanwin symbols: "بًا، بٌ، بٍ". Tanwin always accompanies a letter. It never appears alone.
  — Shadda: "بّ" the doubling of consonants. It also does not appear alone.
  — Madda: "آ" the fusion of two Hamzas into one. "ءأ".
  — Harakat: includes Fatha "بَ", Kasra "بِ", Damma "بُ" and Sukoon "بْ". They always accompany a letter.
- Punctuation. Arabic punctuation is similar to that of English; the difference is mainly in the direction that the punctuation faces. For example, while the English question mark "?" faces the left (faces the question), the Arabic question mark faces the other way "؟". Below is the set of punctuation marks that were observed:
  — Arabic Comma ،
  — Arabic colon :
  — Arabic Semi-colon ؛
  — Arabic Question mark ؟
  — Arabic Exclamation mark !
  — Arabic Single quote ‘
  — Arabic End single quote ’
  — Arabic Qasheeda ـ only used for "decoration"
  In addition to Arabic punctuation, English punctuation marks , . ; " ' and ? were also added to the punctuation set of features.

- Function words. Function words are a set of words that can be grouped together due to a common property, for example, names of months or pronouns. Below we list examples of these function words, keeping in mind that each word is considered a stand-alone feature:
  - —Interrogative nouns "أسماء الاستفهام", e.g., "ماذا، متى، هل، كيف، كم".
  - —Demonstrative nouns "أسماء الإشارة", e.g., "هذا، هذه، ذلك".
  - —Conditional nouns "أسماء الشرط", e.g., "إن، حيثما، كيفما، أيّ".
  - —Exceptional nouns "أدوات الاستثناء", e.g., "إلا، غير، سوى".
  - —Relative pronouns "الاسم الموصول", e.g., "الذي، التي، اللذان، اللتان، اللاتي".
  - —Conjunction pronouns "حروف العطف", e.g., "ثم، أو، بل".
  - —Prepositions "أحرف الجر", e.g., "من، إلى، في، عن، على".
  - —Indefinite pronouns "الضمائر", e.g., "أنا، أنت، هو، هما، هم، هي، هما".
  - —Eljazm pronouns "حروف الجزم", e.g., "إنْ، لم، لا".
  - —Incomplete verbs "الأفعال الناقصة". This family of verbs contains a sub-family of verbs: Kada Wa Akhwatoha "كاد وأخواتها", Inna Wa Akhawatoha "إنّ وأخواتها", Kana Wa Akhawatoha "كان وأخواتها".
  - —Common and famous Arabic names, such as:
  - —Names of Arabic world countries, e.g., "الامارات، السعودية، الاردن" and their capitals' names, e.g., "أبوظبي، الرياض، عمّان".
  - —Some popular Arabic names,[4] e.g., "محمد، عمر، عثمان، عبيدة" [8].
  - —Names of Hijri and Gregorian Arabic months, e.g., "يونيو، محرم، شعبان، جولية".
  - —Numeral names and ordinal numbers, e.g., "الأول، عشرة، عاشر".
  - —Currency names, e.g., "دينار، درهم، دولار".

In total, there are 541 different function words, which means 541 additional distinct features. To summarize how syntactic Arabic-specific features were collected using the feature extraction tool, the following list is provided:

(1) Occurrence of each diacritic (12 features).
(2) Ratio of punctuation to M.
(3) Occurrence of each punctuation (14 features).
(4) Ratio of function words to W.
(5) Occurrence of each function word (541 features).

Preprocessing for the Instance-Based Approach. Instance-based authorship attribution is considered a classification problem, in which a model is used to classify an anonymous text after training this model on the writing samples of the candidate authors. To do that, various data-mining tools can be used, such as WEKA [25] or RapidMiner [27]. We used the Java implementation of WEKA to train and validate the classification models.

At this point, all the features were extracted from the tweets and stored in a relational database. To be able to use WEKA, we need to extract the features from the database and store them in

[4] http://ArabiNames.com/categories.aspx.

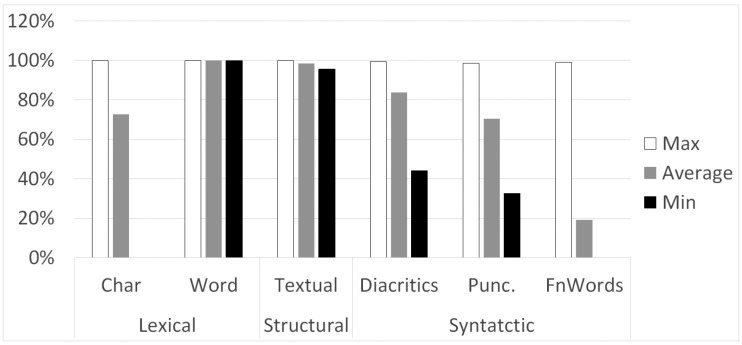


Fig. 3. The maximum, average, and minimum ratio of usage for each feature grouped by the category.

a readable format that WEKA can read. To prepare a dataset for an experiment, the features for each writing sample are collected from the database, normalized, and associated with the author of the writing sample as a class label. After the features are collected for all the writing samples, we used WEKA's "RemoveUseless" filter on these features to exclude all the *useless* ones in the training samples. A feature is deemed useless if, for all the writing samples, it has the same value. For example, if feature FnW_1 has the value 0.5 for all the writing samples, then this feature cannot help in identifying the author. Note that this filter is applied to a specific training set, i.e., after a random set of authors and a random set of writing samples are selected and not on the database of features. This means that a feature that was deemed useless in one experiment may be usable in another one and that depends on the set of candidate authors and the selected writing samples for each author. Because of that, the number of useless features in a specific experiment varies from one experiment to another.

The reason why these useless features appear in the first place is the large number of features that we use. For example, we collect 541 function word-features. It is very unlikely that a small set of tweets will contain all these function words. (See Appendix A for a list of the top 100 highest frequency function words.) However, this should not affect the accuracy of the model, since such features will not be used to build the classification model. Even if such feature appears later in the validation instance (i.e., the anonymous tweet), the already built model will not be able to use it. However, removing these useless features should reduce the time for training and validation for a classification model.

Figure 3 shows the maximum, average, and minimum ratio of usage for each feature grouped by the category. A ratio is calculated by counting the number of times a feature was used in an experiment, divided by 200 experiments.[5] For example, a ratio of 90% means a feature was used in 180 experiments and removed by the RemoveUseless filter in 20 other experiments.

As Figure 3 shows, only function-word features are used less than 60% on average. Word-level lexical features and textual features were used in all experiments with an average ratio of usage being 100% and 98.5%, respectively. The minus sign "–" was the only character-level feature that was not used in any experiment and had a ratio of 0%. For the function-word features, around 26% of them were not used in any experiment. We listed these features in Appendix B.

The Classification Process. Figure 2(a) illustrates the process of attributing an anonymous tweet to one of the candidate authors using the instance-based approach. As the figure shows, the next step after extracting the features is to build the classification model. To do that, we used the 10-fold

[5]All the possible settings as per the experimental setup in Section 4.2.

cross-validation technique and divided the writing samples into training and validation sets. The result of this process is a classification model and its accuracy, which is used as a confidence. If the model's accuracy is low, then this means that the model is not able to differentiate between the authors based on their writing samples. Regardless of how low the accuracy is, the model will always output a candidate author for the anonymous text. In this case, it is up to the authorship attribution domain expert to evaluate to decide whether to accept the results or not.

We used four different classification techniques to evaluate the performance of the instance-based authorship attribution approach to compare it with the performance of the profile-based approach. These techniques are Naïve Bayes, SVM, Decision Trees (DT), and RF. The reason for choosing these classification models is this: One main application of authorship attribution techniques is to use their results and analysis as evidence in courts of law. Because of that, the interpretability of the results is as important as high accuracy. It is crucial that the findings are not only accurate but also intuitive and convincing. For example, SVM and Random Forests are known for high accuracy. However, the resulting models are complex and can only be seen as a black box, as opposed to Naïve Bayes and decision trees, whose results are easier to represent. An authorship attribution expert needs to explain a conclusion rather than merely present it and using such complex models will not enable an expert to do so. Following is a brief description of each model.

(1) **Naïve Bayes.** A Naïve Bayes classifier applies Bayes rule of conditional probability [68] to accurately predict the class of a given instance [31]. In terms of computational complexity, a dataset with many attributes would be too exhaustive for the computer resources [26]. Therefore, to simplify this process, Naïve Bayes assumes that all the attributes are independent and are of the same weight [46]. Because this is not true in most real-world scenarios, the term "Naïve" was associated with this classifier.

(2) **Support Vector Machines (SVM).** We use Chang and Lin [13]'s implementation of SVM. A SVM is a supervised learning [61] algorithm that extends a linear classification model to solve a multi-class, linear, or nonlinear classification problem where the n-attributes are mapped to an n-dimensional plane with n-axes [68]. Such models use various optimization techniques [68] to find the *Maximum Marginal Hyperplane* (MMH) [26] that can separate these instances. This makes SVM models slow and computationally expensive [61] but very accurate [68], as they always provide the global solution [26].

(3) **Decision Trees.** We use WEKA's implementation of the well-known C4.5 algorithm [52] presented in Reference [68]. Initially, Quinlan introduced the ID3 algorithm that uses Information Gain as an attribute splitter [68]. C4.5 is the result of a number of improvements applied to ID3, among them is using the gain ratio instead of only using Information Gain and using pruning to remove the "unreliable" branches caused by noise, or due to over-fitting [26, 68], as well as dealing with instances with missing values or numerical attributes [68].

(4) **Random Forests.** Random Forest [11] is a technique that utilizes bagging and randomization, which are examples of *ensemble learners*, to produce a classification model that outperforms these individual classifiers [68]. While bagging is performed by using decision trees for a number of times (instead of using different classifiers), random splitting is utilized when an attribute is to be chosen in each iteration of the tree induction process. This random attribute is selected from the N best attributes instead of the single "best" attribute [11].

3.2 Profile-Based Authorship Attribution

In this section, we discuss the process of performing authorship attribution using the n-gram approach. First, we analyze the writing samples of each author and extract three sets of n-gram

Table 1. Sample *n*-Grams Extracted from the Text: "It Is Noticed and Appreciated" and the Corresponding Part-of-Speech Tag Sequence Is "PRP VBZ VBN CC VBN"

Modality	*n*-gram	Length	Examples
Lexical	Word	1–3	"It," "it is," "it is noticed," "is noticed," and so on.
Character	Character	1–3	"no," "not," "notic," "tice," "notice," "a," "an," "nd," and so on.
Syntactic	P-O-S	1–3	"PRP VBZ VBN," "CC VBN," "VBN CC," and so on.

features per author, one on each modality level. We perform the same step for the anonymous tweet and produce three feature sets as well. Second, for each modality level, we use the *n*-grams of the anonymous tweet as a feature-vector and use the Term Frequency-Inverse Term Frequency (TF-IDF) technique to calculate a score for each feature. This produces three profiles for each author as well as three profiles for the anonymous tweet, where each profile corresponds to one modality level. Third, a similarity function is used to compare the authors' profiles to the profile of anonymous tweets, and each author will have three similarity scores, one for each modality level. Fourth, we calculate three confidence scores, one for each modality level. Each score describes the model's ability to distinguish between the authors' profiles if only that modality level is used. Neither the anonymous tweet nor its profiles/features are used in this step. Fifth, we use the confidence scores that are calculated in the previous step to weight the similarity scores of the authors and calculate one combined similarity score for each author. Finally, we project the similarity scores and the confidence values on the anonymous tweet to provide a visual representation of the results. Following, we discuss each step in detail.

Extracting the N-Grams from the Anonymous Tweets and the Sample Tweets. This section describes the process of extracting the *n*-gram features from the authors' writing samples. A *gram* is a unit of text (i.e., token) based on the modality level, and *n* is the integer number of consecutive tokens that are considered as one feature. The *n*-gram approach can be applied on three modality levels: characters, words, and parts-of-speech (POS). For example, 2-grams on the character level means that we tokenize a text into a series of characters, then we take every two consecutive characters as one feature. Table 1 shows a sample sentence and the corresponding *n*-gram features for each modality level.

We used Stanford's [45] NLP library[6] for text segmentation, tokenization, and POS tagging. Note that the term *n*-gram is used in the literature to indicate that the number of tokens in a feature is exactly *n*. For example, the term 3-grams, or tri-grams, means each feature has exactly three tokens. In this work, however, we extract all the grams of lengths 1 to *n* as shown in Table 1, and, for brevity, we use the term *n*-gram instead of $1 - n$-grams.

Generating the Authors' Profiles. After extracting the features, we use the TF-IDF technique to score each feature and create the authors' profiles. This scoring technique, which is famous for its simplicity, gives a high score for words that appear many times in one piece of text (Term Frequency) while penalizing terms that appear many times in other documents. To understand the motivation behind this technique, consider the word "the." Due to its usage, the word "the" is likely to appear many times in a document, compared to other terms in that same document. However, it will also appear frequently in other authors' documents. Therefore, it will get a low score for being very common, i.e., not unique for a certain author. In contrast, consider the word "kindly" that might be common in one author's text while being replaced by the word "please" by

[6] http://nlp.stanford.edu/.

another author. If this was the case, then these two terms will have high scores, given that they appeared in one author's text more frequently than for other authors.

Equation (1) and Equation (2) show the formulas to calculate the Term Frequency (TF) and the Inverse Document Frequency (IDF), respectively, where W_i is all the writing samples, i.e., tweets, for candidate author c_i; $|C|$ is the size of the set of candidate authors, i.e., the number of candidate authors; and b is a constant that equals 0.1,

$$TF(gram, W_i) = \frac{frequency(gram, W_i)}{maxGramFrequency(W_i)}, \tag{1}$$

$$IDF(gram) = log\left(\frac{|C|}{b + |AuthorsEverUsed(gram)|}\right). \tag{2}$$

To illustrate how these equations are used to generate the profiles of the anonymous tweet and the authors, we provide the following example: Let a be an anonymous tweet and a set of two candidate authors C, where W_1 and W_2 contain all the tweets written by authors c_1 and c_2, respectively. Let the n-gram features for a, W_1, and W_2 be extracted as per Section 3.2. For the sake of this example, assume that we are performing the attribution process on the word level only and that the feature-vector based on the anonymous tweet a is $[gram_1, gram_2, gram_3]$.

To generate a profile for the anonymous tweet, we only use Equation (1) and we do not use the writing samples of the candidate authors. Assume that the frequencies for $gram_1$, $gram_2$, and $gram_3$ in the anonymous tweet are 1, 2, and 5, then the profile for this anonymous tweet on the word level will be [0.2, 0.4, 1].

To generate a profile for author c_1, we compute the frequencies for the same grams: $gram_1$, $gram_2$, and $gram_3$ in the author's tweets. For example, if we examine all the author's tweets we find that $gram_2$ appears 5 times, then $TF(gram_2, W_1) = 5$. Assume that $TF(gram_1, W_1) = 3$ and $TF(gram_3, W_1) = 0$, i.e., $gram_3$ does not appear in any of author c_1's tweets. So far, using only Equation (1), the feature-vector for c_1 is [0.6, 1, 0]. Similarly for author c_2, we calculate the frequencies for $gram_1$, $gram_2$, and $gram_3$ using Equation (1). Assume that the resulting frequencies for $gram_1$, $gram_2$, and $gram_3$ are 4, 0, and 0. Therefore, the feature-vector for c_2 is [1, 0, 0].

Next, we need to calculate the IDF value for each gram, given by Equation (2). Notice that the number of authors is fixed and b is a constant, so we only need to calculate $|AuthorsEverUsed\,(gram)\,|$, i.e., find the number of authors who used that gram in any of their tweets. $Gram_1$ was used by both authors, so $IDF(gram_1) = log(2/(0.1 + 2)) \approx -0.02$. $Gram_2$ was used by only one author, so $IDF(gram_2) = log(2/(0.1 + 1)) \approx 0.26$. Finally, $gram_3$ was not used by any author, so $IDF(gram_3) = log(2/(0.1 + 0)) \approx 1.3$. Notice that had we not used the constant b, a division by zero would have occurred. Therefore, we added the constant b and set its magnitude to be smaller than 1. Finally, we penalize each TF value with the corresponding IDF value. The resulting feature-vectors for author c_1 and c_2 on the word level are [−0.012, 0.26, 0] and [−0.02, 0, 0], respectively.

We perform the same process to generate the profiles on the remaining modality levels. After generating three profiles for each author, we calculate the similarity scores as explained in the following section.

Computing the Similarity Scores per Modality Level. In the previous step, we generated the writing-style profiles for all the authors as well as for the anonymous tweet on all three modality levels, where each profile is a vector.

To measure the similarity between the anonymous tweet and an author's profile on a certain modality level, we need to calculate the distance between their vector-profiles. One simple technique to calculate this distance is the Cosine Similarity that is shown in Equation (3). From a

geometric perspective, the smaller the angle between two vectors, the more similar they are. By simplifying the formula of the cosine similarity, the distance ends up being the dot product of the two vectors,

$$\begin{aligned} similarity\left(\vec{S_i}, \vec{S_\alpha}\right) &= proj_{\vec{S_\alpha}}\vec{S_i} \times \parallel \vec{S_\alpha} \parallel \\ &= \parallel \vec{S_i} \parallel \times \cos(\theta_i) \times \parallel \vec{S_\alpha} \parallel \\ &= \parallel \vec{S_i} \parallel \times \frac{\vec{S_i}.\vec{S_w}}{\parallel \vec{S_i} \parallel \times \parallel \vec{S_\alpha} \parallel} \times \parallel \vec{S_\alpha} \parallel \\ &= \vec{S_i}.\vec{S_w}. \end{aligned} \tag{3}$$

To understand how the dot product can measure the similarity, consider the following example. Assume that we are comparing an anonymous text a and two documents d_1 and d_2. Let the feature-vector for a be [1, 1, 1, 1, 1], for d_1 be [1, 0, 0, 0, 0], and for d_2 be [1, 0, 0, 1, 1], where the value "1" means the feature is observed in that document and a value "0" means it is not. By simply looking at the vectors, we can see that d_2 is more similar to a, because it contains three observed features, while d_1 has only one. We can reach the same conclusion using the dot product of the vectors. The distance between a and d_1 is $\vec{a}.\vec{d_1} = (1\ x\ 1) + (1\ x\ 0) + (1\ x\ 0) + (1\ x\ 0) + (1\ x\ 0) = 1$. The distance between a and d_2 is $\vec{a}.\vec{d_2} = (1\ x\ 1) + (1\ x\ 0) + (1\ x\ 0) + (1\ x\ 1) + (1\ x\ 1) = 3$.

Using this similarity measure, we compute three similarity scores for each author, one on each modality level.

Calculating the Confidence per Modality Level. In the previous step, we calculated the similarity scores for each author. In this step, we calculate a confidence value that describes the model's ability to discriminate between the authors' profiles. To understand the motivation behind this step, consider these two cases with the following similarity scores on the same modality level for three candidate authors. Case1: $Sim(C_1, a) = 5$, $Sim(C_2, a) = 2$, $Sim(C_3, a) = 1$ and Case2: $Sim(C_1, a) = 5$, $Sim(C_2, a) = 4.7$, $Sim(C_3, a) = 4.2$. In both cases C_1 has the highest similarity score; however, in Case2 the similarity scores are too close to each other, which means that the authors' writing styles are very similar. We quantify the model's ability to discriminate the authors' profiles by measuring the model's ability to correctly predict the author of each one of the writing samples if they were used as anonymous tweets. To do that, we divide the authors' writing samples into 10 folds: Nine folds to be used as writing samples and one fold to be used as anonymous tweets. Note that this is done for each modality level separately.

For example, consider a problem with five candidate authors, each one with 20 tweets, where we are calculating the confidence on the character level, i.e., the degree of similarity between the authors' profiles if only character-level features are used in this attribution problem. We start by dividing these tweets into 10 folds, 9 for training and 1 for validation, where all the authors are represented equally in both sets. In other words, the classes are balanced in both the training and validation sets. For example, each author has 18 writing samples to be used to create his/her profile and 2 samples to test the model. Next, we consider each tweet in the validation set as a separate attribution case, i.e., use its features to create a feature-vector, use this vector to generate its profile as well as the candidate authors profiles and calculate similarity scores between the authors' profiles and the profile of the validation tweet. Finally, the author with the highest similarity score is the most plausible author.

Before we move on to the next validation tweet, we extract the following six features from the current tweet: (1) the highest similarity score, (2) the lowest similarity score, (3) the average similarity score, (4) the difference between the highest and the second to the highest similarity scores (i.e., the runner-up), (5) the length of the validation tweet (in tokens, on the same modality

level), (6) the number of tokens that appears in both the anonymous (i.e., validation) tweet, and the tweets of the author with the highest similarity score. We repeat the same steps for the rest of the tweets in the validation set.

After considering all the validation tweets in that validation fold, we calculate an accuracy score as follows. For each time the algorithm predicted the correct author it receives a score of 1, else it receives a 0. For example, if the algorithm correctly predicted the author for 7 of 10 validation tweets, then the accuracy for this fold is 70%. This accuracy is assigned to each tweet in the validation set, and it is considered as the model's confidence score for that tweet.

We perform this process 10 times, each time considering a new fold for validation. At the end of this 10-fold process, we will have a feature-vector of length 6 and a confidence value for each tweet in the set of writing samples of every author for a specific modality level. To calculate the model's confidence score for the original anonymous tweet, we use a linear regression model. This model is trained on the writing samples and the same six features are extracted from the anonymous tweet based on the similarity scores of the authors that were measured in Section 3.2. The model is then used to predict a confidence score for the anonymous tweet on a specific modality level.

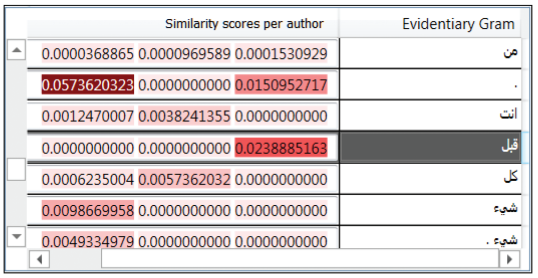
In total, three linear models are trained, one for each modality level. The outcomes of this process are three confidence scores, one for each modality level.

Combining the Similarity Scores and Confidence Values to Predict the Actual Author and Compute the Overall Confidence. The final step in predicting the candidate authors is to use the similarity scores that were calculated in Section 3.2 and the confidence values that were calculated in Section 3.2 to generate a cumulative similarity score per candidate author and an overall confidence value for the model. To compute the cumulative score, we normalize each author's similarity score in each modality by multiplying it by the corresponding confidence value, then we sum all three scores together. The most plausible author is the one with the maximum combined, normalized score. For example, assuming that we have two authors: c_1 and c_2 and that the similarity scores for c_1 on the lexical, character, and POS level are [3, 1, 5], respectively, while the similarity scores for c_2 on the lexical, character, and POS level are [2, 4, 2], respectively, and the model's confidence scores on the lexical, character, and POS level are [0.6, 0.84, 0.5], respectively, then the cumulative score for c_1 equals (1.8 + 0.84 + 2.5) = 5.14, and the cumulative score for c_2 equals (1.2 + 3.36 + 1) = 5.56. Based on the cumulative similarity scores, c_2 is the most plausible author.

It is important to notice that, although c_1 has higher similarity scores on the lexical and POS levels, c_2 has a higher cumulative similarity score. This is because the confidence values are used to weight the similarity scores, where the model gives a higher weight for the modalities in which it has higher confidence.

To choose the overall confidence of the model, we take the maximum confidence among the set of confidence scores for the modalities whose prediction matches the predicted author. For example, consider the same example from above where the lexical, character, and POS normalized scores for c_1 are [1.8, 0.84, 2.5] and for c_2 are [1.2, 3.36, 1]. If we consider each modality separately, then the most plausible authors on the lexical, character, and POS levels are c_1, c_2, and c_1, respectively. Because we only look at the modalities whose prediction matches the predicted author based on the cumulative score, we only consider the confidence of the character-level modality. In this case, the overall confidence is $max(0.84) = 0.84$ or 84%.

Consider the same example from above again, where the lexical, character, and POS normalized scores for c_1 are [1.8, 0.84, 2.5] and for c_2 are [1.2, 3.36, 1], only this time assume that c_1 is the one with the highest cumulative similarity score. In this case, the modalities whose prediction matches the predicted author are the lexical and the POS modalities. In this case, the overall confidence would be $max(0.6, 0.5) = 0.6$ or 60%.



Similarity scores per author	Evidentiary Gram
0.0000368865 0.0000969589 0.0001530929	من
0.0573620323 0.0000000000 0.0150952717	.
0.0012470007 0.0038241355 0.0000000000	انت
0.0000000000 0.0000000000 0.0238885163	قبل
0.0006235004 0.0057362032 0.0000000000	كل
0.0098669958 0.0000000000 0.0000000000	شيء
0.0049334979 0.0000000000 0.0000000000	شيء .

Fig. 4. A sample of features' scores per author. The score for author 1 is the first column from the right.

Visualizing the Result of the Attribution Process. As discussed earlier, the role of an authorship attribution domain expert in courts of law is to present their findings, i.e., the most plausible author of the investigated text, and explain how these findings were reached. It is not the expert's role to make an accusation or a decision on the case; their role is merely to present the findings to the judge or the jury members who usually come from various backgrounds. Therefore, the presentation of the results should be clear and easy-to-understand to help officers of the law make a decision.

The visualization tool provided by Ding et al. [19] was initially designed for English emails. We adapted it to work with both Arabic and English, while no modifications were required for it to work with tweets. Specifically, we used Microsoft Translator[7] to detect the language used in the anonymous text, and based on the results we changed the POS tagger and the text direction in text boxes. We do not translate POS tags to Arabic for readability issues. This automatic detection of language allows for an easier incorporation of other languages by simply including the POS tagger in the source code and specifying the text direction for the added language.

The approach of using the Hue, Saturation, and Lightness to encode the scores and calculate the value per parameter is explained in detail by Ding et al. [19] and will be omitted from this article for brevity.

Given a set of C candidate authors and an anonymous text, we provide the user with four outcomes:

(1) *A tuple of C-scores for each feature.* For each feature that was used, on all three modality levels, a score is provided for each author based on their writing samples. These values are calculated as explained in Section 3.2, and a sample of the output is provided in Figure 4. The highlighted feature is the word "قبل". This word appears only for author 1, and therefore it has scores of 0 for the other two authors, because its frequency in their writing samples is 0.

(2) *The authors' scores projected on the anonymous text.* As Figure 5 shows, the whole authors' profiles are projected on the anonymous text to allow for visual comparison between the candidate authors. A sentence in the anonymous text is represented with three lines: The line in the middle shows the sentence as it is and is used to show the word-level n-grams. The HSL for a word is modified to reflect its score for a particular author. The upper line looks empty; however, it is used to reflect the scores of character-level n-grams. Finally, the lower line contains the corresponding POS tag for each word[8] and is used to reflect the scores of POS-level n-grams. Figure 5 shows the similarity between an anonymous text and three authors' profiles: Author 1, Author 2, and Author 3. The figure suggests

[7] https://msdn.microsoft.com/en-us/library/dd576287.aspx.

[8] Padding was applied in case a word is shorter than its POS tag to prevent overlapping between tags.

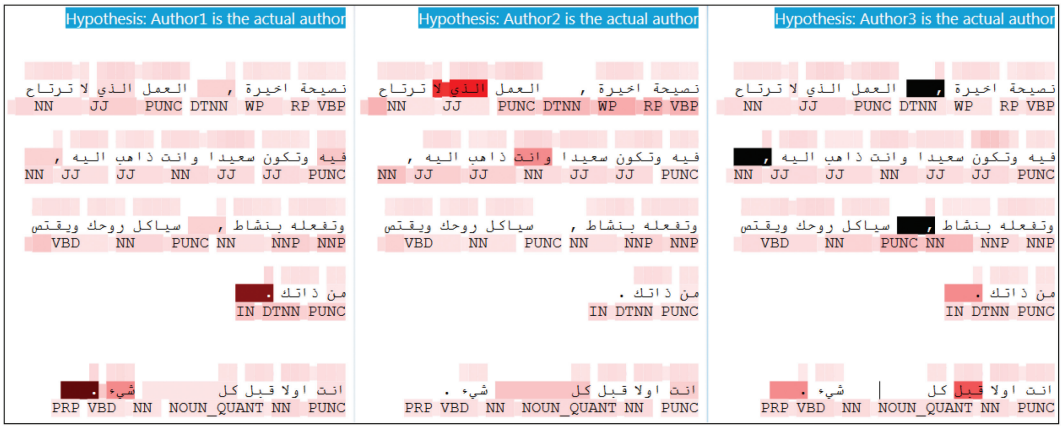


Fig. 5. The results of the attribution problem for three authors and 25 tweets each.

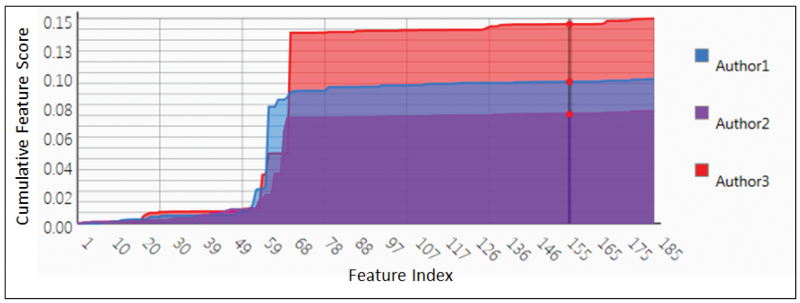


Fig. 6. A cumulative feature score. The area under the curve represents the difference in authors' scores.

that Author 3 is most plausible author as it shows a high score (represented with a dark highlight) for using commas, the word "قبل" that appears only for Author 3 and does not appear for the other two authors and the use of full stops.

(3) *A cumulative features score.* As shown in Figure 5, both Author 1 and Author 2 have similarities to the anonymous texts, but from the number of similar features and the color intensity, the user is expected to identify the most plausible author. However, as the number of candidate authors and the features increase, it is expected that this identification task becomes harder.

Figure 6 presents a cumulative feature score for each author that is based on adding the score of each individual feature, starting with the feature that appears first in the anonymous text. These features include character-, word-, and POS-level features. As shown in the figure, all the scores start and increase by the same level, which can be verified using Figure 5. For the first few features (until feature 20) the similarity scores are low for all the authors, then (at feature 55) and as more features are seen, the similarity scores are clearly different. Even before reaching feature 68, it is clear that Author 3 has accumulated the highest similarity score.

(4) *The prediction and the confidence based on each modality level separately.* Finally, we provide the typical output of an attribution process. Figure 7 shows the prediction from each modality level and its confidence. The overall prediction is chosen based on the highest

Modality	Prediction	Confidence	
Character	Author3	0.734	
Lexical	Author3	0.676	
Syntatic	Author3	0.400	
Over all :	Author3	0.734	

Fig. 7. The most plausible author and the confidence from each modality level.

normalized score as explained in Section 3.2, and the overall confidence is the maximum confidence for the modalities whose prediction match the overall prediction. In this example, all the modalities predictions match the overall prediction, and the overall confidence is the maximum of all the confidence values.

4 EXPERIMENTAL DESIGN

4.1 Dataset

In this section, we explain the process of collecting tweets from Twitter to create one big dataset of Arabic short texts and the sampling procedure that we followed to create smaller subsets to be used in the experiments.

Data Collection. Twitter is a social website that allows its users to share their status updates on their timelines. Each status update, known as a tweet, is limited to 140 characters and is submitted to a user's timeline. A timeline is a collection of tweets listed in a reverse chronological order, i.e., the most recent tweet is shown first.

We wrote a script to communicate with Twitter and gather tweets. This was important due to the lack of a public dataset of Arabic short text that we could use in our research. To build our dataset of tweets, we needed a list of authors for whom the tweets would be collected. In a real-life scenario, a law enforcement officer is likely to have a set of suspects in question, created using their common investigation techniques. As we do not have a similar list, we needed to create our own. As discussed later in the experimental setup, we could collect lots of tweets for some users on Twitter. However, we aimed for a more challenging scenario where the number of tweets in a user's account is small.

We started by retrieving a set of random tweets that are written in Arabic. These tweets were a mixture of Arabic and non-Arabic tweets such as *"Farsi"* or Urdu, because these languages use very similar character sets. We extracted only the "user" information from each tweet and ignored everything else. We repeated this step multiple times, each time keeping only the user information until we collected a list of around 160 different usernames.

Next, for each username in the created list, we retrieved a set of tweets from the user's timeline and filtered out tweets that contained only a hyperlink or an emoji. We also replaced all the usernames and hashtags in the tweets' bodies with the "@" symbol and the "#," respectively. This was necessary, because (1) these elements are not part of the writer's style but the style of the original creator of these usernames or hashtags [39] and (2) usernames can reveal an author's social network, which can give a strong indication of who the real author is. Our goal was to reach 2,000 tweets per author, but most of the retrieved authors had much fewer than that. We manually inspected every author's tweets looking for non-Arabic ones. An author whose tweets were not in Arabic was removed from the dataset. We stored the authors' names and their tweets, along with the tweets' features. This was important for reducing the running time of the experiments, because it is likely for a tweet to be used in multiple experiments.

Table 2. Descriptive Statistics for the Dataset

(a) The whole dataset.

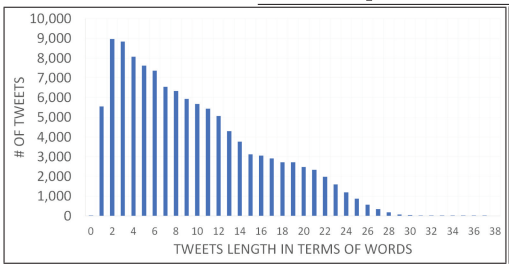
# of authors (A)	= 155
# of tweets (T)	= 115,786
Average number of tweets per author T/A	≈ 747
Creation time span for the collected tweets	01-Feb-2011 to 01-Oct-2016

(b) Tweets.

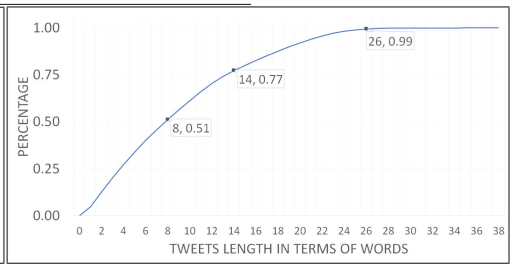
Min # of words	= 0
Max # of words	= 37
Average number of words per tweet	≈ 9.6
% of tweets with less words than the average	≈ 56%

(c) Diacritics.

% of tweets with diacritics	≈ 40%
Total number of diacritic occurrences	= 130,512
Average number of diacritic occurrences per tweet	≈ 1.1
Most frequent diacritic	Hamzat Fateh أ
Least frequent diacritic	Madda ~



(d) Histogram: # of emails vs. # of words.



(e) Empirical distribution function.

After preprocessing the retrieved tweets, 155 Twitter users remained in the list, and 115,786 tweets were collected with an average of 747 tweets per user. Table 2 shows some descriptive statistics of the dataset. The top 100 most frequent function words are provided in Appendix A.

Modern Standard Arabic vs. Colloquial Arabic. On manual inspection of the nature of the collected tweets, we noticed that the tweets are written in a mixture of the Modern Standard Arabic (MSA) and colloquial Arabic, with the majority of the tweets being in Modern Standard Arabic.

The Effect on the Instance-Based Approach. Being in MSA or colloquial Arabic would not affect the process of extracting the lexical or the structural features. It would, however, affect some of the syntactic features. We have three categories of syntactic features: diacritics, punctuation, and function words. Diacritics can appear in both MSA and colloquial Arabic. They are not mandatory for writing in MSA and in colloquial Arabic they can be used for text decoration. This indicates that writing in MSA or colloquial Arabic will not guarantee, nor will it eliminate the appearance of diacritics in a tweet. The same scenario applies for punctuation.

In the case of function words, there are two cases: The first one is when a word is the same in both MSA and colloquial Arabic. Examples of this case are the names of months or currency. In this case, the word would be captured by the proposed function-word features. The other case is when a word is only used in colloquial Arabic, such as the ordinal word "تالت" (pronounced as talet), which means "third." In this case, it will not be captured as the function word "ثالث," (pronounced as thaleth).

The Effect on the Profile-Based Approach. As described earlier, the profile-based approach does not look for certain features. Instead, the features are generated from the writing samples. In our case, the *n*-gram features are extracted from the anonymous tweet, and then the same *n*-grams are extracted from the writing samples for each candidate author. This makes the profile-based approach indifferent of the nature of the language used. In fact, it has an advantage over the instance-based approach, since an instance-based approach is likely to miss typos that appear in the text unless they are hard-coded as features. In case all the words in the text are collected as features (i.e., collecting word 1-grams) then such typos will be caught if no selection of the top k features was applied or if the typo occurs very frequently in the text.

4.2 Experimental Setup

We ran all our experiments[9] on small datasets containing between 2 and 20 authors, each with 25 writing samples. Our decision of conducting this study on a small number of candidate authors is justified by the real-world application of authorship attribution. In most cases, a law enforcement officer or the plaintiff of a civil case has a small number of candidate authors for a piece of anonymous text. Although some of these candidate authors may be very prolific on Twitter and have many writing samples, we aimed for a more challenging scenario where the number of writing samples per author is small. Our experimental setting simulates real-life scenarios of conducting authorship attribution. This setting ensures that the reported results are realistic, and the accuracy is not overestimated.

This decision was also justified by the work of Luyckx and Daelemans [44] and Ding et al. [19]. Luyckx and Daelemans [44] have worked on students' essays authorship analysis and highlighted that the accuracy of the authorship attribution process begins to drop significantly as the number of authors increases beyond two. Ding et al. [19] have conducted a set of experiments on emails authorship analysis to compare the performance of their proposed approach to the performance of SVM and DT classifiers. They conducted their experiments for 2, 5, 10, and 20 authors and the results of these experiments agree with the findings of Luyckx and Daelemans [44]. Given that an average tweet is much shorter than an email, we limited the number of authors in our experiments to 20 authors.

Similarly to Reference [19], we conducted the experiments on groups of 2, 5, 10, and 20 candidate authors. For each group, we sampled five mini-datasets (a, b, c, d, and e) containing the same number of authors. Sampling for each mini-dataset was done without replacement, while sampling across datasets was done with replacement. For example, for a group of 10 authors in a certain experimental setting, we sampled five mini-datasets (a, b, c, d, and e), where each mini-dataset contains 10 authors. Author x may appear in any dataset only once (without replacement) but may appear in one or more datasets (with replacement). Each experimental setting was repeated 10 times. In each time, the 10-fold cross-validation approach was used to divide the dataset into training and validation sets. The goal of such configuration is that we test with various writing styles for the same number of authors and, hence, reduce the variance in the reported results. Table 4 shows that the difference in performance for the profile-based approach on the mini-datasets at the $\alpha = 0.05$ is significant for all the groups of candidate authors ($p < 0.01$).

The results are reported as the average of 50 runs (10-folds are considered 1 run), which is calculated as the average of the 10 runs for each one of the five mini-datasets (a, b, c, d, and e). The

[9] All experiments were conducted on a workstation running Windows 7 (64-bit) on an Intel Core i7-4700HQ CPU @ 2.40GHz (eight CPUs) with 16GB RAM.

Table 3. Implementations of the Classification Algorithms and Their Parameters

Algorithm	Implementation	Param. changed from default
Naïve Bayes	*.bayes.NaïveBayes	—
SVM	*.function.LibSVM	kernel: Radial Basis
Decision Trees	*.trees.J48	—
Random Forests	*.trees.RandomForests	—

*Available under WEKA.Classifiers.

outcomes of an experimental setting are four accuracy values for four datasets containing 2, 5, 10, and 20 candidate authors. Each value resembles the percentage of correctly classified tweets using a certain classifier, i.e., the predicted author for that tweet is the actual author.

We used WEKA's implementation of the classification algorithms in Section 3.1. Table 3 shows the implementation of each algorithm and the parameters that were changed from WEKA's default settings. Among the four implementations, LibSVM is the only model that is not available directly in WEKA's package and had to be included manually.

5 RESULTS AND DISCUSSION

As described in Section 1.2, the aim of these experiments is to answer four research questions (RQ) focusing on the performance of the *n*-gram approach. However, due to the lack of research on Arabic authorship analysis and for the sake of completeness, we extended our analysis to include the behavior of the instance-based classification techniques.

5.1 RQ1. How Does the Performance of the *N*-Gram Approach Compare to State-of-the-Art Instance-Based Classification Techniques?

To answer this question, we have to study the effect of three Independent Variables (IV) on the Dependent Variable (DV), which is the accuracy of the attribution process. These variables are the number of candidate authors, the number of tweets per author (i.e., the writing samples), and the text size for both the writing samples and the anonymous tweet.

5.1.1 Increasing the Number of Candidate Authors. We consider this to be the baseline scenario with 25 tweets per candidate author and without specifying a condition on the minimum number of words per tweet. We ran this experiment on datasets containing 2, 5, 10, and 20 candidate authors as per the experimental setup described in Section 4.2. The results of this experiment are shown in Figure 8.

Figure 8 shows that the accuracy dropped for all the attribution techniques as the number of candidate authors increased. An Analysis of Variance (ANOVA) test was conducted at the $\alpha = 0.05$ level to determine the significance of this change, where the two factors are the classification technique and the number of candidate authors in a dataset. The results showed that both independent variables, increasing the number of authors and using different attribution techniques, had significant effects on the accuracy of the attribution process (DV).

To compare the different numbers of candidate authors, we conducted an ANOVA: single factor[10] and the result showed that there is a significant effect of increasing the number of authors (IV) on the accuracy of the attribution process (DV) at the $\alpha = 0.05$ level. We conducted post hoc comparisons using the Tukey HSD[11] test at the $\alpha = 0.05$ level to compare the four conditions. The

[10]Conducted online, using http://astatsa.com/OneWay_Anova_with_TukeyHSD/.
[11]See footnote 10.

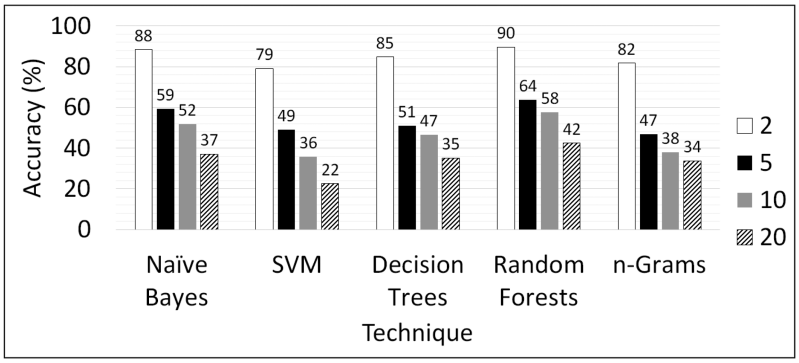


Fig. 8. Baseline scenario: instance-based (NB, SVM, DT, and RF) vs. profile-based (n-gram) for 2, 5, 10, and 20 authors.

results showed that there was a significant difference in the accuracy when the number of candidate authors increased from 2 authors to 5, 10, and 20 authors, as well as from 5 to 20 authors. However, when the number of authors increased from 5 to 10 and from 10 to 20, the difference was insignificant. The Mean, the Standard Deviation (SD), and the results of the ANOVA test are summarized in Table 5.

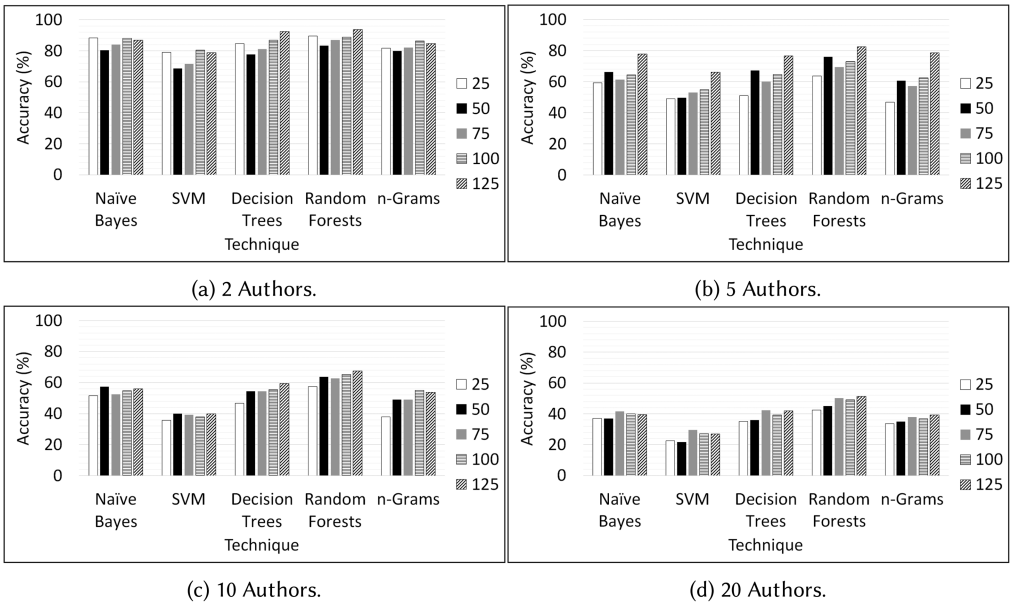
As we are interested in the performance of the n-gram approach, we conducted four paired two-sample t-tests at the $\alpha = 0.05$ level. In each test, we compared the accuracy of the n-gram approach to one of the instance-based classifiers. The results showed that there is an insignificant difference in the accuracy when the n-gram approach is compared to either SVM or DT. In contrast, the n-gram approach performed significantly worse than NB and RF, respectively. The Mean, SD, and results of the ANOVA test are summarized in Table 6.

5.1.2 Increasing the Number of Tweets per Author. In this experiment, we increased the number of tweets per author from 25 (the baseline) to 125 tweets in steps of 25. We wanted to see if the increase in the number of tweets would help the attribution techniques perform better as the number of candidate authors increased. We did not increase the number of tweets per author beyond 125 to keep the scenario realistic, as suggested in Reference [43]. Figure 9 shows the results of this experiment where each set of authors was tested with 25, 50, 75, 100, and 125 tweets per author.

We conducted four one-way ANOVA tests, one for each number of candidate author, at the $\alpha = 0.05$ level to test whether the increase in the number of tweets per candidate yielded a significant change in the performance of the attribution process (Mean and SD are provided in Table 7). The outcome of these four ANOVA tests showed that the change in the performance (DV), which was caused by increasing the number of tweets per authors (IV), was significant only for 5 candidate authors while being insignificant for 2, 10, and 20 candidate authors.

A post hoc Tukey test was conducted on the set of five candidate authors to see which increase in tweets per author had a significant effect. The results of this test showed that among the 10 possible pairwise comparisons only two were significant: increasing the number of tweets from 25 (the baseline) to 125 (the maximum number of tweets) [$p = 0.001$] and from 75 to 125 [$p = 0.02$].

Based on these results, we noticed that a larger number of additional tweets was needed to have a statistically significant increase in the performance; however, this increase was limited to five authors. As the number of authors increased beyond five, having 125 tweets per author was not significant. As mentioned, increasing the number of tweets to more than 125 tweets per author



(a) 2 Authors. (b) 5 Authors.

(c) 10 Authors. (d) 20 Authors.

Fig. 9. Increasing the number of tweets per candidate author (from 25 to 125).

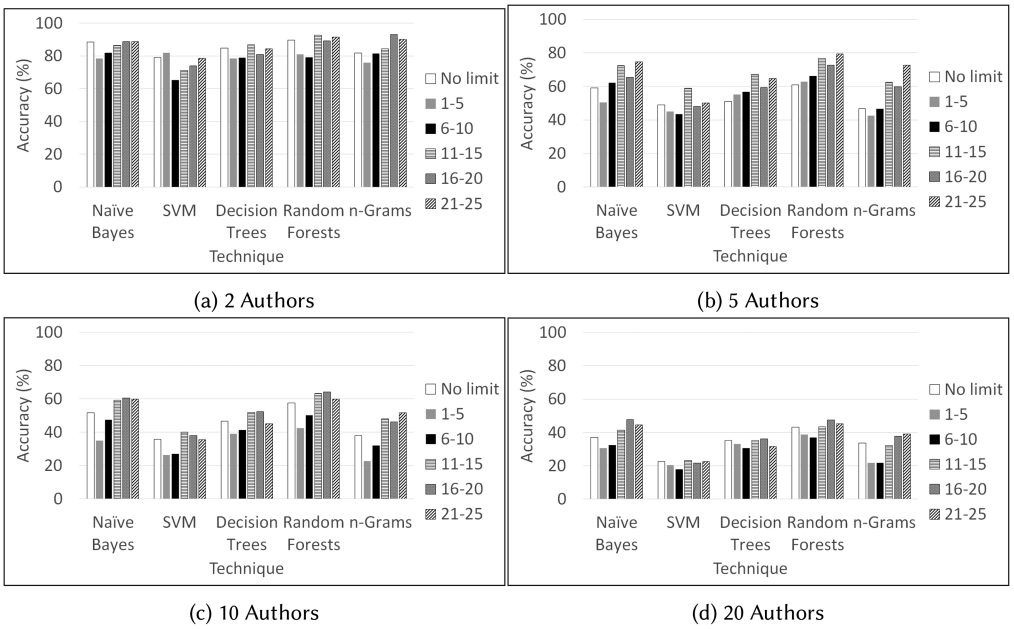
is unrealistic [43]. Furthermore, increasing the number of tweets per author will lead to a longer time span that covers these tweets, and so they are more likely to cover a larger number of topics. As new topics emerge constantly in daily life, it is very likely that the authors' styles have adapted to these topics; therefore, introducing more tweets could have a negative effect on the attribution. These findings agree with the work of Bhargava et al. [10].

As we are interested in the performance of the n-gram-based approach, we conducted four ANOVA: single Factor t-tests, one for each set of candidate authors, at the $\alpha = 0.05$ level (the mean and SD are shown in Table 8). All four t-test results showed that the performance of the various attribution techniques is significantly different.

Using four post hoc Tukey tests, we investigated the significance of the difference in performance between the n-gram approach versus other instance-based algorithms. The results of these tests showed that for all four tests there is no significant difference between using the n-gram approach versus using Naïve Bayes or DT. As for SVM, the n-grams approach performs significantly better only when the number of authors is 10 and 20. Finally, the difference between Random Forests and using n-gram is insignificant for 2 and 5 authors, but when the number of authors increases to 10 or 20 Random Forests performs significantly better than the n-grams approach. Below is a summary of the post hoc Tukey test results (Table 8).

Specifying the Minimum Number of Words per Tweet. The goal of this experiment is to compare the performance of the n-gram approach to that of instance-based algorithms when the size of the anonymous text changes, whether for the anonymous text, i.e., the number of words in the anonymous text increases, or for the writing samples of each candidate author.

We set the baseline to be 25 tweets per author, with no conditions on the word count for each tweet. To compare the performance of the different algorithms, we sampled five additional datasets in which we randomly sampled 25 tweets per author, where the range of word count for the sampled tweets is [1–5], [6–10], [11–15], [16–20], or [21–25] words per tweet. These datasets were sampled for 2, 5, 10, and 20 authors. The results of this experiment are shown in Figure 10.



(a) 2 Authors (b) 5 Authors

(c) 10 Authors (d) 20 Authors

Fig. 10. Specifying the minimum number of words per tweet.

Figures 10 shows that there is an increase in the performance when the size of tweets increases. To test the significance of this increase we run four ANOVA: single Factor t-tests, one for each set of candidate authors, at the $\alpha = 0.05$ level. The results of these four tests showed that the difference is insignificant for 2 and 20 authors, while being significant for 5 and 10. On further inspection of the significance of the results for 5 authors using a post hoc Tukey test, we noticed that the differences for all the pairwise comparisons were insignificant. This agrees with Reference [64], where it is explained that it is possible to have a significant F-score while having insignificant post hoc test p-values. In contrast, the post hoc Tukey test for 10 authors shows that the increase in the performance when the range of tweets' length increased from 1–5 to 6–10 or to 11–15 is significant. The detailed results are presented in Appendix C in Table 9.

To evaluate the performance of using n-grams compared to other classifiers, we looked at the F-scores of four ANOVA: single Factor t-tests at the $\alpha = 0.05$ level. The scores showed that for all four number of author settings, the difference among the classification techniques is significant. To identify the significant pairwise comparisons, we ran four post hoc Tukey tests (detailed results are in Appendix C, in Table 10). The results of the Tukey tests showed that, except for Random Forests, using n-grams is either the same (i.e., the difference is insignificant) or better than using the other classification techniques. Only Random Forests performed significantly better than n-grams, and that was when the number of authors was 5, 10, and 20. When the number of authors was 2, the difference between Random Forests and n-grams was insignificant.

Merging Tweets into Artificial Tweets. In this last attempt to evaluate the performance of the attribution techniques, we try to address the problem of the small text size by merging groups of five tweets into single artificial tweets. Based on that, the 25 tweets per author that were used in the "Increasing the number of tweets per author" experiment (Section 5.1) are grouped into 5 artificial tweets, where each artificial tweet is created by concatenating the text of the 5 tweets.

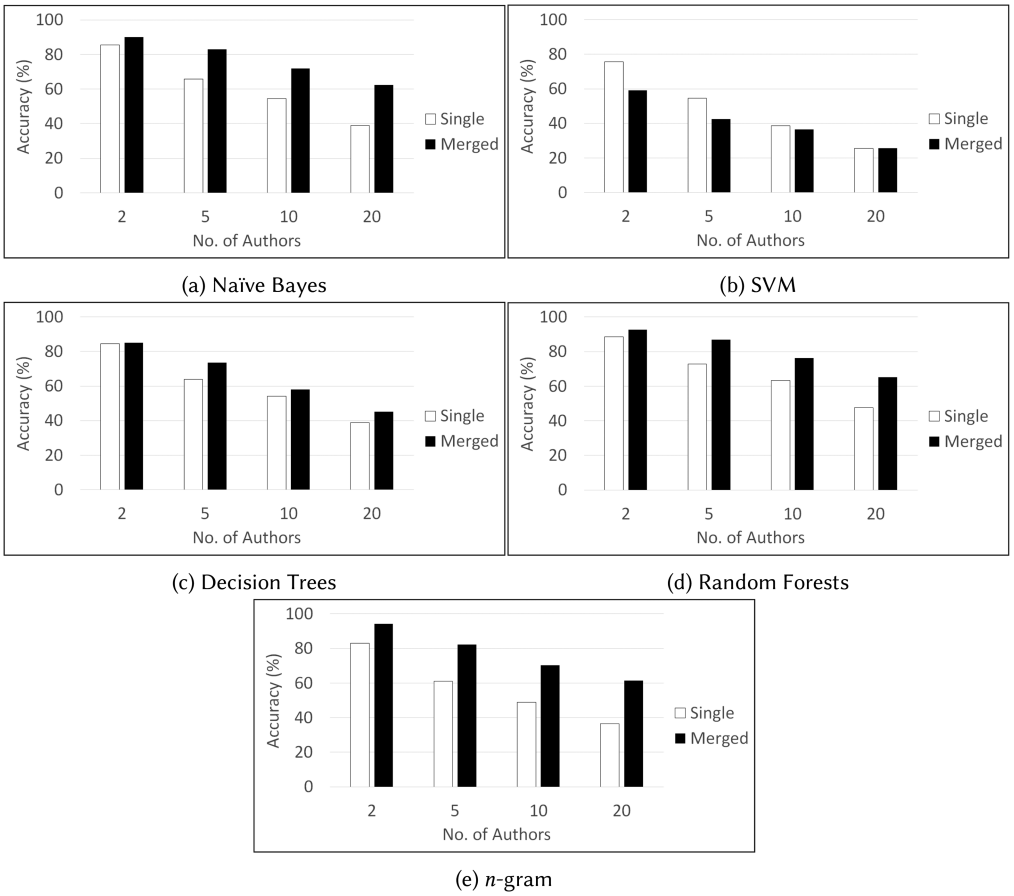


Fig. 11. Merging tweets into groups of five tweets.

The resulting experimental setting is the following: Similarly to Section 5.1, we use sets of 2, 5, 10, and 20 authors with 5, 10, 15, 20, and 25 artificial tweets in each experiment. These artificial tweets are generated from 25, 50, 75, 100, and 125 tweets, respectively; the exact same tweets that were used in Section 5.1. The reason we used the same tweets is to have one variable in the new experiment, which is merging the tweets into groups. Had we used a new set of single tweets, and then merged them for this experiment, that would have generated some bias based on the content of the new tweets, even if we controlled the text size over the selected tweets. Therefore, we kept the tweet ID for the tweets used in the aforementioned experiments and then used the IDs to group the tweets into artificial tweets.

Similarly to previous experiments, we start by looking at the performance of the various attribution techniques. We use 2, 5, 10, and 20 authors with 5, 10, 15, 20, and 25 artificial tweets per author. As every 5 tweets are grouped into one artificial tweet, this is equivalent to 25, 50, 75, 100, and 125 tweets per author. The results of these experiments are shown in Figure 11. We report the accuracy of a certain classification technique as the average of its performance for the varying number of tweets per author, i.e., the average performance for 5, 10, 15, 20, and 25 tweets per author.

Merging tweets has two effects on the attribution process: First, instance-based classifiers will have fewer training examples to build a model from. However, these instances are supposedly richer in text. Second, since we are using k-fold cross-validation, the anonymous tweet will also contain richer text. This affects both instance-based and profile-based techniques.

Except for SVM, the figure shows that merging tweets into artificial ones helps classifiers achieve better results. For SVM, the results showed that using a small number of training samples will negatively affect the performance. For example, with 2 authors and 5 tweets per author, SVM had only 10 instances to train a model.

On further analysis of the results using Paired Two Sample t-tests at the $\alpha = 0.05$ level, the difference was significant for Naïve Bayes, Random Forests, and n-grams. In contrast, the difference for SVM and DT was insignificant. The detailed results are presented in Table 11.

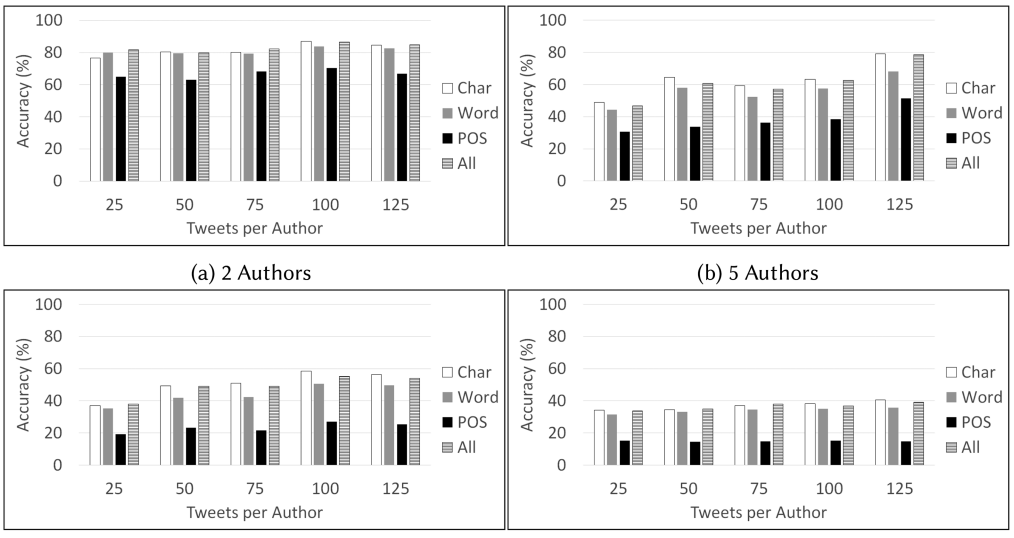
We compare the performance of n-grams to other classification techniques using ANOVA t-test at the $\alpha = 0.05$ level. The results of this t-test show a significant difference among the five techniques. On using a post hoc Tukey test, the only significant comparison was between SVM and n-grams. The detailed results are presented in Table 12. In general, our experimental result is in line with the experiments shown in References [19] and [44]. As the number of candidate authors increases, the complexity of the classification also increases, which in general leads to decreased accuracy. However, the performance of the profile-based approach with n-grams is on par with instance-based models. This indicates that there is no tradeoff between accuracy and visualization.

5.2 RQ2. Which N-Gram Level (Character, Word, or Part-Of-Speech (POS)) Is the Most Helpful in Distinguishing the Authors' Writing Styles?

In this section, we answer the second research question: Which n-gram level has the highest effect on the attribution process? As mentioned earlier, n-grams are the N consecutive tokens from a tokenized text, where the tokenization is on the character, word, or POS level. In all the previous experiments, we used all three levels of modalities, and these modalities were evaluated using a linear regression model to calculate the confidence, as shown in Section 3.2. The modality level with the highest confidence was used in predicting the candidate author. In this set of experiments, however, we evaluate each modality level separately for 2, 5, 10, and 20 authors, with 25, 50, 75, 100, and 125 tweets per author. The results of these experiments are depicted in Figure 12.

Statistical analysis using one-way ANOVA tests at the $\alpha = 0.05$ level shows a significant difference among the four modality levels (character, word, POS, or all of levels together). We further analyzed the results using post hoc Tukey tests, and the results showed that there is an insignificant difference between using either character-level or word-level modalities or all three levels of modalities combined. In contrast, the difference was significant when only POS n-grams were used. The results of the ANOVA test and the Tukey tests are provided in Tables 13 and 14, respectively.

Generally, the experimental result is in line with the results reported in Reference [19]. They showed that lexical modality performs the best for English. However, for Arabic text, we find that character modality performs better than lexical modality with respect to the mean value. Statistically, their difference is insignificant. We suspect that for Arabic, matching n-grams is harder than English, as Arabic tends to merge pronouns with words instead of separating them [42]. For example English uses "his book" and "her book," which translates to "كتابه" and "كتابها" in Arabic. If we look at 1-grams for these sentences, then we have three grams for English: "his," "her," and "book"; and two grams for Arabic: "كتابه" and "كتابها". As noticed in Arabic, the word "book" was distributed over 2 grams and could be distributed over more grams, depending on the pronouns attached to it. However, using character modalities, specifically, using 4-grams, the word "book" i.e., "كتاب" will be matched to the same gram in both cases.



(a) 2 Authors (b) 5 Authors

(c) 10 Authors (d) 20 Authors

Fig. 12. Evaluating each n-gram modality separately.

5.3 RQ3. How Important Are Diacritics to the Attribution Process When the N-Gram Approach Is Used?

The use of diacritics is one of the major morphological properties that make the Arabic language different from English, and this prevents authorship attribution techniques that are developed for English from being used in Arabic. The use of diacritics in Arabic is optional both in Modern Standard Arabic and Colloquial Arabic. In this set of experiments, we aim to see whether removing diacritics before the attribution process or keeping them would have an effect on the outcome. As a reminder, 40% of the tweets in our dataset contain diacritics (see Section 4).

We use the same experimental setup as the previous experiments: 2, 5, 10, and 20 authors with 25, 50, 75, 100, and 125 tweets per author. We evaluate the diacritics effect on character- and word-level modalities but not POS. This is because diacritics should be removed before retrieving POS tags. The results of these experiments are shown in Figure 13.

As shown in the figure, removing diacritics barely has any effect on the attribution process. We verify that using four one-way ANOVA tests at the $\alpha = 0.05$ level. The results of these tests confirm that the difference for the different modality levels (including POS) are insignificant for all four sets of authors. Mean and SD for the experiments with and without diacritics are provided in Table 15(a) and Table 15(b), respectively. The results of the ANOVA tests are provided in Table 16.

5.4 RQ4. When Using Classification Techniques, How Important Is It to Use All Three Categories of Stylometric Features (Lexical, Structure, Syntactic)?

In this section, we investigate the effect of adding more features to the attribution process. As described in Section 3, we have three categories of features, namely: lexical, structural, and syntactic features. The goal of this section is to evaluate which category (or combination of categories) gives the best performance in the attribution process. Additionally, is adding more features helpful for the attribution process or not?

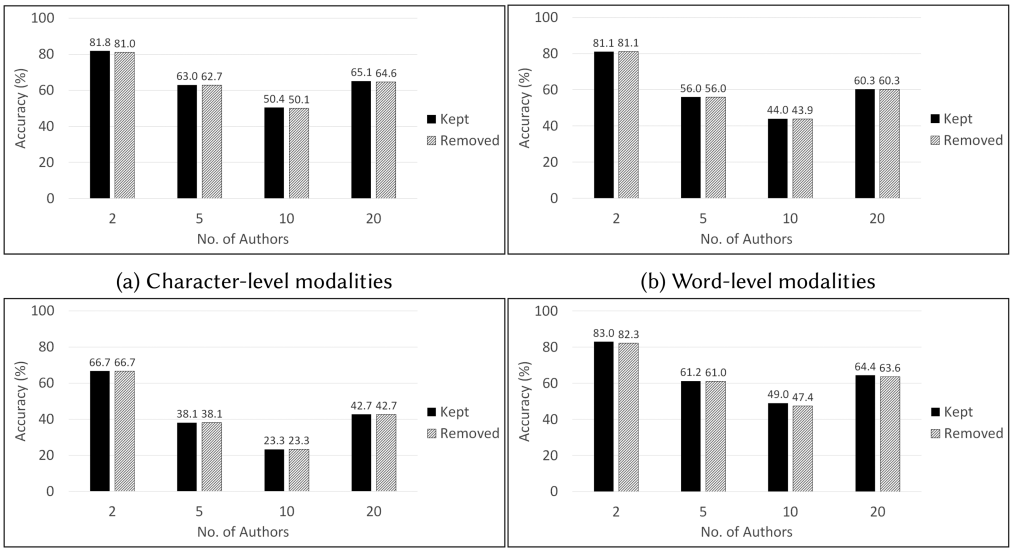


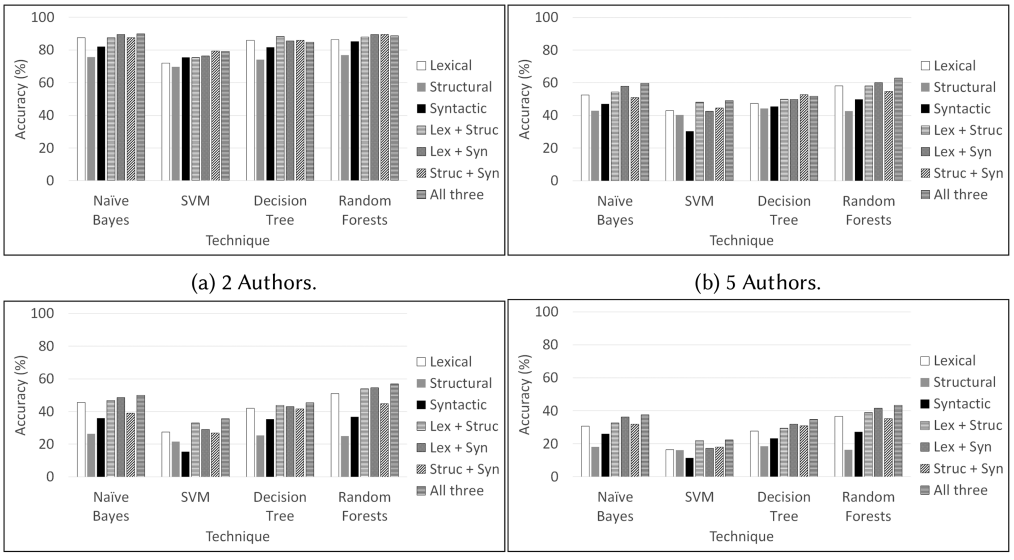
Fig. 13. Evaluating the effect of diacritics on the *n*-gram approach. The figures showed the average performance for a set of authors on 25, 50, 75, 100, and 125 tweets per author.

To do that, we performed the attribution process using instance-based classifiers for 2, 5, 10, and 20 authors with 25 tweets per author. For each set of authors we used one of the following seven sets of features: lexical (Lex), structural (Struc), syntactic (Syn), lexical and structural (Lex + Struc), lexical and syntactic (Lex + Syn), structural and syntactic (Struc + Syn), and all three sets of features. Figure 14 shows the results of these experiments.

Figure 14 shows a variation in the results as the number of authors change (Mean and SD are provided in Table 17). To help explain this variation, we used ANOVA tests at the $\alpha = 0.05$ level to evaluate the significance of difference for each set of authors.

For two authors, the difference was statistically insignificant $[F(6, 21) = 2.48, p = 0.056]$. For five authors, the ANOVA test showed a statistically significant difference $[F(6, 21) = 2.60, p = 0.047]$, but a post hoc Tukey test showed that the difference is statistically insignificant for all the possible pairwise evaluations (results are provided in Table 18). As mentioned earlier, it is possible to have contradicting results between the ANOVA and the Tukey tests due to the difference in sensitivity for each test, as explained by Simon [64]. These results are in line with literature on English, which suggests that the problem of authorship attribution is relatively easy when the number of authors is small [44]; therefore, a small number of features is enough for a classification algorithm to reach its best performance on a small number of authors (in this case, two and five authors), given enough training samples for each author.

In contrast, an ANOVA test for 10 authors showed a statistically significant difference among the seven sets of features $[F(6, 21) = 3.39, p = 0.017]$. The results of a post hoc Tukey test (provided in Table 19) showed that the difference is statistically significant only for the pairwise comparison between Structural features and using all three categories of features $[p = 0.024]$. This suggests that as the number of authors increased from 5 to 10, the classification algorithms benefited from adding more features. As Figure 14(c) shows, structural features came in last for three out of four



(a) 2 Authors. (b) 5 Authors.

(c) 10 Authors. (d) 20 Authors.

Fig. 14. Evaluating feature categories with instance-based classifiers.

classifiers. However, using other features was statistically insignificant for all the cases except for using all the feature categories together.

Finally and similarly to the first case with two authors, the difference for 20 authors was statistically insignificant for all seven feature categories. $[F(6, 21) = 2.38, p = 0.064]$. This indicates that with this large number of authors, classification algorithms are not able to perform well regardless of the number of features that are used. Based on the result of this experiment, we believe that applying authorship attribution on a large scale, i.e., with a much larger set of candidate authors, will not be possible by using classification techniques. Instead, one should investigate developing new techniques for authorship attribution or propose a new feature representation that can be used with traditional classification techniques.

6 CONCLUSION, LIMITATIONS, AND FUTURE WORK

In this work, we investigated the authorship attribution problem for short Arabic text, specifically Twitter posts. Extensive work has been done for English short texts. However, due to the morphological nature of the Arabic language, techniques developed for English are not directly applicable to Arabic. Literature on Arabic authorship attribution has focused on longer texts such as books, poems, and blog posts. None of this work tackled shorter forms such as SMS, chat, or social media posts that, by nature, are much shorter.

We investigated the performance of various classification techniques that are either instance-based or profile-based techniques. We showed that profile-based approaches, specifically using *n*-grams, are in line with state-of-the-art techniques. Although the state-of-the-art performed better in some cases, these models are very complex and cannot be used as evidence in courts of law. In contrast, profile-based approaches are simpler, and their results can be visualized in a more intuitive way, which gives them an edge to be used in court.

Among the limitations that still face the *n*-gram attribution approach is the scarcity of text, whether that is in the anonymous text or the writing samples of the authors. The effect can be

seen when very few features appear in both the anonymous text and the writing samples; therefore, it will be very hard to compare the visualized writing styles of three or four authors using a tweet with three to four words. Additionally, recent studies showed that current authorship attribution techniques capture the topic in addition to an author's writing style. This means that if the anonymous text is about a topic that is not represented in the author's writing style, then the performance of the attribution techniques will decrease drastically. Using *n*-grams on various modality levels instead of only using word-level *n*-grams partially mitigates the issue of the topic. However, there is a need for a better representation of an author's style.

This article aims at laying the foundation for future work in Arabic authorship attribution for short texts. We hope that this work will open the door for further work on Arabic to keep up with the work on other languages. In addition to investigating new techniques for style representation, such techniques should utilize the huge development in deep learning, specifically in the representation learning domain. This is because applying deep learning directly for authorship attribution will be ineffective, due to the small size of data available for training.

APPENDICES

A THE TOP 100 MOST FREQUENT FUNCTION-WORD FEATURES

#	FnWord	Freq.	#	FnWord	Freq.	#	FnWord	Freq.
1.	من	21873	34.	كيف	1084	67.	العراق	495
2.	في	17495	35.	أنا	1072	68.	نفس	482
3.	على	10446	36.	التي	1067	69.	اليمن	475
4.	لا	8467	37.	لكن	1053	70.	راح	455
5.	ما	7146	38.	غير	1050	71.	ماذا	422
6.	يا	5152	39.	أو	1020	72.	عاد	419
7.	كل	5000	40.	واحد	975	73.	دون	399
8.	عن	3792	41.	إن	968	74.	الذين	390
9.	ولا	3762	42.	قد	894	75.	بل	380
10.	انا	3538	43.	أنت	881	76.	وقت	378
11.	أن	3417	44.	إذا	844	77.	أم	366
12.	مع	3276	45.	لما	799	78.	متى	362
13.	ان	3176	46.	عند	762	79.	تحت	352
14.	هذا	2873	47.	بعض	755	80.	الآن	330
15.	مكة	2611	48.	أكثر	693	81.	حين	318
16.	كان	2262	49.	اني	686	82.	نعم	317
17.	لو	2252	50.	ليس	668	83.	بما	312
18.	هو	2156	51.	لن	665	84.	مصر	312
19.	بعد	2130	52.	مثل	663	85.	خلال	310
20.	انت	1889	53.	مساء	616	86.	رمضان	306
21.	حتى	1560	54.	هم	613	87.	فوق	304
22.	لم	1559	55.	كذا	596	88.	انتم	298
23.	إلى	1454	56.	ثم	590	89.	حول	293
24.	يوم	1449	57.	السعودية	579	90.	صار	285
25.	اذا	1313	58.	هناك	573	91.	الكويت	273
26.	قبل	1305	59.	ذلك	570	92.	هلا	271
27.	الذي	1292	60.	أحد	570	93.	جميع	271
28.	صباح	1218	61.	هنا	558	94.	مهما	261
29.	هل	1190	62.	سبحان	551	95.	عمر	254
30.	الى	1181	63.	كم	526	96.	ثاني	233
31.	بين	1172	64.	نحن	521	97.	سوريا	221
32.	هي	1168	65.	أي	510	98.	قطر	220
33.	هذه	1126	66.	أول	499	99.	الرياض	200
						100.	ضمن	199

B FUNCTION-WORD FEATURES WITH ZERO USAGE

#	FnWord	#	FnWord	#	FnWord	#	FnWord	#	FnWord
1.	اللتان	31.	لكنْ	61.	آذار	91.	حادي	121.	غداة
2.	هذان	32.	هلاّ	62.	أفريل	92.	عاشر	122.	ذيت
3.	هاتان	33.	حم	63.	جانفي	93.	سنتيم	123.	كأين
4.	ذانك	34.	أيان	64.	جوان	94.	ليرة	124.	كأيّن
5.	تانك	35.	برح	65.	جويلية	95.	مليم	125.	بَسْ
6.	اولئك	36.	استحال	66.	شباط	96.	يوان	126.	بطآن
7.	ايانا	37.	أوشك	67.	فيفري	97.	ثلاثاء	127.	بَلْهَ
8.	إياي	38.	اخلولق	68.	كانون	98.	بغتة	128.	حَذارِ
9.	اياي	39.	حرى	69.	اثنا	99.	تعسا	129.	حيَّ
10.	إياكن	40.	هبَّ	70.	اثني	100.	دواليك	130.	رويدك
11.	اياكن	41.	طفق	71.	اربعون	101.	سحقا	131.	شَتَّانَ
12.	إياكما	42.	أنشأ	72.	أربعمئة	102.	سمعا	132.	عمّان
13.	اياكما	43.	أخذ	73.	أربعمائة	103.	عيانا	133.	طربلس
14.	أنتن	44.	انبرى	74.	تسعمئة	104.	قاطبة	134.	الجزائر
15.	أنتما	45.	ابتدأ	75.	تسعون	105.	كَثِيرًا	135.	مورتانيا
16.	إياهن	46.	كلَّما	76.	ثلاثمئة	106.	لَبَّيْكَ	136.	ناوكشوط
17.	اياهن	47.	لمّا	77.	ثلاثون	107.	مَعَاذَ	137.	جيبوتي
18.	اياهم	48.	أمَّا	78.	ثمانمئة	108.	هَاتِه	138.	الفجيرة
19.	إياهما	49.	أنَّي	79.	ثمانون	109.	هَاتِي	139.	عجمان
20.	اياهما	50.	إيَّان	80.	ثماني	110.	هَاتَيْنِ	140.	رأس الخيمة
21.	هنّ	51.	ايان	81.	ثمنمئة	111.	هاهنا	141.	ام القيوين
22.	كأنَّ	52.	ينبع	82.	خمسمئة	112.	هَذانِ		
23.	أنَّ	53.	تعزَّ	83.	سبعمئة	113.	هَذِي		
24.	إنَّ	54.	دعد	84.	سبعمائة	114.	هَذَيْنِ		
25.	ولات	55.	بلخ	85.	ستمئة	115.	كلتا		
26.	إذما	56.	طلحة	86.	ستمائة	116.	ريث		
27.	إيّا	57.	إسحاق	87.	ستون	117.	لدن		
28.	بجل	58.	بورسودان	88.	مئتان	118.	آنفا		
29.	جير	59.	بورتوفيق	89.	نيف	119.	تارة		
30.	كلاّ	60.	معديكرب	90.	ثامن	120.	ضحوة		

C STATISTICAL RESULTS

Table 4. Using the ANOVA Test to Show That the Difference among the Datasets a, b, c, d, and e is Statistically Significant

(a) 2 Authors

set	Mean	SD
a	79.17	16.78
b	96.67	7.03
c	80.00	18.51
d	69.17	14.19
e	84.17	12.08
	$p < 0.005$	

(b) 5 Authors

set	Mean	SD
a	43.67	11.49
b	51.67	12.50
c	53.33	18.05
d	26.67	13.61
e	58.33	11.47
	$p < 0.005$	

(c) 10 Authors

set	Mean	SD
a	18.67	8.71
b	46.00	9.79
c	43.83	6.94
d	40.50	12.15
e	40.67	11.89
	$p < 0.005$	

(d) 20 Authors

set	Mean	SD
a	39.33	4.44
b	34.42	3.97
c	27.75	3.75
d	35.08	6.13
e	32.00	5.35
	$p < 0.005$	

Table 5. Post Hoc Tukey HSD Test with $\alpha = 0.05$ to Evaluate the Effect of Increasing the Number of Authors on the Performance (ANOVA Result: $[F(3, 12) = 249.57, p < 0.001]$)

(a) Mean and SD

# of Authors	Mean	SD
2	84.8	4.44
5	54	7.21
10	46.2	9.28
20	34	7.38

(b) t-Test results and p-values.

# of Authors	Tukey HSD p-value
From 2 to 5	$p < 0.001$
From 2 to 10	$p < 0.001$
From 2 to 20	$p < 0.001$
From 5 to 10	$p = 0.36$
From 5 to 20	$p = 0.003$
From 10 to 20	$p = 0.07$

Table 6. Paired Two-Sample *t*-Tests with $\alpha = 0.05$ to Evaluate the Significance of the Difference in the Performance of *n*-Grams vs. NB, SVM, DT, and RF [$F(4, 12) = 19.37, p < 0.001$]

(a) Mean and SD

Attribution Tech.	**Mean**	**SD**
n-grams	50.05	21.87
NB	59.07	21.62
SVM	46.58	24.26
DT	54.41	21.34
RF	63.34	19.65

(b) *t*-Test Results and *p*-value

***n*-grams vs.**	***t*-Test Results**
NB	$t(3) = -3.65, p = 0.04$
SVM	$t(3) = 1.23, p = 0.31$
DT	$t(3) = -2.78, p = 0.07$
RF	$t(3) = -4.49, p = 0.02$

Table 7. Mean and SD for 2, 5, 10, and 20 Authors with Varying Number of Tweets per Author

# of Author	**Tweets/Author**	**Mean**	**SD**	**ANOVA result**	***p*-value**
2	25	84.75	4.39	$F(4, 20) = 2.79$	$p = 0.054$
	50	78.00	5.63		
	75	81.20	5.84		
	100	86.15	3.23		
	125	87.33	6.01		
5	25	53.94	7.19	$F(4, 20) = 6.39$	$p = 0.054$
	50	63.94	9.66		
	75	60.24	6.10		
	100	63.90	6.39		
	125	76.30	6.08		
10	25	45.91	9.20	$F(4, 20) = 0.74$	$p = 0.57$
	50	52.90	8.91		
	75	51.53	8.57		
	100	53.74	9.85		
	125	55.33	10.13		
20	25	34.16	7.34	$F(4, 20) = 0.64$	$p = 0.64$
	50	34.92	8.36		
	75	40.41	7.51		
	100	38.52	7.80		
	125	39.78	8.68		

Table 8. ANOVA and Post Hoc Tukey Tests (with $\alpha = 0.05$) to Evaluate the Significance of the Difference in the Performance of n-Grams vs. NB, SVM, DT, and RF for 2 (a), 5 (b), 10 (c), and 20 Authors (d)

(a) 2 Authors (ANOVA: $[F(4, 20) = 6.20, p = 0.002]$)

Attribution Technique	**Mean**	**SD**	***n*-grams vs.**	***p*-value**
n-grams	83.00	2.59		
NB	85.60	3.36	NB	$p = 0.86$
SVM	75.74	5.35	SVM	$p = 0.09$
DT	84.60	5.56	DT	$p = 0.89$
RF	88.48	3.76	RF	$p = 0.29$

(b) 5 Authors (ANOVA: $[F(4, 20) = 3.02, P = 0.04]$)

Attribution Technique	**Mean**	**SD**	***n*-grams vs.**	***p*-value**
n-grams	61.18	11.51		
NB	65.78	7.27	NB	$p = 0.89$
SVM	54.56	6.90	SVM	$p = 0.72$
DT	63.92	9.36	DT	$p = 0.89$
RF	72.88	6.99	RF	$p = 0.23$

(c) 10 Authors (ANOVA: $[F(4, 20) = 22.89, P < 0.001]$)

Attribution Technique	**Mean**	**SD**	***n*-grams vs.**	***p*-value**
n-grams	49.01	6.79		
NB	54.34	2.83	NB	$p = 0.29$
SVM	38.52	1.77	SVM	$p = 0.007$
DT	54.08	4.63	DT	$p = 0.36$
RF	63.37	3.72	RF	$p = 0.001$

(d) 20 Authors (ANOVA: $[F(4, 20) = 33.94, P < 0.001]$

Attribution Technique	**Mean**	**SD**	***n*-grams vs.**	***p*-value**
n-grams	36.55	2.20		
NB	39.06	2.06	NB	$p = 0.67$
SVM	25.63	3.39	SVM	$p = 0.001$
DT	38.94	3.35	DT	$p = 0.70$
RF	47.62	3.75	RF	$p = 0.001$

Table 9. Mean and SD for 2, 5, 10, and 20 Authors with Specifying the Minimum Number of Words per Tweet

# of Authors	Tweets/Author	Mean	SD	ANOVA result $F(5, 24)$	p-value
2	1–5	79.11	2.42	1.87	0.13
	6–10	77.31	6.90		
	11–15	84.25	7.98		
	16–20	85.17	7.74		
	21–25	86.63	5.24		
	No limit	84.75	4.39		
5	1–5	51.19	8.13	3.49	0.02
	6–10	54.98	9.81		
	11–15	67.57	7.22		
	16–20	61.07	9.00		
	21–25	68.28	11.46		
	No limit	53.40	6.33		
10	1–5	33.15	8.37	3.29	0.02
	6–10	39.60	9.88		
	11–15	52.44	9.17		
	16–20	52.22	10.60		
	21–25	50.34	10.34		
	No limit	45.89	9.17		
20	1–5	28.89	7.74	1.15	0.36
	6–10	27.89	7.83		
	11–15	35.08	8.08		
	16–20	38.07	10.66		
	21–25	36.54	9.57		
	No limit	34.28	7.50		

Table 10. ANOVA and Post Hoc Tukey Tests (with $\alpha = 0.05$) to Evaluate the Significance of the Difference in the Performance of n-Grams vs. NB, SVM, DT and RF

(a) 2 Authors (ANOVA: $F(4, 25) = 4.84, p = 0.005$)

Attribution Technique	**Mean**	**SD**	***n*-grams vs.**	***p*-value**
n-grams	84.42	6.29		
NB	85.47	4.33	NB	$p = 0.90$
SVM	74.97	6.18	SVM	$p = 0.04$
DT	82.33	3.48	DT	$p = 0.90$
RF	87.17	5.67	RF	$p = 0.89$

(b) 5 Authors (ANOVA: $F(4, 25) = 5.60, P = 0.002$)

Attribution Technique	**Mean**	**SD**	***n*-grams vs.**	***p*-value**
n-grams	55.11	11.71		
NB	64.03	8.91	NB	$p = 0.36$
SVM	49.12	5.39	SVM	$p = 0.69$
DT	59.01	6.04	DT	$p = 0.90$
RF	69.80	7.60	RF	$p = 0.04$

(c) 10 Authors (ANOVA: $F(4, 25) = 7.04, P < 0.001$)

Attribution Technique	**Mean**	**SD**	***n*-grams vs.**	***p*-value**
n-grams	39.79	11.00		
NB	52.20	9.92	NB	$p = 0.10$
SVM	33.77	5.71	SVM	$p = 0.71$
DT	46.01	5.34	DT	$p = 0.67$
RF	56.25	8.40	RF	$p = 0.02$

(d) 20 Authors (ANOVA: $F(4, 25) = 15.57, P < 0.001$)

Attribution Technique	**Mean**	**SD**	***n*-grams vs.**	***p*-value**
n-grams	31.02	7.60		
NB	38.92	6.75	NB	$p = 0.09$
SVM	21.30	1.95	SVM	$p = 0.02$
DT	33.61	2.21	DT	$p = 0.90$
RF	42.42	3.98	RF	$p = 0.01$

Table 11. Paired Two-Sample t-Tests with $\alpha = 0.05$ to Evaluate the Significance of the Difference in the Performance of NB, SVM, DT, RF, and n-Grams When Groups of 5 Tweets are Merged into One Artificial Tweet

Classification Technique	Tweets	Mean	SD	t-test	p-value
NB	Single	61.22	19.60	−3.93	$p = 0.03$
	Merged	76.86	12.22		
SVM	Single	48.61	21.61	1.94	$p = 0.15$
	Merged	40.89	14.02		
DT	Single	60.38	19.14	−2.65	p =0.08
	Merged	65.38	17.55		
RF	Single	68.09	17.13	−4.32	$p = 0.02$
	Merged	80.23	12.18		
n-gram	Single	57.43	19.79	−6.66	$p = 0.006$
	Merged	77.07	14.23		

Table 12. Post Hoc Tukey HSD Test with $\alpha = 0.05$ to Evaluate the Effect of Merging Groups of Five Tweets into One Artificial Tweet on the Performance (ANOVA Result: $[F(4, 15) = 5.30, p = 0.007]$)

Attribution Technique	Mean	SD	n-grams vs.	p-value
n-gram	77.07	14.23		
NB	76.86	12.22	NB	$p = 0.90$
SVM	40.89	14.02	SVM	$p = 0.02$
DT	65.38	17.55	DT	$p = 0.74$
RF	80.24	12.18	RF	$p = 0.90$

Table 13. ANOVA *t*-Test with $\alpha = 0.05$ to Evaluate the Different *n*-Grams Modalities

(a) Mean and SD

# of Authors	Modality	Mean	SD
2	All	83.0	2.6
	Char	81.8	4.0
	Word	81.1	2.0
	POS	66.7	2.9
5	All	61.2	11.5
	Char	63.0	10.9
	Word	56.0	8.7
	POS	38.1	7.9
10	All	49.0	6.8
	Char	50.4	8.4
	Word	44.0	6.3
	POS	23.3	3.0
20	All	36.5	2.2
	Char	36.9	2.6
	Word	34.0	2.6
	POS	14.9	0.3

(b) ANOVA results and *p*-values

	ANOVA results	
# of Authors	$F(3, 16)$	p-value
2	33.49	$p < 0.001$
5	6.64	$p = 0.004$
10	19.30	$p < 0.001$
20	149.58	$p < 0.001$

Table 14. Post Hoc Tukey Test (with $\alpha = 0.05$) to Evaluate the Significance of the Difference in Using Character Level, Word Level, POS Level, or All Modalities Combined

(a) 2 Authors

Modality	**vs.**	***p*-value**
Character		
	Word	$p = 0.90$
	POS	$p = 0.001$
	All three levels	$p = 0.89$
Word		
	POS	$p = 0.001$
	All three levels	$p = 0.71$
POS		
	All three levels	$p = 0.001$

(b) 5 Authors

Modality	**vs.**	***p*-value**
Character		
	Word	$p = 0.67$
	POS	$p = 0.005$
	All three levels	$p = 0.89$
Word		
	POS	$p = 0.048$
	All three levels	$p = 0.83$
POS		
	All three levels	$p = 0.009$

(c) 10 Authors

Modality	**vs.**	***p*-value**
Character		
	Word	$p = 0.41$
	POS	$p = 0.001$
	All three levels	$p = 0.89$
Word		
	POS	$p = 0.001$
	All three levels	$p = 0.6$
POS		
	All three levels	$p = 0.001$

(d) 20 Authors

Modality	**vs.**	***p*-value**
Character		
	Word	$p = 0.12$
	POS	$p = 0.001$
	All three levels	$p = 0.89$
Word		
	POS	$p = 0.001$
	All three levels	$p = 0.2$
POS		
	All three levels	$p = 0.001$

Table 15. Mean and SD for Using Diacrticis with Different *n*-Gram Modalities

(a) Diacritics are **kept**: Mean and SD

# Authors	Modality	Mean	SD
2	Char	81.8	4.0
	Word	81.1	2.0
	POS	66.7	2.9
	All	83.0	2.6
5	Char	63.0	10.9
	Word	56.0	8.7
	POS	38.1	7.9
	All	61.2	11.5
10	Char	50.4	8.4
	Word	44.0	6.3
	POS	23.3	3.0
	All	49.0	6.8
20	Char	36.9	2.6
	Word	34.0	1.7
	POS	14.9	0.3
	All	36.5	2.2

(b) Diacritics are **removed**: Mean and SD

# Authors	Modality	Mean	SD
2	Char	81.6	4.6
	Word	81.1	2.3
	POS	66.7	2.9
	All	82.3	3.2
5	Char	62.7	10.6
	Word	56.0	8.9
	POS	38.1	7.9
	All	61.0	11.4
10	Char	50.1	8.2
	Word	43.9	6.1
	POS	23.3	3.0
	All	47.4	9.1
20	Char	36.3	2.5
	Word	33.8	1.7
	POS	14.9	0.3
	All	34.2	3.4

Table 16. ANOVA *t*-Test with $\alpha = 0.05$ to Evaluate the Effect of Using Diacrticis with Different *n*-Gram Modalities

(a) ANOVA results and *p*-values

# Authors	ANOVA results $F(5, 24)$
2	$0.31, \quad p = 0.89$
5	$0.47, \quad p = 0.79$
10	$0.75, \quad p = 0.59$
20	$1.75, \quad p = 0.16$

Table 17. Mean and SD for Using Different Categories of Features

Authors	Features	Mean	SD	Authors	Features	Mean	SD
2	Lexical	83.02	7.32	10	Lexical	41.49	10.07
	Structural	74.03	3.09		Structural	24.56	2.1
	Syntactic	81.07	4.06		Syntactic	30.86	10.3
	Lex + Struc	84.87	6.27		Lex + Struc	44.39	8.64
	Lex + Syn	85.3	6.23		Lex + Syn	43.79	10.87
	Struc + Syn	85.66	4.4		Struc + Syn	38.06	7.86
	All three	85.68	4.91		All three	47	8.91
5	Lexical	50.23	6.52	20	Lexical	27.81	8.5
	Structural	42.48	1.6		Structural	17.3	1.23
	Syntactic	43.19	8.76		Syntactic	21.99	7.26
	Lex + Struc	52.61	4.51		Lex + Struc	30.74	7.07
	Lex + Syn	52.55	8.02		Lex + Syn	31.71	10.45
	Struc + Syn	50.68	4.41		Struc + Syn	28.96	7.56
	All three	55.82	6.53		All three	34.55	8.79

Table 18. 5 Authors: Post Hoc Tukey Test to Evaluate Different Categories of Features

(a) ANOVA results and p-values

Pair	p-value	Pair	p-value
Lexical vs Structural	0.572	Syntactic vs Lex + Struc	0.361
Lexical vs Syntactic	0.659	Syntactic vs Lex + Syn	0.369
Lexical vs Lex + Struc	0.90	Syntactic vs Struc + Syn	0.603
Lexical vs Lex + Syn	0.90	Syntactic vs All three	0.104
Lexical vs Struc + Syn	0.90	Lex + Struc vs Lex + Syn	0.90
Lexical vs All three	0.837	Lex + Struc vs Struc + Syn	0.90
Structural vs Syntactic	0.90	Lex + Struc vs All three	0.90
Structural vs Lex + Struc	0.283	Lex + Syn vs Struc + Syn	0.90
Structural vs Lex + Syn	0.29	Lex + Syn vs All three	0.90
Structural vs Struc + Syn	0.515	Struc + Syn vs All three	0.894
Structural vs All three	0.076		

Table 19. 10 Authors: Post Hoc Tukey Test to Evaluate Different Categories of Features

(a) ANOVA results and p-values

# Authors	ANOVA results
2	$F(5, 24) = 0.31, \quad p = 0.89$
5	$F(5, 24) = 0.47, \quad p = 0.79$
10	$F(5, 24) = 0.75, \quad p = 0.59$
20	$F(5, 24) = 1.75, \quad p = 0.16$

REFERENCES

[1] Ahmed Abbasi and Hsinchun Chen. 2005a. Applying authorship analysis to arabic web content. In *Proceedings of the 2005 IEEE International Conference on Intelligence and Security Informatics (ISI'05)*. Springer-Verlag, Berlin, Heidelberg, 183–197. DOI: http://dx.doi.org/10.1007/11427995_15

[2] Ahmed Abbasi and Hsinchun Chen. 2005b. Applying authorship analysis to extremist-group web forum messages. *IEEE Intell. Syst.* 20, 5 (2005), 67–75.

[3] Ahmed Abbasi and Hsinchun Chen. 2006. Visualizing authorship for identification. In *Proceedings of the International Conference on Intelligence and Security Informatics*. Springer, 60–71.

[4] Mahmoud Al-Ayyoub, Ahmed Alwajeeh, and Ismail Hmeidi. 2017. An extensive study of authorship authentication of arabic articles. *Int. J. Web Inf. Syst.* 13, 1 (2017), 85–104. DOI: http://dx.doi.org/10.1108/IJWIS-03-2016-0011

[5] Mahmoud Al-Ayyoub, Yaser Jararweh, Abdullateef Rabab'ah, and Monther Aldwairi. 2017. Feature extraction and selection for arabic tweets authorship authentication. *J. Ambient Intell. Hum. Comput.* 8, 3 (01 Jun 2017), 383–393. https://doi.org/10.1007/s12652-017-0452-1

[6] Alaa Saleh Altheneyan and Mohamed El Bachir Menai. 2014. Naïve bayes classifiers for authorship attribution of arabic texts. *J. King Saud Univ. Comput. Inf. Sci.* 26, 4 (2014), 473–484.

[7] Ahmed Alwajeeh, Mahmoud Al-Ayyoub, and Ismail Hmeidi. 2014. On authorship authentication of arabic articles. In *Proceedings of the 5th International Conference on Information and Communication Systems (ICICS'14)*. IEEE, 1–6.

[8] ArabiNames.com. 2015. Arabi Names. Retreived from http://arabinames.com/categories.aspx.

[9] Victor Benjamin, Wingyan Chung, Ahmed Abbasi, Joshua Chuang, Catherine A. Larson, and Hsinchun Chen. 2014. Evaluating text visualization for authorship analysis. *Secur. Inf.* 3, 1 (2014), 10.

[10] Mudit Bhargava, Pulkit Mehndiratta, and Krishna Asawa. 2013. Stylometric analysis for authorship attribution on twitter. In *Big Data Analytics*. Springer, 37–47.

[11] Leo Breiman. 2001. Random forests. *Mach. Learn.* 45, 1 (2001), 5–32.

[12] Thiago Cavalcante, Anderson Rocha, and Ariadne Carvalho. 2014. Large-scale micro-blog authorship attribution: Beyond simple feature engineering. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer, 399–407.

[13] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2, 3 (2011), 27.

[14] Carole E. Chaski. 2005. Who's at the keyboard? Authorship attribution in digital evidence investigations. *J. Dig. Evidence* 4, 1 (2005), 1–13.

[15] Na Cheng, Rajarathnam Chandramouli, and K. P. Subbalakshmi. 2011. Author gender identification from text. *Dig. Invest.* 8, 1 (2011), 78–88.

[16] Rosa María Coyotl-Morales, Luis Villaseñor-Pineda, Manuel Montes-y Gómez, and Paolo Rosso. 2006. Authorship attribution using word sequences. In *Progress in Pattern Recognition, Image Analysis and Applications*. Springer, 844–853.

[17] Olivier de Vel, Alison Anderson, Malcolm Corney, and George Mohay. 2001. Mining E-mail content for author identification forensics. *ACM SIGMOD Reco.* 30, 4 (2001), 55–64.

[18] Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2007. Automated methods for processing arabic text: From tokenization to base phrase chunking. *Arabic Computational Morphology: Knowledge-Based and Empirical Methods*. (2007).

[19] Steven H. H. Ding, Benjamin C. M. Fung, and Mourad Debbabi. 2015. A visualizable evidence-driven approach for authorship attribution. *ACM Trans. Inf. Syst. Secur.* 17, 3, Article 12 (March 2015), 30 pages. DOI: http://dx.doi.org/10.1145/2699910

[20] Georgia Frantzeskou, Efstathios Stamatatos, Stefanos Gritzalis, Carole E. Chaski, and Blake Stephen Howald. 2007. Identifying authorship by byte-level N-grams: The source code author profile (SCAP) method. *Int. J. Dig. Evidence* 6, 1 (2007), 1–18.

[21] Zhenhao Ge, Yufang Sun, and Mark J. T. Smith. 2016. Authorship attribution using a neural network language model.. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 4212–4213.

[22] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep Learning*. Vol. 1. MIT Press Cambridge.

[23] Nizar Habash and Owen Rambow. 2005. Arabic tokenization, morphological analysis, and part-of-speech tagging in one fell swoop. In *Proceedings of the Conference of American Association for Computational Linguistics*. 578–580.

[24] M. A. Hall. 1998. Correlation-based feature subset selection for machine learning. (unpublished).

[25] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *ACM SIGKDD Explor. Newslett.* 11, 1 (2009), 10–18.

[26] Jiawei Han and Micheline Kamber. 2001. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco, CA.

[27] Markus Hofmann and Ralf Klinkenberg. 2013. *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC.

[28] Giacomo Inches, Morgan Harvey, and Fabio Crestani. 2013. Finding participants in a chat: Authorship attribution for conversational documents. In *Proceedings of the International Conference on Social Computing (SocialCom'13)*. IEEE, 272–279.

[29] Farkhund Iqbal, Rachid Hadjidj, Benjamin C. M. Fung, and Mourad Debbabi. 2008. A novel approach of mining write-prints for authorship attribution in e-mail forensics. *Dig. Invest.* 5 (Suppl.) (2008), S42–S51.

[30] Shunichi Ishihara. 2011. A forensic authorship classification in sms messages: A likelihood ratio based approach using N-Gram. In *Proceedings of the Australasian Language Technology Association Workshop 2011*. 47–56.

[31] George H. John and Pat Langley. 1995. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, San Francsisco, CA, 338–345.

[32] Patrick Juola. 2006. Authorship attribution. *Found. Trends Inf. Retr.* 1, 3 (Dec. 2006), 233–334. DOI: http://dx.doi.org/10.1561/1500000005

[33] Vlado Kešelj, Fuchun Peng, Nick Cercone, and Calvin Thomas. 2003. N-gram-based author profiles for authorship attribution. In *Proceedings of the Conference Pacific Association for Computational Linguistics (PACLING'03)*, Vol. 3. 255–264.

[34] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). Association for Computational Linguistics, Doha, Qatar, 1746–1751.

[35] Bradley Kjell, W. Addison Woods, and Ophir Frieder. 1994. Discrimination of authorship using visualization. *Inf. Process. Manage.* 30, 1 (1994), 141–150.

[36] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *J. Am. Soc. Inf. Sci.Technol.* 60, 1 (2009), 9–26.

[37] Sushil Kumar and Mousmi A. Chaurasia. 2012. Assessment on stylometry for multilingual manuscript. *Assessment* 2, 9 (2012), 1–6.

[38] Robert Layton, Stephen McCombie, and Paul Watters. 2012. Authorship attribution of irc messages using inverse author frequency. In *Proceedings of the 3rd Cybercrime and Trustworthy Computing Workshop (CTC'12)*. IEEE, 7–13.

[39] Robert Layton, Paul Watters, and Richard Dazeley. 2010. Authorship attribution for twitter in 140 characters or less. In *Proceedings of the Second Cybercrime and Trustworthy Computing Workshop*. IEEE, 1–8.

[40] Robert Layton, Paul Watters, and Richard Dazeley. 2012. Recentred local profiles for authorship attribution. *Nat. Lang. Eng.* 18, 3 (7 2012), 293–312. DOI: http://dx.doi.org/10.1017/S1351324911000180

[41] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. 681–691.

[42] Mark Liberman. 2008. Ask Language Log: Comparing the Vocabularies of Different Languages. Retrieved from http://itre.cis.upenn.edu/ myl/languagelog/archives/005514.html.

[43] Kim Luyckx and Walter Daelemans. 2008. Authorship attribution and verification with many authors and limited data. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 513–520.

[44] Kim Luyckx and Walter Daelemans. 2011. The effect of author set size and data size in authorship attribution. *Liter. Ling. Comput.* 26, 1 (2011), 35–55.

[45] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford coreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics (ACL), 55–60. DOI: http://dx.doi.org/10.3115/v1/P14-5010

[46] Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *Proceedings of the AAAI Workshop on Learning for Text Categorization*, Vol. 752. Citeseer, 41–48.

[47] Miniwatts Marketing Group. 2013. Internet World Users by Language. Retreived from http://www.internetworldstats.com/stats7.htm.

[48] Frederick Mosteller and David Wallace. 1964. Inference and Disputed Authorship: The Federalist. Addison-Wesley.

[49] Ahmed Fawzi Otoom, Emad E. Abdullah, Shifaa Jaafer, Aseel Hamdallh, and Dana Amer. 2014. Towards author identification of arabic text articles. In *Proceedings of the 5th Internation Conference on Information and Communication Systems (ICICS'14)*. IEEE, 1–4.

[50] Siham Ouamour and Halim Sayoud. 2013. Authorship attribution of short historical arabic texts based on lexical features. In *Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC'13)*. IEEE, 144–147.

[51] Arfath Pasha, Mohamed Al-Badrashiny, Mona T. Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'14)*, Vol. 14. 1094–1101.

[52] John Ross Quinlan. 1993. C4.5: Programs for machine learning. Vol. 1. The Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann, San Mateo, CA.
[53] Abdullateef Rabab'ah, Mahmoud Al-Ayyoub, Yaser Jararweh, and Monther Aldwairi. 2016. Authorship attribution of arabic tweets. In *Proceedings of the IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA'16)*. 1–6.
[54] Roshan Ragel, Pramod Herath, and Upul Senanayake. 2013. Authorship detection of SMS messages using unigrams. In *Proceedings of the 8th International Conference on Industrial and Information Systems (ICIIS'13)*. IEEE, 387–392.
[55] Dylan Rhodes. 2015. Author attribution with CNNs. Retrieved August 22, 2016 from https://www.semanticscholar.org/paper/Author-Attribution-with-Cnn-s-Rhodes/0a904f9d6b47dfc574f681f4d3b41bd840871b6f/pdf.
[56] David C. Rubin. 1978. Word-initial and word-final Ngram frequencies. *J. Literacy Res.* 10, 2 (1978), 171–183.
[57] Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *arXiv Preprint arXiv:1609.06686* (2016).
[58] Roy Schwartz, Oren Tsur, Ari Rappoport, and Moshe Koppel. 2013. Authorship attribution of micro-messages. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP'13)*. 1880–1891.
[59] Kareem Shaker and David Corne. 2010. Authorship attribution in arabic using a hybrid of evolutionary search and linear discriminant analysis. In *Proceedings of The Computational Intelligence (UKCI'10) Workshop*. IEEE, UK, 1–6. DOI: http://dx.doi.org/10.1109/UKCI.2010.5625580
[60] Kareem Shaker, David Corne, and Richard Everson. 2007. Investigating hybrids of evolutionary search and linear discriminant analysis for authorship attribution. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2071–2077.
[61] Armin Shmilovici. 2005. *Support Vector Machines*. Vol. 12. Springer New York, NY, 257–276. DOI: http://dx.doi.org/10.1007/0-387-25465-X_12
[62] Prasha Shrestha, Sebastian Sierra, Fabio A. González, Paolo Rosso, Manuel Montes-y Gómez, and Thamar Solorio. 2017. Convolutional neural networks for authorship attribution of short texts. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL'17)*. 669.
[63] Rui Sousa Silva, Gustavo Laboreiro, Luís Sarmento, Tim Grant, Eugénio Oliveira, and Belinda Maia. 2011. 'twazn me!!!;('Automatic authorship analysis of micro-blogging messages. In *Natural Language Processing and Information Systems*. Springer, 161–168.
[64] Steve Simon. 2005. When the F Test Is Significant, but Tukey Is Not. Retrieved from http://www.pmean.com/05/TukeyTest.html.
[65] Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci. Technol.* 60, 3 (2009), 538–556.
[66] Nick Taylor. 2015. Twitter and Open Data in Academia. Retrieved from https://twittercommunity.com/t/twitter-and-open-data-in-academia/51934.
[67] Twitter. 2017. Developer Agreement and Policy. Retrieved from https://dev.twitter.com/overview/terms/agreement-and-policy
[68] Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). Morgan Kaufmann.
[69] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*. 649–657.
[70] Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. 2006. A framework for authorship identification of online messages: Writing-style features and classification techniques. *J. Am. Soc. Inf. Sci. Technol.* 57, 3 (2006), 378–393.

Received October 2015; revised March 2018; accepted June 2018