Evaluating Author Attribution on Emirati Tweets

MAHMOUD KHONJI¹, (Member, IEEE), AND YOUSSEF IRAQI², (Senior Member, IEEE)

¹ECCS Department, Khalifa University, Abu Dhabi, United Arab Emirates
²Center of Cyber-Physical Systems, College of Engineering, Khalifa University, Abu Dhabi, United Arab Emirates

Corresponding author: Youssef Iraqi (youssef.iraqi@ku.ac.ae)

This work was supported by the Buhooth scholarship, UAE.

ABSTRACT Author Attribution (AA) is a critical stylometry problem that tries to deduce the identity of the authors of electronic texts (e-texts) by only examining the texts. AA is essential for enhancing various application domains, such as recommender systems and forensics. Nevertheless, existing techniques in AA have not been assessed with Emirati social media e-texts. The reason is that no suitable dataset exists for evaluating AA techniques in this context. This paper introduces the Khonji-Iraqi Emirati Tweets Author Identification (AID) dataset with 30 authors (KIT-30), and detailed evaluations. Compound grams, a new definition of grams, are introduced, which allows us to achieve higher classification accuracy. Also, when the number of suspect authors increases, the classification accuracy degradation is not as severe as previously reported, when using suitable data representation. Furthermore, in order to work towards addressing the lack of conveniently-available implementations of stylometry methods, we have developed an extensive e-text feature extraction library, namely Fextractor, with a highly intuitive API. The library generalizes all existing \( n \)-gram-based feature extraction methods under the \( \text{at least } l \)-frequent, \( \text{dir-directed}, \ k\text{-skipped } n\text{-grams} \), and allows grams to be diversely defined, including definitions that are based on high-level grammatical aspects, such as Part of Speech (POS) tags, as well as lower-level ones, such as the distribution of function words and word shapes.

INDEX TERMS Stylometry, author identification, text analysis, supervised learning, unsupervised learning, classification.

I. INTRODUCTION

E-text stylometry is concerned about analyzing the writing styles of input e-texts in order to extract information about their authors. Such inferred information could be the identity of their authors, their genders, age groups, personality types, or even the diagnosis of certain illnesses [1], [2]. Author Attribution (AA) is an important problem in e-text stylometry and is defined as follows: given a set of texts with known authors, find a classification model that predicts which of these known authors is also the author of the input test texts whose authors are not known. The target classification label, in this case, is the identity of the author [3]–[5]. This is a closed-set classification task, which means that the classification model is expecting the actual author of the input test text to be represented in the learning set.

While various stylometry problem solvers have been evaluated against texts of various domains, the accuracy of AA techniques on Emirati social media texts is unknown. This work aims to address the following two challenges that face e-text stylometry problems, namely:

- The lack of evaluation datasets for stylometry problem solvers, when executed against e-texts that are written in Emirati Arabic, a dialect of the Arabic language that is natively spoken in the United Arab Emirates (UAE). This effectively casts uncertainty concerning the performance of all stylometry methods, when evaluated against electronic texts that are written in this dialect. As a result, the applicability of e-text stylometry methods against Emirati texts to enhance forensics, anti-forensics, or market analysis, is unknown.

- The lack of conveniently-available, and extensive, software that implement the many existing stylometry methods and feature extraction functions. There is often a tremendous need in re-developing the many proposed methods or functions, and because of the sheer amount of effort that is required to develop as such, it is common that most of the methods or functions are not adequately evaluated. As a result, the actual value of the numerous

The associate editor coordinating the review of this manuscript and approving it for publication was Victor S. Sheng.
independent contributions, relative to each other, is often not adequately known.

Hence, this work has these main contributions:

- The construction of an original AA assessment dataset made of Emirati tweets (the KIT-30 dataset).
- The new category of grams, namely compound grams, that allows a significant classification accuracy increase.
- The extensive assessment of AA classification techniques against the introduced dataset.
- The implementation of an extensive stylometry feature extraction library with easy-to-use interface in Python. While alternative feature extraction libraries exist [6], to the best of our knowledge, our library Fextractor is, by far, the most extensive library of its kind to date. Our library supports language-independent features, as well as language-dependent features for the following languages: Arabic, English, Chinese, French, German, and Spanish.
- The generalization of numerous feature extraction methods. This allows us to define novel variants of the existing feature extraction methods, in addition to simplifying the implementation.
- The release of the library under a permissible open-source library. We hope that this would enable other researchers to conveniently study the feature extraction methods, or evaluate their methods against the existing ones, without facing the time and effort barrier that is currently required to implement the many methods.

The results of the performance evaluation of more than 10,000 AA models show that the techniques using the introduced compound grams have significantly higher accuracy than those using other types of grams. The results also show that even when the number of suspect authors increases to 30, some AA models can achieve high accuracy in the context of Emirati tweets when using suitable text vectorization methods.

These results are remarkable as they also imply that the accuracy decrease when adding more authors, is not as severe as proclaimed before [7]. More specifically, while the most accurate Twitter AA models are less accurate than 0.8 even with two authors [7], our top performing technique achieves 0.98 accuracy with 30 authors.

The remaining of this paper is as follows. Section II discusses related works. Section III introduces the KIT-30 dataset. Section IV presents the technique used for solving the AA problems, while compound grams are introduced in Section V. Sections VI and VII show the evaluation approach and the results, respectively. Section VIII introduces Fextractor, our extensive feature extraction library, and the conclusion is given in Section IX.

II. RELATED WORKS

The most relevant investigation to the present work is the PAN’12 closed-set AA challenge [8], where a number of AA models are evaluated against several problems, including closed-set AA ones. Although the considered datasets were only in the English language, this investigation is important as it shows the performance of leading AA algorithms.

Khonji et al. have shown in [9] that the Random Forests (RFs) classification algorithm can achieve an accuracy that is equivalent to that of the best AA models of the closed-set AA evaluation of PAN’12.

Other AIDs evaluation problems in the literature, including the following editions of PAN competitions expanded the set of considered languages within their datasets. E.g., the following languages were added in recent PAN evaluations: Dutch, Greek, and Spanish. However, evaluation of stylometry methods, such as AA solvers, against Emirati texts remained absent in the literature.\(^1\)

The most related evaluation dataset to our constructed KIT-30 is perhaps the Arabic Sentiment Tweets Dataset (ASTD) by Nabil et al. [11]. Still, while our techniques of collecting the tweets are influenced by their work, the ASTD dataset has the authors' identifiers removed (rightfully so due to the nature of Twitter terms of use, and the nature of the study targeted by the ASTD dataset). Therefore, ASTD is not suitable to evaluate AA techniques.

III. THE KHONJI-IRAQI EMIRATI TWEETS AID EVALUATION DATASET (KIT-30)

This section introduces the objective of our dataset, methods that were used in order to construct it, as well as its various statistics.

The goal of the KIT-30 dataset is to provide e-text fit for generating and answering AID questions to evaluate AID models. AID can be the AA problem addressed in this paper, or the Author Verification (AV) problem (verifying if the same author wrote a pair of texts), or the Author Diarization (AD) problem (grouping sections in a particular document according to their authors).

To accomplish these goals, we execute the tasks in Algorithm 1. This resulted in obtaining a total number of 30 Emirati Twitter accounts [10]. The first two steps of the algorithm were inspired by Nabil et al. [11].

Algorithm 1 Obtaining a Set of Twitter User Accounts

1) Detect the most active accounts in the UAE by using SocialBakers.
2) Detect more accounts by looking for specific tags that are unique to the UAE.
3) Manually examine the saved accounts to drop non-UAE or non-Arabic accounts.

Next, Algorithm 2 was used to save, discard, and preprocess the tweets as deemed appropriate for the objectives of the evaluation dataset at hand. This resulted in obtaining the finalized KIT-30 dataset, which is comprised of over 50,000 tweets in total.

\(^1\) A preliminary version of part of this work appeared in [10]
To allow meaningful comparisons among the evaluation results against this Emirati tweets dataset and those of other languages, we have repeated the same process by adapting it for the Dutch, Greek, Spanish, and English languages. Table 1 shows the statistics of KIT-30, while Table 2 presents the per-author statistics of Emirati tweets subset.

### Table 1. Overall KIT-30 statistics.

| Dataset          | #Authors | #Tweets |
|------------------|----------|---------|
| Emirati tweets   | 30       | 51957   |
| Dutch tweets     | 30       | 62018   |
| Greek tweets     | 30       | 63632   |
| Spanish tweets   | 30       | 59996   |
| US tweets        | 30       | 70837   |

The implementation of the Tweets downloader in Step 1 of Algorithm 2, along with the finalized KIT-30 dataset are released under permissible open-source licenses.

### IV. AUTHOR ATTRIBUTION MODEL

In the context of this work, we adopt RFs as the learning technique as it was shown in [9] that this algorithm achieves competitive high classification accuracy when solving AA problems.

RFs need the input samples to be described as vectors. We follow a vector representation approach similar to that of Khonji et al. [9]. Every text $x$ is denoted by a vector $x$, where $x[i]$ denotes the frequency of a unique $k$-skip $n$-gram [12] in the text $x$. The element $i$ always refers to the frequency of the same unique $k$-skip $n$-gram pattern. For example, given a pair of vectors $x_1$ and $x_2$ that represent different texts $x_1$ and $x_2$, respectively, $x_1[i]$ is the frequency of a unique pattern in text $x_1$, while $x_2[i]$ is the frequency of the same pattern in text $x_2$. This will allow a meaningful comparison of the pair of vectors.

Then the learning set of texts represented as vectors and their author identities are used to train an RF model. The trained model is subsequently used for predicting the author of texts in the testing set.

Before we define $k$-skip $n$-gram patterns, we define $n$-gram patterns, and then expand the definition of $n$-grams by the addition of $k$-skips.

An $n$-gram pattern is a series of $n$ neighboring grams in a given text. A gram is a parameter that defines the most fundamental unit of the processed text. For example, if grams are words, then the most basic unit of any text is considered to be words. Figure 1 depicts all $n$-grams for the text “The quick fox jumped over the lazy dog” when grams are words, and $n = 3$.

![Figure 1. List of n-grams when grams are words and $n = 3$.](image-url)
n-grams such that the grams adjacency constraint can be violated for up to $k$ many skips [12].

For example, if $k = 2$, and the starting gram is “The”, then we not only identify the 3-gram “The quick fox”, but also all of its 3-gram variants as listed in Table 3.

**Table 3.** $k$-skip n-grams in text “The quick fox jumped over . . . ” for when $k = 2, n = 3$, grams are words, and the first gram is “The”.

| $k = 2, n = 3$, and grams=words | #skips |
|---------------------------------|--------|
| The quick fox                   | 0      |
| The fox jumped                  | 1      |
| The quick jumped                | 1      |
| The jumped over                 | 2      |
| The quick                      | over   | 2      |
| The fox                        | over   | 2      |

Similar inflation affects all other n-grams, except those near the end of the string by which only fewer skips become possible in order to avoid overrunning after the end of the string.

It can be seen that the concept of $k$-skip n-grams is a generalization of the concept of n-grams. This makes n-grams a special case of $k$-skip n-grams for when $k = 0$. I.e., 0-skip n-grams and n-grams are identical (for any value of n, and any definition of what constitutes as a gram).

Note that since $k$-skip n-grams inflate each n-gram into multiple variants with skips ranging from 0 up to $k$ (inclusive of 0 and $k$), the total number of n-gram parameters increases combinatorially. Therefore large values of $k$ are sometimes computationally infeasible.

Additionally, it is customary to disregard less-frequent $k$-skip n-grams. This is to reduce dimensionality, as well as due to the fact that such measures are often found to be too noisy for the purpose of solving AA problems. A successful rule that has been used in the literature is to drop all $k$-skip n-grams that only occur for less than $n$ times in any single text in the dataset. In our preliminary evaluations of the proposed models, we found that $l = 5$ was optimal for the evaluation. I.e., if a pattern fails to appear five or more times in any text, it is ignored and therefore not used in subsequent analysis.

**V. COMPOUND GRAMS**

Table 3 presents 2-skip 3-grams, when grams are defined to be words. Another definition of grams that is known in the literature of stylometry is defining them to be the POS tags, or dependency tags. For example, Table 4 presents an example of the case when grams are defined to be POS tags. Note that each word is substituted by its corresponding POS tag, as defined by the Penn Treebank project. The same could be trivially extended to dependency tags, word lengths, and word shapes.

Compound grams essentially aim to aggregate multiple definitions of grams that refer to the same text segment.

Table 4 presents $k$-skip n-grams in text “The quick fox jumped over . . . ” for when $k = 2, n = 3$, grams are POS tags, and the first gram is the POS tag of “The”.

| $k = 2, n = 3$, and grams=POS tags | #skips |
|-----------------------------------|--------|
| DT JJ NN                          | 0      |
| DT NN VBD                        | 1      |
| DT JJ VBD                        | 1      |
| DT VBD IN                       | 2      |
| DT JJ IN                        | 2      |
| DT NN IN                        | 2      |

Table 5 presents examples of some compound grams, when aggregating the definition “word” and “POS tag” into one gram.

Compound grams allow for capturing additional information than the classical ones. For example, measuring the frequencies of grams, as shown in Tables 3 and 4, allows for identifying the tendency of words or POS tags to independently occur in a given text. On the other hand, as shown in Table 5, measuring the frequency of compound grams allows for identifying the tendency of certain words to jointly take certain POS tags in a given text. This can be valuable information for identifying authors, as authors can be made unique not only by the independent frequency of certain grams (words or POS tags), but rather by their tendency of choosing certain words in certain positions of their sentences. For example, the word “saw” can be used as both, a verb, and a noun as in the sentence “I saw the saw”.

**VI. EVALUATION METHODOLOGY**

Once AA models are trained as described in Section IV, we evaluate them by using 10-fold cross-validation. However, in order to ensure that each fold is comprised of realistic learning and testing samples, we add the constraint that limits the free mixing of tweets that were written at different times. Specifically, the tweets per author are chronologically grouped into 10 chunks such that their tweets do not exist in two adjacent time intervals.

This constraint increases the difficulty of the AA problems, as it substantially minimizes the possibility of test tweets being chronologically too close from their learning counterparts.

The statistics of the evaluation dataset, after grouping the tweets into 10 chronological chunks on per author basis, are presented in Table 6.
Recall from earlier sections that the at least $l$-frequent $k$-skip $n$-grams have the following parameters:

- Parameter $l$.
- Parameter $k$.
- Parameter $n$.
- The definition of what constitutes a gram.

For completeness, we repeat the evaluation many times, each with a distinct AA RF model, such that each makes use of a unique data representation function. Specifically, we exhaustively implement all possible definitions of the at least $l$-frequent $k$-skip $n$-grams for the following sets of parameter values:

- $1 \leq l \leq 9$.
- $0 \leq k \leq 1$.
- $1 \leq n \leq 3$.
- Gram $\in \{\text{word, word length, POS tag, word-POS tag tuple, dependency tag, word-dependency tag tuple, POS-dependency tags tuple}\}$. The tuple grams essentially represent a special case of our proposed compound grams with two components.

This process results in $9 \times 2 \times 3 \times 7 = 378$ unique text vectorization methods, each of which is used by an RF model that is evaluated by 10-fold cross-validation.

The only exception to this is the Dutch and Greek datasets by which gram can be word or word length. This is because the POS tagger that we use does not support these languages.

Additionally, since the accuracy of AA problem solvers is sensitive to the number of considered authors, we repeat the entire evaluation for 29 times, each time while evaluating against a unique size of suspects space. I.e., we evaluate for all suspect space sizes in $\{2, 3, \ldots, 30\}$. Therefore, the total number of evaluations is $378 \times 29 = 10,962$ many 10-fold cross-validations.

Subsequently, to investigate the statistical significance of the different performance results, Approximate Randomization (AR) [13] is applied to compute the $p$ values. The labels of various significance levels are shown in Table 7. For example, if $0.01 < p \leq 0.05$, we view the variation between considered classification accuracy as statistically significant, as is the case in [14], and indicate it by one asterisk ‘‘*’’.

| Symbol | Level | Name |
|--------|-------|------|
| $=$    | $p \geq 0.05$ | Significance not shown |
| *      | $0.01 < p \leq 0.05$ | Significant |
| **     | $0.01 > p > 0.001$ | Very significant |
| ***    | $p < 0.001$ | Highly significant |

### VII. EMIRATI TWEETS EVALUATION RESULTS

#### A. ACCURACY OF AUTHOR ATTRIBUTION MODELS AS A FUNCTION OF SUSPECTS SPACE SIZE

Recall from earlier that, in total, $10,962$ 10-fold cross-validations are performed in order to evaluate RF AA models exhaustively with various parameter values of $l$, $k$, and $n$.

Figure 2 depicts the empirical cumulative density function (ECDF) of all of the $10,962$ classification accuracies found by 10-fold cross-validation using the Emirati tweets in KIT-30, such that, for any line $i \in \{2, 3, \ldots, 30\}$ (each line $i$ is denoted by a unique color), $i$ represents the ECDF of the classification accuracy of all models that are assessed against problems with a suspects space of $i$ many authors.

![Figure 2](image-url)

The results in Figure 2 show that the larger the number of suspect authors, the more there are text representation techniques that achieve lower RFs AA accuracy. Nonetheless,
even with the 30 authors, there are specific text representation techniques that allow the RFs AA models to achieve an accuracy very close to 1. More details on such successful configurations will be outlined in the next subsection of this evaluation.

Figure 3 presents the classification accuracy versus the number authors. This accuracy is measured by considering the performance of all of the feature extraction functions. It can be seen that the performance of solving AA problems against Emirati tweets is superior to those of Dutch and Greek datasets, and inferior to those of Spanish and US English.

However, this is not necessarily an indication that solving AA problems is more difficult under Emirati tweets than Spanish or US English. This is due to the fact that some poorly performing features could degrade the overall classification accuracy, and mask the effect of the well-performing features.

To demonstrate this, Figures 4, 5, 6 and 7 present the same results like those in Figure 3, except for choosing specific feature extraction functions that tend to perform well under specific datasets.

It can be seen that the performance of solving AA problems with Emirati tweets can be highly similar to those of Spanish and US tweets datasets when certain feature extraction methods are chosen. Namely, when defining grams as the tuple of word-POS tags. However, the performance on Emirati tweets degrades significantly when grams are words, as shown in Figures 5 and 6.

This suggests that, while the current methods of stylometry analysis were never previously assessed with Emirati social media texts (and rarely against Arabic texts in general), accurately solving AA problems with Emirati tweets is nonetheless possible by using compound grams that are formed by...
TABLE 8. The accuracy with 30 authors for Emirati tweets of the best model for the given value of $I$.

| $I$ | acc = 0.975 | acc = 0.983 | acc = 0.980 | acc = 0.973 | acc = 0.983 | acc = 0.973 | acc = 0.970 | acc = 0.973 | acc = 0.973 |
|-----|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 1   | $p = 1.0000$| $p = 0.4346$| $p = 0.7932$| $p = 0.8002$| $p = 0.4585$| $p = 0.4596$| $p = 0.8042$| $p = 0.7752$|
| 2   | acc = 0.977 | =           | =           | =           | =           | =           | =           | =           | =           |
| 3   | acc = 0.983 | =           | $p = 0.7682$| $p = 0.7972$| $p = 0.4665$| $p = 0.4306$| $p = 0.7872$| $p = 0.8202$|
| 4   | acc = 0.980 | =           | =           | $p = 0.2717$| $p = 1.0000$| $p = 0.2048$| $p = 0.2837$| $p = 0.3147$|
| 5   | acc = 0.973 | =           | =           | =           | $p = 0.7952$| $p = 0.3257$| $p = 0.4665$| $p = 0.4885$|
| 6   | acc = 0.983 | =           | =           | =           | =           | $p = 0.2897$| $p = 0.6743$| $p = 1.0000$| $p = 1.0000$|
| 7   | acc = 0.970 | =           | =           | =           | =           | =           | $p = 0.2088$| $p = 0.2717$| $p = 0.2887$|
| 8   | acc = 0.973 | =           | =           | =           | =           | =           | =           | $p = 0.6643$| $p = 0.6603$|
| 9   | acc = 0.973 | =           | =           | =           | =           | =           | =           | =           | $p = 1.0000$|

combining successful feature extraction methods as found based on the literature of stylometry for other languages.

B. ACCURACY OF AUTHOR ATTRIBUTION MODELS AS A FUNCTION OF TEXT VECTORIZATION METHODS

Since this section discusses the effect of the various parameters of the feature extraction functions in greater detail, the discussion is focused on Emirati tweets and a suspects space of 30 authors for brevity.

Figure 8 depicts the ECDFs of the evaluated RF AA classification models with varying values of $I$ when tested against a set of 30 suspect authors. The ECDFs generally indicate that the most accurate classification models can be identified when $I \in \{3, 6\}$. Interestingly, this is close to the value $I = 5$ that was found by Khonji et al. [15] for the other languages (i.e., Dutch, English, Greek, and Spanish).

However, the most accurate AA classification models under each value of $I \in \{1, 2, \ldots, 9\}$, are not statistically significantly different than those found with different values of $I$. Table 8 presents the pair-wise statistical significance results against the most accurate models that are found under each value of $I$.

Figure 9 depicts the ECDFs of the evaluated RF AA classification models with varying values of $k$. The ECDFs indicate that when $k = 0$ more accurate classification models can be identified than when $k = 1$, which suggests that the tolerated violations of the grams adjacency assumption are unhealthy for identifying authors of Emirati social media texts. However, Table 9 indicates that the difference between the best performing classifiers under each value of $k$ is not statistically significant.

Figure 10 depicts the ECDFs of the evaluated RF AA classification models with varying values of $n$. The ECDFs indicate that when $n = 1$ more accurate classification models can

Figure 8. ECDF of classification accuracy for all $I \in \{1, 2, \ldots, 9\}$ with a suspects space of 30 authors for Emirati tweets.

Figure 9. ECDF of classification accuracy by $k$ with a suspects space of 30 authors for Emirati tweets.

TABLE 9. The accuracy with 30 authors for Emirati tweets of the best model for the given value of $k$.

| $k$ | acc = 0.967 |
|-----|-------------|
| 0   | $p = 0.1279$|
| 1   | acc = 0.987 |

VOLUME 8, 2020
be identified than when \( n > 1 \), which suggests that observing the distribution of grams in relation to their adjacent ones is unhealthy for identifying authors of Emirati social media texts. However, Table 10 indicates that the difference in accuracy between the most accurate models under each value of \( n \) is not statistically significant. The only exception is between the cases when \( n = 1 \) and \( n = 3 \) by which the difference in accuracy is statistically significant.

Figure 11 depicts the ECDFs of the evaluated RF AA classification models with varying definitions of grams. The ECDFs indicate that compound grams allow for the identification of more accurate classification models than otherwise. Table 11 indicates that the increase in accuracy with the most accurate models using compound grams is always statistically significant, except for the gram \( \text{pos} \), where the difference is not statistically significant (\( p = 0.2348 \)), which may be due to the size of the dataset.

Table 12 presents a classification accuracy ranked list, and parameters of the text vectorization methods, of the best performing classifiers with a small enough difference between classification accuracy not to be statistically significant. It can be seen that the top 10 best performing classifiers exclusively make use of compound grams. This suggests that our novel definition of grams is successful in allowing for the achievement of higher classification accuracy under the Emirati tweets domain than when grams are defined classically.

It is important to note that an accurately performing AA classifier is not necessarily an indication of the model’s ability in identifying the writing styles of authors. For example, if the dataset contains a significant author-topic bias, then a model that is originally intended to be an AID model, can be partly both, an AID model, as well as a topic identification model. Therefore care must be taken to ensure that the used features do not contain too much topic information, as such topic information could confuse the learning algorithm and transform it into a topic classifier up to a larger degree than otherwise. This is specifically a concern when features that contain words are used, as such words could be content words (as opposed to function words).

If compound grams contain excessive topic information, this may lead the model to become a topic classifier instead of an author classifier. Therefore, to ensure that this is not the case for our best performing compound gram (\( \text{word-POS} \) or \( \text{word-dep} \) as shown in Table 12), Figure 12 lists the 20 most important features. The features were aggregated from each of the 10 evaluation folds as used in our RF models (duplicate entries are removed).

In this case, none of the recorded compound grams contains content or topic words. Interestingly, the identified Arabic words in the list above are also Arabic function words. The only arguable word is “الله”, which translates to “God”. However, since the word “الله” is often used in various expressions that are independent of the topic, we believe that it is fair to consider it a word that does not contain significant topic information.
TABLE 12. The top 45 most accurate classifiers with 30 authors for Emirati tweets.

| l  | k | n  | Gram   | Accuracy |
|----|---|----|--------|----------|
| 6  | 0 | 1  | word-pos | 0.9833  |
| 3  | 0 | 1  | word-pos | 0.9833  |
| 4  | 0 | 1  | word-pos | 0.9800  |
| 2  | 0 | 1  | word-deep | 0.9767  |
| 1  | 0 | 1  | word-pos | 0.9767  |
| 9  | 0 | 1  | word-pos | 0.9733  |
| 8  | 0 | 1  | word-deep | 0.9733  |
| 5  | 0 | 2  | pos     | 0.9700  |
| 7  | 0 | 1  | word-pos | 0.9700  |
| 2  | 0 | 1  | word-deep | 0.9700  |
| 4  | 1 | 2  | word-pos | 0.9667  |
| 4  | 0 | 2  | word-pos | 0.9667  |
| 3  | 0 | 2  | word-pos | 0.9667  |
| 7  | 0 | 1  | word-deep | 0.9667  |
| 5  | 0 | 1  | word-deep | 0.9667  |
| 4  | 0 | 1  | word-deep | 0.9667  |
| 8  | 0 | 2  | word-pos | 0.9633  |
| 8  | 1 | 2  | word-pos | 0.9633  |
| 3  | 1 | 2  | word-pos | 0.9633  |
| 6  | 0 | 1  | word-deep | 0.9633  |
| 9  | 0 | 1  | word-deep | 0.9600  |
| 6  | 1 | 2  | word-pos | 0.9567  |
| 6  | 0 | 1  | word-deep | 0.9567  |
| 2  | 0 | 1  | word-deep | 0.9567  |
| 7  | 1 | 2  | word-pos | 0.9533  |
| 7  | 0 | 2  | word-pos | 0.9533  |
| 5  | 0 | 2  | word-pos | 0.9533  |
| 2  | 1 | 2  | pos     | 0.9533  |
| 2  | 1 | 2  | word-pos | 0.9533  |
| 2  | 0 | 2  | pos     | 0.9533  |
| 1  | 0 | 1  | word-deep | 0.9533  |
| 9  | 1 | 2  | word-pos | 0.9500  |
| 9  | 1 | 2  | pos     | 0.9500  |
| 8  | 1 | 2  | word-pos | 0.9500  |
| 3  | 1 | 2  | word-deep | 0.9500  |
| 1  | 0 | 3  | pos     | 0.9500  |
| 8  | 1 | 3  | pos     | 0.9467  |
| 7  | 1 | 2  | pos     | 0.9467  |
| 9  | 0 | 2  | word-pos | 0.9433  |
| 7  | 0 | 2  | word-pos | 0.9433  |
| 4  | 0 | 2  | pos     | 0.9433  |
| 3  | 1 | 2  | pos     | 0.9433  |

FIGURE 12. Most performing features.

On the other hand, the least performing features (i.e., features that contribute least to AA model’s decision in solving AA problems) contain a significant amount of content or topic words. A list of such features is presented in Figure 13.

This supports the claims that the KIT-30 dataset does not include meaningful author-topic bias, and that the suggested compound grams are reasonably assisting the learning algorithm to find AID models, as opposed to topic classification models.

FIGURE 13. Least performing features.

VIII. FEXTRACTOR: EXTENSIVE STYLOMETRY FEATURE EXTRACTION LIBRARY

One of the critical issues that face today’s research on stylometry is the fact that implementations of most of the stylometry-related proposed methods are not released publicly. As a result, re-evaluating, or comparing newer methods against the previous ones is often extremely difficult due to the need for re-implementing those methods again (which requires a tremendous amount of time and effort).

A notable aspect of the research in e-text stylometry, is the enhancement of feature extraction methods. Currently, such methods are highly diverse, and range from simple letter counts, up to more sophisticated ones that use independent statistical models, such as POS taggers. For example, it is quite common in the literature that a good portion of the considered feature extraction methods are evaluated in isolation, without adequate comparison against existing methods to truly justify their relative effectiveness. Another issue is the lack of adequate generalizations of the proposed methods, which leaves some of the novel variants unstudied.

A. SUPPORTED FEATURE EXTRACTION METHODS

The following feature extraction methods are supported:

- n-grams (classical n-grams), with parameters:
  - Normalize (Boolean): If set to True, the library will normalize the number of occurrences of a
TABLE 13. The currently supported definitions of grams as used by the at least \( l \)-frequent \( \text{dir} \)-directed \( k \)-skip \( n \)-grams, and rewrite-rules feature extraction methods.

| Gram   | Compound | Generalized \( n \)-grams | Rewrite rules | Languages          |
|--------|----------|----------------------------|---------------|--------------------|
| letter | No       | Yes                        | No            | Any                |
| word   | No       | Yes                        | Yes           | Any                |
| wordlen| No       | Yes                        | Yes           | Any                |
| wordshape| No    | Yes                        | Yes           | Any with A-Z        |
| wordshape -word | Yes | Yes                        | Yes           | Any with A-Z        |
| funcword| No       | Yes                        | Yes           | English            |
| pos    | No       | Yes                        | Yes           | Arabic, Chinese, French, German, and Spanish |
| dep    | No       | Yes                        | No            | Arabic, Chinese, French, German, and Spanish |
| word-pos| Yes    | Yes                        | No            | Arabic, Chinese, French, German, and Spanish |
| word-dep| Yes    | Yes                        | No            | Arabic, Chinese, French, German, and Spanish |
| pos-dep| Yes      | Yes                        | No            | Arabic, Chinese, French, German, and Spanish |

given \( n \)-gram by the total number of occurrences of all \( n \)-grams. If set to \text{False}, then raw frequency counts are returned.

- \( l \): The minimum frequency an \( n \)-gram must have in order for the library to list it. E.g., if \( l \leq 1 \), then all identified \( n \)-grams are returned. If \( l = 5 \), then only those that occur for at least five times are returned.
- \( n \): The size of the sliding window of \( n \)-grams, in the unit of grams.
- Gram: The definition of a gram. Table 13 presents a list of supported grams.
- Cache (path): If set to \text{None}, then caching is disabled. If set to a path, then caching is enabled. This can be useful for expensive features, such as those that require making use of POS taggers (the cache will save time by avoiding parsing same sentences twice).
- \( k \)-skip \( n \)-grams, with parameters:
  - \( k \): the total number of tolerated adjacency violations in an \( n \)-gram, in the unit of grams. E.g., \( k = 2 \) will tolerate up to 2 adjacency violations, while \( k = 0 \) will not tolerate any and cause it to be identical to classical \( n \)-grams.
  - Normalize (Boolean).
  - \( l \).
  - \( n \).
  - Gram.
  - Cache (path).
- Syntactic \( n \)-grams when (using dependency trees) [16], with parameters:
  - Normalize (Boolean).
  - \( l \).

Rewrite-rules. Unlike other rewrite-rules implementations, ours has the novelty in that it allows us to substitute the terminal words by their alternative forms (e.g., word shape). For consistency, we refer to this as “gram”. Additionally, compound grams are also made available to the rewrite-rules feature extraction function. The parameters are:

- Normalize (Boolean).
- \( l \).
- Gram. Table 13 lists all supported gram definitions.
- Cache (path).

Richness measures (Hapax legomena, unique words rate).

B. GENERALIZATION OF \( n \)-GRAM METHODS

This section presents the mechanism by which our library implements \( n \)-grams, \( k \)-skip \( n \)-grams, and syntactic \( n \)-grams. In order to simplify the implementation, enhance the ability to introduce more novel variants, as well as extend the coverage of the library, we have generalized all of the \( n \)-gram-based methods as the at least \( l \)-frequent \( \text{dir} \)-directed \( k \)-skip \( n \)-grams. Then, we implemented this generalization instead. As a result of this, we get a more-extensive library that is also simpler and allows superior code re-use. Further details are presented below.

Consider the text example that is presented in Figure 14, and, for simplicity, suppose that grams are defined to be words. Then, if the parameter \text{dir} = \text{spatial}, the example text in Figure 14 is represented in the following row matrix in (1).

\[
\begin{array}{l}
\text{The quick fox jumped over the lazy dog} \\
\end{array}
\] (1)

Then, the sliding window, as depicted in Figure 1, will operate on the matrix in (1) on a row-by-row basis. Since there is a single (but long) row, the sliding window will move along that one row, depending on the chosen value of parameter \( n \).

On the other hand, if the parameter \text{dir} = \text{deptree}, the example text in Figure 14 is represented in the following row matrix in (2). Note that each row represents a path from the \text{root} node towards the numerous leaf nodes as we walk.
down the dependency tree that is depicted in the Figure.

\[
\begin{bmatrix}
\text{jumped} & \text{fox} & \text{quick} \\
\text{jumped} & \text{fox} & \text{The} \\
\text{jumped} & \text{dog} & \text{over} \\
\text{jumped} & \text{dog} & \text{the} \\
\text{jumped} & \text{dog} & \text{lazy}
\end{bmatrix}
\]  

(2)

Then, similar to the \( \text{dir} = \text{spacial} \) case, the sliding window, as depicted in Figure 1, will operate on the matrix in (2) on a row-by-row basis. Since there are 5 rows, the sliding window will move along each row, independently. It can be seen that, because of this design, we are able to re-use our sliding window code for both \( \text{dir} = \text{spacial} \) (classical \( n \)-grams) and \( \text{dir} = \text{deptree} \) (syntactic \( n \)-grams with dependency trees).

Alternatively, one may decide to construct a matrix similar to the one in (1), but while using a different type of trees, or methods that might not necessarily be based upon linguistics basis. Extending this library is as simple as introducing code that defines a matrix out of sentences.

For completeness, (3) and (4) present variants of the matrices (1) and (2), respectively, except for defining grams to be POS tags.

\[
\begin{bmatrix}
\text{DT} & \text{JJ} & \text{NN} & \text{VBD} & \text{IN} & \text{DT} & \text{JJ} & \text{NN} \\
\text{VBD} & \text{NN} & \text{JJ} \\
\text{VBD} & \text{NN} & \text{DT} \\
\text{VBD} & \text{NN} & \text{IN} \\
\text{VBD} & \text{NN} & \text{DT} \\
\text{VBD} & \text{NN} & \text{JJ}
\end{bmatrix}
\]  

(3)

\[
\begin{bmatrix}
\text{DT} & \text{IN} & \text{NN} & \text{VBD} & \text{IN} & \text{DT} & \text{JJ} & \text{NN} \\
\text{VBD} & \text{NN} & \text{JJ} \\
\text{VBD} & \text{NN} & \text{IN} \\
\text{VBD} & \text{NN} & \text{DT} \\
\text{VBD} & \text{NN} & \text{IN} \\
\text{VBD} & \text{NN} & \text{JJ}
\end{bmatrix}
\]  

(4)

As for the parameter \( k \) that specifies the total number of permissible gram skips, it is implemented in the sliding window code and is therefore fully re-used elsewhere, independent of the direction. Therefore, the rest of the code is re-used, independent of how the matrices are defined.

C. EXAMPLES

The example below demonstrates how to obtain the unique words richness measures for the example in Figure 14, which prints 0.875.

```python
# load the library
import fextractor

# define some text
text = 'the quick fox jumped over the lazy dog'

# find the CFG rules
r = fextractor.getcount_rewriterules(text, 'en',
grammar='word', normalize=False, server='http://corenlp.run')

# print the score
print(r)
```

Which prints the rewrite-rules:

```python
1. ["u' DT ' , [ u ' the ' ]] : 2 ,
2. ["u' IN ' , u ' over ' ] : 1 ,
3. ["u' JJ ' , u ' lazy ' ] : 1 ,
4. ["u' J J ' , u ' quick ' ] : 1 ,
5. ["u' NN ' , u ' dog ' ] : 1 ,
6. ["u' NN ' , u ' fox ' ] : 1 ,
7. ["u' NP ' , u ' DT ' , u ' JJ ' , u'NN' ] ] : 2 ,
8. ["u' PP ' , [ u ' IN ' , u ' NN ' ] ] : 1 ,
9. ["u' ROOT ' , [ u ' S ' ] ] : 1 ,
10. ["u' S ' , [ u ' NP ' , u ' VP ' ] ] : 1 ,
11. ["u' VBD ' , [ u ' jumped ' ] ] : 1 ,
12. ["u' VP ' , [ u ' VBD ' , u ' PP ' ] ] : 1 ]
```

If \( \text{gram}='\text{wordlen}' \), then terminal nodes are replaced by their lengths as follows:

```python
1. ["u' DT ' , [ u ' the ' ]] : 2 ,
2. ["u' IN ' , [ u ' over ' ]] : 1 ,
3. ["u' JJ ' , u ' lazy ' ] : 1 ,
4. ["u' J J ' , u ' quick ' ] : 1 ,
5. ["u' NN ' , [ u ' 6 ' ] ] : 1 ,
6. ["u' NP ' , u ' DT ' , u ' JJ ' , u'NN' ] ] : 2 ,
7. ["u' PP ' , [ u ' IN ' , u ' NP ' ] ] : 1 ,
8. ["u' ROOT ' , [ u ' S ' ] ] : 1 ,
9. ["u' S ' , [ u ' NP ' , u ' VP ' ] ] : 1 ,
10. ["u' VBD ' , [ u ' jumped ' ] ] : 1 ,
11. ["u' VP ' , [ u ' VBD ' , u ' PP ' ] ] : 1 ]
```

The example below demonstrates how to represent the same example sentence as a matrix based on syntactic \( n \)-grams.

```python
# load the library
import fextractor

# define some text
text = 'the quick fox jumped over the lazy dog'

# find the syntactic n-gram matrix
m = fextractor.get_grams(text, 'en',
gram='pos', direction='deptree')

# print the score
print(m)
```

Which give the output (note how the output resembles the matrix in 3):

```python
1. ["u'VBD' , u'NN' , u' DT' ] ,
2. ["u'VBD' , u'NN' , u' JJ' ] ,
3. ["u'VBD' , u'NN' , u' IN ' ] ,
4. ["u'VBD' , u'NN' , u' DT' ] ,
5. ["u'VBD' , u'NN' , u' JJ' ]
```

The matrix can then be used in order to identify sequences of grams. Below is an example for identifying all 2-skip 2-grams, along with their normalized frequencies.

```python
# load the library
import fextractor

# define some text
text = 'the quick fox jumped over the lazy dog'

# find the syntactic n-gram matrix
m = fextractor.get_grams(text, 'en',
gram='pos', direction='deptree')

# print the score
print(m)
```

4https://stanfordnlp.github.io/CoreNLP/
# count raw k–skip n–grams patterns with raw frequencies
p = fextractor.getcount_ksgrams(m, k=2, n=2, normalize=False)

# print the score
print(p)

Which prints the output:
1. {u 'NN : : DT ' : 0.13, u 'NN : : IN ' : 0.06, u 'NN : : JJ ' : 0.13, u 'VB : : DT ' : 0.13, u 'VB : : IN ' : 0.13, u 'VB : : JJ ' : 0.13, u 'VB : : NN ' : 0.33}

If normalize=True, then the output would be normalized as follows:
1. {u 'NN : : DT ' : 0.13, u 'NN : : IN ' : 0.06, u 'NN : : JJ ' : 0.13, u 'VB : : DT ' : 0.13, u 'VB : : IN ' : 0.13, u 'VB : : JJ ' : 0.13, u 'VB : : NN ' : 0.33}

Additionally, if using a vector-representation is required, the represented texts can be trivially converted into vectors. Below is a code of an example where two distinct texts, text1 and text2, are transformed into a vector space.

# load the library
import fextractor

# define some text
text1 = 'the quick fox jumped over the lazy dog'
text2 = 'this is an example of a different text'

# find the syntactic n-gram matrix
ml = fextractor.get_grams(text1, 'en', gram='pos', direction='deptree')
m2 = fextractor.get_grams(text2, 'en', gram='pos', direction='deptree')

# count raw k–skip n–grams patterns with raw frequencies
pl = fextractor.getcount_ksgrams(ml, k=2, n=2, normalize=True)
p2 = fextractor.getcount_ksgrams(m2, k=2, n=2, normalize=True)

# agree on some order for the components
pmaster = {i for i in list(pl) + list(p2)}
features_order = list(pmaster)

# represent them as vectors
x1 = []
x2 = []
for f in features_order:
    if f in pl:
        x1.append(pl[f])
    else:
        x1.append(0)
for f in p2:
    if f in p2:
        x2.append(p2[f])
    else:
        x2.append(0)

# print the vectors
print('x1 = ' + str(x1))
p

Which will print the vector-representation of the texts as follows:
1. x1 = [0.13, 0.13, 0.06, 0.00, 0.13, 0.13, 0.00, 0.33, 0.06]
2. x2 = [0.00, 0.16, 0.16, 0.08, 0.33, 0.00, 0.25, 0.00, 0.00]

Such vectors could then be used by other classifiers as required.

IX. CONCLUSION
A key contribution of this paper relates to the uncertainty that is associated with the applicability of the stylometry problemsolvers against the domain of Emirati Arabic texts. To work towards addressing this issue, we have constructed the KIT-30 dataset, which is the first Emirati social media author-identification evaluation dataset. Interestingly, our studies found that the scalability issues of AA problem solvers, that are generally reported in the literature concerning the size of the suspect authors space, is noticeably more aggressive than what our findings indicate. For example, we were able to achieve a classification accuracy of over 0.98 when solving AA problems that were constructed based on chunks of Emirati tweets, with a set of 30 suspect authors. This accuracy is notably higher than evaluated in the literature on similar size spaces of suspect authors in literature [7], [17], especially when knowing that our chunks of tweets, per author, remained relatively small (only a few hundreds of tweets per chunk).

Additionally, in order to work towards addressing the lack of conveniently-available implementations of stylometry methods, we have developed an extensive e-text feature extraction library, with a highly intuitive API. This library offers, by far, the most extensive set of e-text stylometry feature extraction methods to date, which is partly thanks to our generalization of n-gram-based feature extraction methods. The library also contains a number of novelties, such as novel definitions of grams (e.g., compound grams) for both, n-gram-based methods, as well as CFG rewrite-rules. Interestingly, when using our feature extraction library, our evaluation of efficient AA solvers against Emirati tweets AA problems indicates that the use of compound grams allows for the identification of more accurate AA models.

Supplementary
- The implementation of the Tweets downloader in Step 1 of Algorithm 2 is released under permissible open-source license at http://khonji.org/stylometry.
- The KIT-30 dataset is available at https://gitlab.com/mmaakh/kit-30.git.
- The feature extraction library, Fextractor, is available at https://gitlab.com/mmaakh/fextractor.git.
- The evaluation code is available at https://gitlab.com/mmaakh/stylometry-survey-evaluation.git.

ACKNOWLEDGMENT
The authors of this article are highly grateful to the Khalifa University students, Z. Al-Blooshi and F. Al-Hammadi, for their generous help in identifying Emirati Twitter accounts to extend the KIT-30 dataset.

REFERENCES
[1] W. Daelemans, “Explanation in computational stylometry,” in Proc. 14th CICLing, Berlin, Germany: Springer-Verlag, vol. 2, 2013, pp. 451–462.
[2] D. Kernot, T. Bossomaier, and R. Bradbury, “The stylometric impacts of ageing and life events on identity,” J. Quant. Linguistics, vol. 26, no. 1, pp. 1–21, Jan. 2019.

[3] P. Juola, “Authorship attribution,” Found. Trends Inf. Retrieval., vol. 1, no. 3, pp. 233–334, Dec. 2006.

[4] M. Koppel, J. Schler, and S. Argamon, “Computational methods in authorship attribution,” J. Amer. Soc. Inf. Sci. Technol., vol. 60, no. 1, pp. 9–26, Jan. 2009.

[5] E. Stamatatos, “A survey of modern authorship attribution methods,” J. Amer. Soc. Inf. Sci. Technol., vol. 60, no. 3, pp. 538–556, Mar. 2009.

[6] M. Eder, J. Rybicki, and M. Kestemont, “Stylometry with R: A package for computational text analysis,” R J., vol. 8, no. 1, pp. 107–121, Aug. 2016. [Online]. Available: http://journal.r-project.org/archive/2016-1/eder-rybicki-kestemont.pdf

[7] A. Rocha, W. J. Scheier, C. W. Forstall, T. Cavalcante, A. Theophilou, B. Shen, A. R. B. Carvalho, and E. Stamatatos, “Authorship attribution for social media forensics,” IEEE Trans. Inf. Forensics Security, vol. 12, no. 1, pp. 5–33, Jan. 2017.

[8] P. Juola, “An overview of the traditional authorship attribution subtask,” in Proc. CLEF Eval. Labs Workshop Working Notes Papers, P. Forner, J. Karlsgren, and C. Womser-Hacker, Eds. Rome, Italy: CEUR-WS.org, Sep. 2012, pp. 37–41.

[9] M. Khonji, Y. Iraqi, and A. Jones, “An evaluation of authorship attribution using random forests,” in Proc. Int. Conf. Inf. Commun. Technol. Res. (ICTRC), May 2015, pp. 68–71.

[10] M. Khonji and Y. Iraqi, “Attributing authors of emirati tweets,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2018, pp. 206–212.

[11] M. Nabil, M. Aly, and A. Atiya, “ASTD: Arabic sentiment tweets dataset,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2015, pp. 2513–2519. [Online]. Available: http://aclweb.org/anthology/D/D15/D15-1299.pdf

[12] D. Guthrie, B. Allison, W. Liu, L. Guthrie, and Y. Wilks, “A closer look at skip-gram modelling,” in Proc. 5th Int. Conf. Lang. Resour. Eval. (LREC), Genoa, Italy, 2006, pp. 1222–1225.

[13] E. W. Noreen, Computer-Intensive Methods for Testing Hypotheses: An Introduction, Hoboken, NJ, USA: Wiley, Apr. 1989.

[14] E. Stamatatos, W. Daelemans, B. Verhoeven, B. Stein, M. Potthast, P. Juola, M. A. Sánchez-Pérez, and A. Barrón-Cedeño, “Overview of the author identification task at PAN 2014,” in Proc. CLEF Eval. Labs Workshop Work. Notes Papers, Sheffield, U.K., Sep. 2014, pp. 877–897.

[15] M. Khonji and Y. Iraqi, “A slightly-modified GI-based Author-verifier with lots of features (ASGALF),” in Proc. CLEF Eval. Labs Workshop Work. Notes Papers, L. Cappellato, N. Ferro, M. Halvey, and W. Kraaij, Eds. Sheffield, U.K.: CEUR-WS.org, Sep. 2014, pp. 15–18.

[16] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, Advances in Computational Intelligence, Berlin, Germany: Springer, 2013, ch. Syntactic Dependency-Based N-grams as Classification Features, pp. 1–11.

[17] M. Brennan, S. Afroz, and R. Greenstadt, “Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity,” ACM Trans. Inf. Syst. Secur., vol. 15, no. 5, pp. 12:1–12:22, Nov. 2012.

MAHMOUD KHONJI (Member, IEEE) received the Masters by Research and Ph.D. degrees in engineering from Khalifa University, in 2012 and 2017, respectively, where he worked on various text classification problems relating to security, forensics, and privacy.

YOUSSEF IRAQI (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from the University of Montreal, in 2000 and 2003, respectively. From 2004 to 2005, he was a Research Assistant Professor with the David R. Cheriton School of Computer Science, University of Waterloo, Canada. He is currently an Associate Professor with the ECCS Department, Khalifa University, UAE. Before that, he was the Chair of the Computer Science Department, Dhofar University, Oman, for four years. He has published more than 110 research articles in international journals and refereed conference proceedings. His research interests include adaptive resource management in multimedia wireless networks, trust and reputation management, cloud computing, and stylometry. In 2008, he received the IEEE Communications Society Fred W. Ellersick Paper Award in the Field of Communications Systems. He is on many technical program committees of international conferences and always approached for his expertise by international journals in his field.

* * *