A Visualizable Evidence-Driven Approach for Authorship Attribution

STEVEN H. H. DING, Concordia Institute for Information Systems Engineering, Concordia University
BENJAMIN C. M. FUNG, School of Information Studies, McGill University
MOURAD DEBBABI, Concordia Institute for Information Systems Engineering, Concordia University

The Internet provides an ideal anonymous channel for concealing computer-mediated malicious activities, as the network-based origins of critical electronic textual evidence (e.g., emails, blogs, forum posts, chat logs, etc.) can be easily repudiated. Authorship attribution is the study of identifying the actual author of the given anonymous documents based on the text itself, and for decades, many linguistic stylometry and computational techniques have been extensively studied for this purpose. However, most of the previous research emphasizes promoting the authorship attribution accuracy, and few works have been done for the purpose of constructing and visualizing the evidential traits. In addition, these sophisticated techniques are difficult for cyber investigators or linguistic experts to interpret. In this article, based on the End-to-End Digital Investigation (EEDI) framework, we propose a visualizable evidence-driven approach, namely VEA, which aims at facilitating the work of cyber investigation. Our comprehensive controlled experiment and the stratified experiment on the real-life Enron email dataset demonstrate that our approach can achieve even higher accuracy than traditional methods; meanwhile, its output can be easily visualized and interpreted as evidential traits. In addition to identifying the most plausible author of a given text, our approach also estimates the confidence for the predicted result based on a given identification context and presents visualizable linguistic evidence for each candidate.

Categories and Subject Descriptors: K.4.1 [**Computers and Society**]: Public Policy Issues—*Abuse and crime involving computers*; I.7.5 [**Document and Text Processing**]: Document Capture—*Document analysis*; I.2.7 [**Natural Language Processing**]: Text Analysis

General Terms: Design, Algorithms, Experimentation

Additional Key Words and Phrases: Authorship attribution, computational linguistics, text mining, cyber forensics

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

1. INTRODUCTION

Research in authorship attribution on anonymous documents is experiencing a continuing exponential growth in recent years because a reliable authorship attribution technology is useful and valuable in many fields, such as literary science, sociolinguistic research, Psycholinguistics, social psychology, forensics, and medical diagnosis

The research is supported in part by Discovery grant 356065-2013 from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Authors' addresses: S. H. H. Ding and B. C. M. Fung (corresponding author), School of Information Studies, McGill University, Montreal, QC, Canada H3A 1X1; emails: steven.h.ding@mail.mcgill.ca, ben.fung@mcgill.ca; M. Debbabi, Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada H3G 1M8; email: debbabi@ciise.concordia.ca.

DOI: http://dx.doi.org/10.1145/2699910

[Daelemans 2013]. Especially under the globalized and decentralized nature of the Internet, the communications of malicious activities (e.g., illegal material distribution, ransom, and harassment [Abbasi and Chen 2008; Iqbal et al. 2013]) can be easily hidden or repudiated. Authorship analysis techniques are capable of delving into the information from different linguistic levels and of identifying the textual identity trace, which potentially greatly facilitates the work of cyber forensic investigators and sustains the social accountability. Stylometry even has been employed as evidence in a court of law [Brennan et al. 2012].

The study of authorship attribution has a long-standing history [Mosteller and Wallace 1964], and many linguistic stylometry and computational techniques have been developed for solving this problem. These methods have demonstrated outstanding effectiveness in identifying the actual authors; however, those techniques that achieve the highest accuracy always involve sophisticated, obscure computational models [Stamatatos 2009], and their output is too simple to use as evidence in courts of law. Some of these models, such as neural network and support vector machine (SVM), can be hardly interpreted by an investigator as black-box approaches. Other relatively simple models also require time and resources to obtain a justifiable result through manual inspection.

These issues handicap traditional methods from being widely applied to the real-life lawsuits as convincing evidence. Practically, computational stylometry is calling for “more explanation as opposed to purely quantitative measure” [Daelemans 2013]. A better approach should provide explainable and presentable convincing traces as evidence.

Most of the previous research did not measure the degradation of their methods’ performance, as the quantity/quality of the available information degraded simultaneously, which is also noted by Solan [2013]. These models are mostly evaluated only on formal writings, which are relatively long, informative, well-structured, and free from grammatical errors. On the contrary, short snippets are relatively casual, and their stylometric features have larger variation. As shown in recent research [Koppel et al. 2011; Luyckx and Daelemans 2011; Narayanan et al. 2012], authorship attribution accuracy is greatly and directly affected by many objective factors (text length, number of known author samples, etc.) due to the unstructured nature of the text itself. It is critical for authorship analysis researchers to conduct attribution evaluation experiments in varying attribution scenarios to “exclude a bogus conclusion based on inadequate data” [Solan 2013] when applied to real-life legal cases.

In this article, we present a visualizable evidence-driven approach, namely VEA, for the purpose of facilitating the work of cyber investigation and the decision-making process in a court of law. Our approach is driven by evidence and based on the lazy learning scheme [Narayanan et al. 2012]. Basically, our method searches inside the anonymous document for all writing styles of different linguistic modalities as evidence and matches them to the prebuilt candidate profiles. Evidence from different linguistic modalities are combined by using confidence estimation. Finally, it visualizes all of the evidence on the given hypotheses, and it is able to present a visual discrimination between hypotheses. Besides, it also provides an estimated confidence value based on the quality of the evidence and the amount of available information in a given attribution scenario. More importantly, we modeled the attribution scenario and conducted our experiments in varying situations (varying length of text, varying candidate size, etc.) to fully evaluate our method.

In the authorship attribution problem, a set of candidate authors, along with their corresponding individual writing samples, are available, and the task is to identify the most plausible author among these candidates based on the given anonymous document [Mosteller and Wallace 1964; Holmes 1998; Iqbal et al. 2013]. In most of the previous studies, the candidate sets involved in their scenarios are mostly of size ranging from 2 to 20. Some recent studies [Koppel et al. 2011; Narayanan et al. 2012] present the authorship attribution problem with thousands of possible candidate authors and try to solve it in a scalable way. In this case, it is more appropriate to first employ scalable methods from these studies to determine a potential candidate subset, then use other relatively more accurate techniques to figure out the most plausible conclusion and derive the justifiable result.

An open-set authorship attribution problem is a variant of the original authorship attribution problem [Koppel et al. 2011]. In this research problem, the solution is allowed to output an alternative "unknown" option to indicate that the actual author could not be found or determined from the given candidate set based on presented available information. In fact, any solutions that are capable of outputting a monotonous probability indicating the confidence of a predicted result can be applied to this problem by setting an appropriate threshold on this output probability value.

The authorship attribution problem is similar to the text classification problem. The plain text classification task is tough, inherently due to unstructured nature of textual data. By unifying the feature vector and extracting the vector for each sample text, the textual data can be transformed into structured samples, which is the typical and traditional authorship attribution solution [Holmes 1994; Stamatatos 2009]. However, the deviation of each element inside the vector is still strongly affected by the length of available text. Online texts are mostly very short and therefore contain limited information about the writing style [Iqbal et al. 2013], which causes a larger fluctuation around the mean value in the unified feature vector. This introduces difficulties in achieving higher accuracy due to the presence of more outliers.

To retain reasonable accuracy in the identification task, we try to maximize the information gained from the given anonymous document and combine both statistical similarity and data mining techniques to develop a hybrid model using the lazy learning mechanism. Specifically, our contributions are summarized as follows:

—To the best of our knowledge, this is the first trial to design an authorship attribution approach with the goal of promoting not only the accuracy measure but also the interpretability and the visualizability of the predicted result. From the very beginning, this approach is designed from the perspective of collecting evidence. We systematically outlined our approach by employing the End-to-End Digital Investigation (EEDI) framework [Bosworth et al. 2012], one of the recognized forensic processes used in digital forensics investigations. By doing this, we are able to construct a cumulative evidentiary effect supporting the final output result, and the construction process can be easily explained using the EEDI framework.

—Our approach is concise in design, and its output is visualizable. Inspired by the visualization of fingerprint matching in Figure 1, where the correlations among fingerprint minutiae can be visually compared, we devise an approach visualizing all supporting evidence on top of our visual representation of hypotheses rather than presenting a simple numeric result. We are able to present a visual discrimination among these hypotheses and present detailed supporting evidence. More importantly, we systematically conducted our experiments under varying authorship attribution scenarios to fully evaluate our approach. Our experiments demonstrate that our approach achieves state-of-art attribution accuracy, and the output evidence is visualizable, presentable, and explainable.

—Based on the specific context of the given authorship attribution problem, our approach is also able to estimate a confidence value. Based on those scenario-related features that we identified, our method can accurately model and predict the final classification accuracy. Moreover, to our best knowledge and differing from previously employed voting-based ensemble methods such as Koppel et al. [2011], it is the first trial to combine multiple classifiers by normalizing their scoring vector

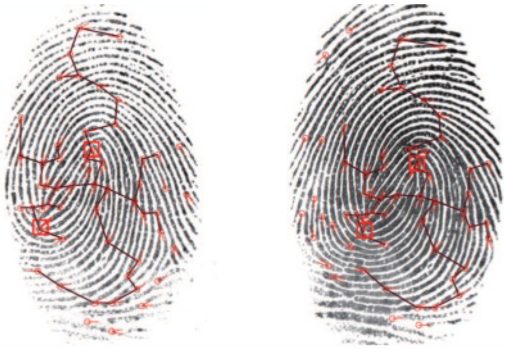

Fig. 1. A sample fingerprint minutiae matching diagram generated by using fingerprint software and data from *NEUROtechnology*.[1]

using individually estimated confidence values on given classification contexts. We consider classifiers built on features of different linguistic modalities separately. We explain the necessity of this step by arguing that stylistic features from different linguistic modalities have different capacity in determining the actual author and varying sensitivity to the objective conditions in a given scenario. This is due to the unpredictable coherence of writing style among known authors' sample writings, and it is in accordance with our observations in the experiments. In addition, our approach is extensible, where other features from different linguistic modalities or nonlinguistic features can be further added as additional events.

The rest of this article is organized as follows. Section 2 reviews and discusses recent development and issues in authorship analysis. Section 3 elaborates our visualizable evidence-driven approach of authorship attribution in detail. Section 4 evaluates our proposed method—VEA—on the Enron real-life dataset. Section 5 concludes the article.

2. RELATED WORKS

The history of authorship attribution backed up by computational and statistical methods can be dated from the 19th century [Stamatatos 2009]. Contributions to this area can be broadly categorized from three aspects: the involved stylometric features, the employed attribution techniques, and the attacks against authorship attribution techniques. Previous research mainly focuses on promoting quantitative evaluation, and limited research has been done for visualization or explanation. Most explanations for the choice of features and algorithmic parameters are simply driven by the classification accuracy. In this section, we will discuss several recent related works and research trends in authorship analysis research. An inclusive survey on the complete history is beyond the scope of this work. Broader comprehensive surveys can be referred to Holmes [1994], Juola [2006], and Stamatatos [2009].

2.1. Stylometric Features

Stylometry is the solution of authorship recognition by investigating the linguistic characteristics inside the given text document, and stylometric features are those linguistic marks that could qualify or quantify these linguistic characteristics [Stamatatos 2009; Brennan et al. 2012]. Stylometric features can be categorized into different linguistic levels [Daelemans 2013; Stamatatos 2009] or, more precisely, linguistic modalities

[1] The *NEUROtechnology* software used to generate this diagram is available at http://www.neurotechnology.com/.

[Sapkota et al. 2013; Solorio et al. 2011]. Various features of different modalities have demonstrated their effectiveness in distinguishing human writing patterns, including lexical [Koppel et al. 2006; Halteren 2007; Savoy 2012], character-based [Koppel et al. 2011, 2012; Escalante et al. 2011], syntactic [Kim et al. 2011; Sidorov et al. 2013; Raghavan et al. 2010], semantic [Hedegaard and Simonsen 2011; Seroussi et al. 2011, 2012], and application-specific modalities [Cristani et al. 2012].

Among all of these stylometric features, the *character n-gram model* in character-based linguistic modality performs the best, and it is comparatively more robust against the others [Luyckx and Daelemans 2011; Koppel et al. 2011]. The character *n*-gram model actually captures information crossing different modalities [Houvardas and Stamatatos 2006]; for example, a frequent ‘ed’ bigram in a character-based modality may also carry the frequent usage of past tense in a syntactic modality. However, as pointed out in Narayanan et al. [2012], solutions using these features also take the risk of capturing the context rather than the authors’ writing style. Regarding the relationship between stylometric modalities, Sapkota et al. [2013] employed the word *orthogonal* to assimilate them as independent components. To the best of our knowledge, the correlation among linguistic modalities has not formally been investigated in previous authorship studies. In this work, we are not going to evaluate whether correlations may exist among linguistic modalities, but we argue that they have different capacity in attributing the correct author based on the given problem context.

Stylometric feature sets involved in previous studies can also be divided into two groups: the unified feature set and the class-specific feature set. Under the unified feature set, which is employed by most previous solutions, every candidate is modeled using the same set of features; however, under the class-specific feature set, candidates are given distinct feature sets and a model is learned for each candidate. As shown by Abbasi and Chen [2008] and Iqbal et al. [2013], the distinct algorithmic feature set can better distinguish among candidates’ writing styles and achieve higher performance.

2.2. Attribution Techniques

After the selection of the specific feature scheme, attribution techniques are employed to predict the actual author of a given snippet. Attribution techniques can be divided into the similarity-based approach [Peng et al. 2003; Halteren 2007; Koppel et al. 2011] and the model-based approach [Sanderson and Guenter 2006; Lambers and Veenman 2009]. The similarity-based approach employs distance functions [Savoy 2012] to quantify the proximity between a candidate profile and a given anonymous document, whereas the model-based approach builds complicated models to classify the given document. For the supervised–unsupervised distinction in model learning, previous methods fall into to the supervised and semisupervised categories. Those solutions that achieve the best performance on benchmark datasets are mostly related to machine learning models.[2] Among the model-based approaches, the SVM-based approach [Abbasi and Chen 2008] and the association-rule–based approach [Iqbal et al. 2013] achieve higher accuracy because they both consider the combination of feature values among the high-dimensional space. Other machine learning models, including decision tree, neural network [Tweedie et al. 1996], metalearning [Koppel et al. 2007], and clustering [Layton et al. 2013], are also employed to solve the problem of authorship attribution. Typically, a one-versus-all SVM is chosen as the standard method when comparing different stylometric features because it has a better multiclass classification capacity [Duan and Keerthi 2005].

Even though a model-based approach can achieve higher quantitative performance, most such approaches involve a complicated computational model, and it is difficult to

[2] Contest organized in 2004 ALLC/ACH.

interpret its decision-making process. The similarity-based approach is much easier to visualize and interpret because it retains a monotonous linear relationship between evidence and conclusion: the smaller the distance between author profile and the targeted document, the more similar writing styles they possess.

2.3. Ensemble Method

Recent studies in authorship analysis demonstrate a trend of employing ensemble methods to combine several separately trained classifiers due to the fact that multiple classifiers can better fit into sample data and boost the attribution accuracy. In Koppel et al. [2011], multiple classifiers are built based on different feature sets that are randomly selected from all available space-free character 4-grams, and the final output depends on their votes. In Kourtis and Stamatatos [2011], a co-training approach is employed by using two classifiers. In Narayanan et al. [2012], an agreement-based combination of the nearest neighbor model and the SVM model achieve higher identification accuracy for blog data. Additionally, in Raghavan et al. [2010], higher performance is achieved by employing the votes from classifiers built on different feature sets.

However, all of these works consider classifiers equally weighted. Based on different classification contexts (the length of an anonymous snippet, candidate score distribution, etc.), classifiers built by using features of varying linguistic modalities will have varying capacity to attribute the author correctly. It is more rational to weight them accordingly: under the specific classification context, the one that can better discriminate writing style should be weighted more. In our approach, each classifier is built based on features from different linguistic modalities and is weighted based on its demonstrated consistency among prior written samples. In machine learning literature, there are lots of works that have studied the *boosting*, *stacking*, and *ensemble* methods, but few of them have been applied to the authorship attribution problem. Our purpose is not to show that our approach is advantageous over these approaches, but rather we try to illustrate that such an approach can promote the prediction accuracy and the interpretability. As a whole, it is a one-step-forward real-life application of authorship attribution techniques. A full comparison between our approach and other stacking approaches is outside the scope of this work.

2.4. Adversary Stylometry

From the perspective of the adversary, several studies are trying to circumvent authorship attribution techniques [Kacmarcik and Gamon 2006; Juola and Vescovi 2010; Brennan et al. 2012]. The most influential study is by Brennan et al. [2012]. They conducted an experiment on the effectiveness of stylometry obfuscation and imitation. By recruiting volunteers and using the Amazon Mechanical Turk[3] platform, they asked participants to submit their prior written samples and then write an imitation passage and an obfuscation passage (no guideline was given to participants on how to obfuscate or imitate). Their results demonstrate that there is a significant drop in identification accuracy when it comes to these attacks. In addition, the accuracy drops when it comes to one-step, two-step translation attacks.

However, their experimental setup may not truly reflect the effectiveness of their obfuscating approach. First, the decrease in identification accuracy is mostly caused by the mismatch of context between the obfuscated passages and the training passages. Obfuscated passages are about the description of participants' neighbors, whereas preexisting writing samples are mostly "scholarly" and thus are more formal. Second, their experiment also combined and split passages to generate known author writing samples, which may also lead to a high contextual correlation among samples. As we

[3] https://www.mturk.com/mturk/welcome.

know, word-level tokens are good at capturing contextual and thematic correlation [Fung et al. 2003]. We ran our model based on pure lexical *n*-gram on their dataset, and it showed a high correlation of word-level *n*-gram among training samples (86.01% identification accuracy for 45 authors; around 500 tokens per sample), with a low correlation between obfuscated texts and training texts. Additionally, in the study of Juola [2012], a method for detecting the obfuscated texts is proposed using character 3-grams and word 3-grams. Their experiments also demonstrated a large difference in *n*-gram usage between pre-existing samples and obfuscated samples. The difference in the *n*-gram usage pattern implies the contextual and thematic variations, which naturally leads to the unsatisfactory result when it comes to authorship attribution techniques that employ character bigrams and trigrams.

2.5. Attribution Result and Its Visualization

Most of the aforementioned studies simply display the most plausible candidate as their output result. Some recent research is able to add an estimated value indicating the attribution confidence [Koppel et al. 2011; Narayanan et al. 2012]. However, because authorship analysis techniques are not reliable enough to be widely recognized, this kind of simple output will still raise doubts when applied in real-life cases. Instead, visualized evidence corroborating why this candidate author is selected to be the most plausible one will be more helpful. The only work that we found on formally visualizing attribution output is by Abbasi and Chen [2006]. Nonetheless, the visual representation of the *Writeprint*, which consists of a coordinate graph for each single feature, cannot scale up with large numbers of features, and it is difficult to compare different *Writeprints* holistically.

3. VISUALIZABLE EVIDENCE-DRIVEN APPROACH FOR AUTHORSHIP ATTRIBUTION

In this section, we present our visualizable evidence-driven approach for the authorship attribution problem, addressing the issues and problems mentioned in Section 1. For the purpose of promoting its interpretability and explainability, our approach is designed according to the nine processes defined by the EEDI framework [Bosworth et al. 2012]. Considering that every digital crime fundamentally consists of a source point and a destination point, the EEDI framework is a structured flow of processes to establish an evidence chain connecting these two points. EEDI is a popular framework employed by digital investigators due to its capacity for structurally organizing multiple evidence sources to test the conclusion.

We design our approach by adopting the EEDI framework based on the fact that the authorship attribution problem can also be fundamentally regarded as consisting of two points: hypothesis and conclusion. By elaborating the linguistic evidences to establish an evidentiary chain, we can connect these two points together and thus enable our approach to present the completed chain as visualized evidence. In addition, the process of chain construction can be easily explained by employing the EEDI framework. The briefs of procedures employed are outlined in Figure 2.

To begin with, we formally define the authorship identification problem with a probability confidence value output, as mentioned in Section 1. To be consistent in terminology, in this article, “candidates” or “candidate authors” refers to the potential authors of the anonymous message, and “author” or “actual author” refers to the true author of the anonymous message. Let $C = \{C_1, C_2, \ldots, C_N\}$ be a set of N candidate authors and $M = \{M_1, M_2, \ldots, M_N\}$ be a set of their corresponding writing samples, where M_i denotes the set of known samples authored by C_i. The task is to identify the actual author of given anonymous snippet ω from the candidate set C based on the information available in M. Furthermore, the algorithm should be able to output a probability value $p \in [0, 1]$, which denotes the algorithm’s confidence in its predicted result on the

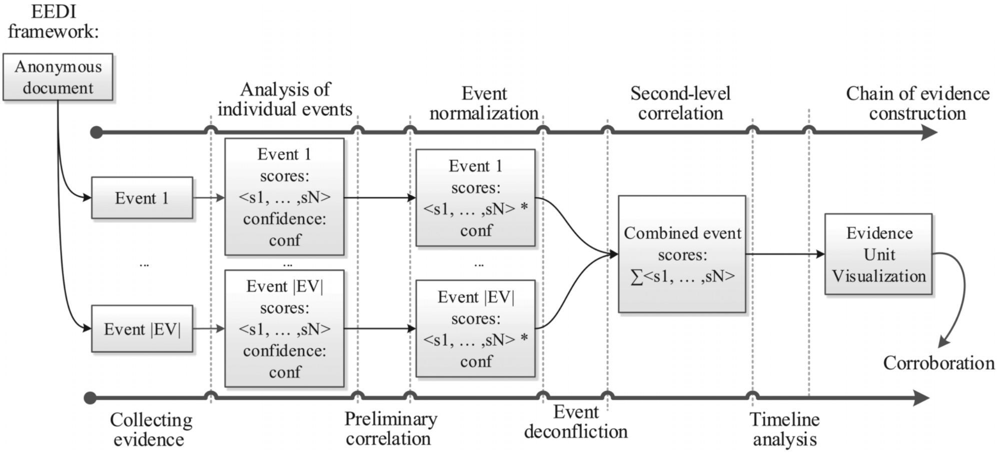
EEDI framework:
Anonymous document
Analysis of individual events
Event normalization
Second-level correlation
Chain of evidence construction
Event 1
Event |EV|
Event 1 scores: <s1, … ,sN> confidence: conf
Event |EV| scores: <s1, … ,sN> confidence: conf
Event 1 scores: <s1, … ,sN> * conf
Event |EV| scores: <s1, … ,sN> * conf
Combined event scores: ∑<s1, … ,sN>
Evidence Unit Visualization
Corroboration
Collecting evidence
Preliminary correlation
Event deconfliction
Timeline analysis

Fig. 2. Overview of VEA in the EEDI framework.

given problem context: $p = 0$ indicates an unreliable result, whereas $p = 1$ indicates a fully reliable result.

At the same time, we also formally define the term *authorship hypothesis* (see Definition 3.1). Basically, an authorship hypothesis is a statement claiming a candidate to be the author of a given anonymous snippet ω. According to the problem defined earlier, where N candidate authors are involved, N hypotheses are thus formulated, respectively targeting on each candidate in C.

Definition 3.1 (*authorship hypothesis*). Given an unknown author snippet ω and a known candidate C_i, a hypothesis in the authorship attribution problem is the statement that candidate C_i authored snippet ω.

3.1. Collecting Evidence

The first phase in the original EEDI framework is collecting evidence [Bosworth et al. 2012]. The purpose of this phase is to detect and collect potential evidence from all available sources of information. The type of evidence may vary. For example, to identify an intrusion, evidentiary types could be logs of system access, logs of network packages, and firewall logs. They require different collection and preprocessing methods. Under the EEDI framework, evidence of different types are grouped together and initiated into independent events, which will be passed to the next process of EEDI.

Accordingly, based on the given anonymous snippet ω, during this phase our task is to identify all linguistic evidence. Likewise, linguistic characteristics reflected on the given snippet ω are of varying types based on their particular linguistic modalities (syntactic, lexical, character based, etc.), and linguistic characteristics of certain modality require specific techniques for feature extraction [Stamatatos 2009]. Thus, we group evidence into independent events based on their linguistic modalities and construct them respectively.

We start this phase by defining the term *evidence unit*. Let $F(\omega) = \{f_1, f_2, \ldots, f_u\}$ denote the universe of writing style features extracted from the anonymous snippet ω. Basically, an evidence unit is defined as one specific writing style feature element with its associated scoring vector (see Definition 3.2). The similarity metric that we employed to describe the correlation between candidate C_i and the linguistic feature f_{eu_m} will be discussed in Section 3.2 (see Equation (3)). Evidence unit is the

Table I. Employed Linguistic Features

Modality	Characteristics	Details	Examples
Lexical	Word-level *n*-gram	Length:1–8	'It,' 'it is,' 'it is noticed,' 'is noticed,' etc.
Character	Character-level *n*-gram	Length:1–8	'no,' 'not,' 'notic,' 'tice,' 'notice,' 'a,' 'an,' 'nd,' etc.
Syntactic	POS *n*-gram	Length:1–8	'PRP VBZ VBN,' 'CC VBN,' 'VBN CC VBN,' etc.

Note: Features in the examples are extracted from the text: '*it is noticed and appreciated*'; the corresponding Part-of-Speech (POS) tag sequence is '*PRP VBZ VBN CC VBN*.' The *n*-gram is extracted in an overlapping manner.

minimum scoring unit and minimum visualization unit, which will be discussed further in Section 3.2.

Definition 3.2 (*evidence unit*). Evidence unit eu_m is formulated as set $\{f_{eu_m}, \vec{v}_{eu_m}\}$: given a certain linguistic feature f_{eu_m}, $\vec{v}_{eu_m} \in \mathbb{R}^N$ is a numeric vector $(v_1, \ldots, v_i, \ldots, v_N)$, where N indicates the number of candidates in C and value v_i indicates the score describing the correlation between candidate C_i and the linguistic feature f_{eu_m}.

The linguistic writing characteristics employed in this article include lexical modality, character modality, and syntactic modality. Specifically, they include lexical word *n*-gram, character-level *n*-gram, and syntactic-level Part-of-Speech (POS) *n*-gram [Stamatatos 2009]. For the POS tagging, we used the pretrained *Maxent* model from Opennlp.[4] Table I provides detailed information and examples. The length of these *n*-grams varies from 1 to 8 because we can hardly find any *n*-gram present repetitively with length more than 8 in the dataset. We employ the *n*-gram technique because previous studies [Koppel et al. 2011; Savoy 2012; Sidorov et al. 2013] show its effectiveness in capturing the writing style. Additionally, they are comparatively easier to visualize and present as evidence units; more details will be discussed in Section 3.5.

ALGORITHM 1: Event Construction (EC)

```
Input number of candidates $N$, linguistic type $Type$, anonymous snippet $\omega$
Output event $ev$
$T_{ev} \leftarrow Type$    ▷ associate this event with the given type of linguistic modality
$features \leftarrow$ extract all linguistic characteristics of type $T_{ev}$ from snippet $\omega$
for $m = 1$ to $|features|$ do
    $\vec{v}_{eu_m^{ev}} \in \mathbb{R}^N,\ \vec{v}_{eu_m^{ev}} \leftarrow \{0\}$    ▷ initialize as a zero vector
    $f_{eu_m^{ev}} = features[m]$    ▷ pair each feature with a new evidence unit
    $EU_{ev} \leftarrow EU_{ev} \cup \{eu_m^{ev}\}$
end for
return $ev$
```

To preserve the explainability of our approach, unlike previous research, we do not employ any feature selection techniques, such as methods found in Yang and Pedersen [1997], meaning that we employ the full set of *n*-grams rather than an optimal top-*K* subset. Previous research, such as Houvardas and Stamatatos [2006], demonstrate that such a top-*K* culled subset can already achieve high accuracy in the authorship attribution problem, but it is difficult to explain why and how this parameter *K*, which indicates the size of employed features, is chosen. In the previous research, the optimal *K* value is learned from the presented experimental results, and it is assumed that this value would work accordingly against other data. Moreover, forensic investigation prefers completeness, and selecting a subset of evidence requires an explanation. Taking the full set can avoid such an issue. Even though this approach introduces high

[4] Available at http://opennlp.apache.org.

runtime complexity, it is acceptable in an investigation scenario to run it only once for the purpose of collecting evidence. We believe that this trade-off between explainability and runtime complexity is reasonable.

Definition 3.3 (*event*). Given an event ev_n denoted by $\{T_{ev_n}, Conf_{ev_n}, \vec{V}_{ev_n}, EU_{ev_n}\}$, T_{ev_n} is the type of linguistic modality with which this event is associated, EU_{ev_n} is a set of evidence units such that $\forall eu_m^{ev_n} \in EU_{ev_n}$, and $f_{eu_m^{ev_n}}$ is of type T_{ev_n}. In addition, $\vec{V}_{ev_n} \in \mathbb{R}^N$ is a numeric vector of size N that describes to what extent this event ev_n supports each predefined hypothesis, and $Conf_{ev_n} \in [0, 1]$ is a numeric value that indicates the confidence that this event will arrive at its conclusion based on the present classification context.

We define *event* as a set of evidence units of same linguistic modality and other associated properties (see Definition 3.3). Based on the selected linguistic feature scheme, the extraction procedure is shown in Algorithm 1. The input includes the number of candidates in C, linguistic modality type $Type$, and the anonymous snippet ω. In line 2, all features of given linguistic type are extracted from the anonymous snippet ω. Based on our selected features, all n-grams of given length 1 to 8 are thereby extracted and then assigned to the evidence units (see line 5).

For each linguistic modality, we construct an event by using Algorithm 1. After event constructions, all events will be passed into the next process, as shown in Figure 2. In our case, three events are created: a lexical event (1- to 8-word n-grams), a character event (1- to 8-character n-grams), and a syntactic event (1- to 8-POS n-grams).

Example 3.4. Considering a sample text 'it is,' to construct the lexical event ev_1 in our case, first all n-grams are extracted as evidence units: $f_{eu_1^{ev_1}}$ = 'it,' $f_{eu_2^{ev_1}}$ = 'it is,' and $f_{eu_3^{ev_1}}$ = 'is.' Assuming that we have two candidate authors, thus $C = \{C_1, C_2\}$, after applying Algorithm 1, the feature vectors for these evidence units are $\vec{v}_{eu_1^{ev_1}} = (0, 0)$, $\vec{v}_{eu_2^{ev_1}} = (0, 0)$, and $\vec{v}_{eu_3^{ev_1}} = (0, 0)$.

3.2. Analysis of Individual Event

The second phase in EEDI process flow is to analyze each event independently. The goal in this phase is to isolate each event and access the impact of each event on the overall investigation problem individually [Bosworth et al. 2012]. Correspondingly, during this phase in our algorithm, we are going to independently assess each event with respect to its contribution in the overall author identification problem. For each event, two analyses are conducted:

—*Scoring*: Used to score each hypothesis (i.e., to score each candidate author) based on the given event's feature set, and determine which hypothesis is more plausible to be the correct one.
—*Consistency analysis*: Used to evaluate the feature set of a given event regarding its capability of distinguishing the writing styles among different candidates based on all known samples M.

The first analysis adopts the similarity-based approach to score each hypothesis, which is shown in Algorithm 2. To begin with, by using a $tf-idf$ scoring scheme and regarding all extracted n-grams from an event as an unified feature vector, $N+1$ numeric vectors are constructed: one numeric vector $\vec{a}$ for anonymous snippet (line 2) and N candidate author numeric vectors ($\vec{c}$ in line 7).

Although there exist other scoring functions that may achieve higher identification accuracy [Martineau et al. 2009; Lambers and Veenman 2009], we use the $tf-idf$ scheme [Zobel and Moffat 1998] for its simplicity. As in Equations (1) and

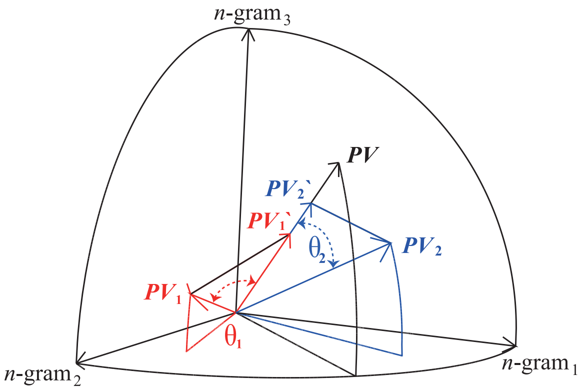

Fig. 3. A sample 3 n-gram space. Each n-gram represents one dimension.

(2), the tf score captures the normalized frequency of a given n-gram, and the idf score gives weight to each n-gram by considering its discriminant power. Variable $|AuthorsEverUsed(gram)|$ represents the number of candidate authors who ever used n-gram $gram$ in their writing samples. The constant b is used to avoid the divide-by-zero problem, and it is typically chosen as 1. We set b as 0.1, and in this way it is in a smaller order of magnitude when compared to $|AuthorsEverUsed(gram)|$. For the anonymous text, we only consider the tf score. Other scoring schemes could be employed by considering them as separate events, which could be explored in future studies.

$$tf(gram, M_i) = \frac{frequency(gram, M_i)}{maxGramFrequency(M_i)} \tag{1}$$

$$idf(gram) = \log\left(\frac{N}{b + |AuthorsEverUsed(gram)|}\right) \tag{2}$$

After the construction of aforementioned $N + 1$ numeric vectors, a final score is derived for each hypothesis (candidate) by comparing the distance between each candidate vector $\vec{c}$ and the vector for anonymous snippet $\vec{a}$. Here we adopt the *dotproduct* distance to derive this score, as shown in line 12 of Algorithm 2.

$$\begin{aligned} similarity(\vec{P_i}, \vec{P_\omega}) &= proj_{\vec{P_\omega}} \vec{P_i} \times \|\vec{P_\omega}\| \\ &= \|\vec{P_i}\| \times cos(\Theta_i) \times \|\vec{P_\omega}\| \\ &= \|\vec{P_i}\| \times \frac{\vec{P_i} \cdot \vec{P_\omega}}{\|\vec{P_i}\| \times \|\vec{P_\omega}\|} \times \|\vec{P_\omega}\| \ = \vec{P_i} \cdot \vec{P_\omega} \end{aligned} \tag{3}$$

Considering a sample 3 n-gram space in Figure 3, $\vec{PV_1}$, $\vec{PV_2}$, and $\vec{PV}$, respectively, are the style vectors of $candidate_1$, $candidate_2$, and the anonymous snippet ω. In previous work, such as Koppel et al. [2011], where n-gram–related features are employed, the *cosine* distance [Salton and Buckley 1988] is generally used to measure the distance between vectors. It only considers the included angles between vectors: the difference between Θ_1 and Θ_2 in the example. However, the difference in writing style is reflected in both n-gram coverage and normalized frequency of n-gram usage. Regarding the direction of $\vec{PV}$ as the anonymous snippet's writing style, we take the projection $\vec{PV_1'}$ of $\vec{PV_1}$ on $\vec{PV}$ and the projection $\vec{PV_2'}$ of $\vec{PV_2}$ on $\vec{PV}$ for comparison. The projection models the amount of demonstrated evidence from a given vector and shows the strength of support

Table II. Features for Confidence Estimation (Identification Context)

$score_{avg}$	Average score in scoring vector ($\vec{V}_{ev}$)
$score_{max}$	Maximum score in scoring vector ($\vec{V}_{ev}$)
$score_{min}$	Minimum score in scoring vector ($\vec{V}_{ev}$)
$dist_{max-runnerup}$	Gap statistic between max and the runner-up
$test_{length}$	Number of tokens in testing (anonymous) document ω
$tokens_{common}$	Number of shared tokens between M and ω

of the vector in this direction. The distance function is shown in Equation (3), and for the ease of computation, we multiply the norms of the anonymous vector, which is independent to the values of other vectors, and finally derive the *dotproduct* distance function.

ALGORITHM 2: Event-Based Scoring (ES)

```
Input event $ev$, writing samples $M$, anonymous snippet $\omega$
Output scoring vector: $\vec{s}$
 1: $\vec{s} \in \mathbb{R}^N$, $\vec{s} \leftarrow \{0\}$          ▷ create a numeric vector of size N
 2: $\vec{a} \in \mathbb{R}^{|EU_{ev}|}$, $\vec{a} \leftarrow \{0\}$
 3: for $m = 1$ to $|EU_{ev}|$ do
 4:     $\vec{a}[m] = \text{tf}(f_{eu_m^{ev}}, \omega)$          ▷ this vector is for anonymous snippet $\omega$
 5: end for
 6: for $i = 1$ to $N$ do
 7:     $\vec{c} \in \mathbb{R}^{|EU_{ev}|}$, $\vec{c} \leftarrow \{0\}$          ▷ this vector is for candidate author $i$
 8:     for $m = 1$ to $|EU_{ev}|$ do
 9:         $\vec{c}[m] = \text{tf}(f_{eu_m^{ev}}, M_i) \times \text{idf}(f_{eu_m^{ev}})$          ▷ here feature $f_{eu_m^{ev}}$ is an $n$-gram
10:         $\vec{v}_{eu_m^{ev}}[i] \leftarrow \vec{c}[m] \times \vec{a}[m]$          ▷ global values that will be used for evidence unit
                                                                                    visualization
11:     end for
12:     $\vec{s}[i] = \vec{a} \cdot \vec{c}$
13: end for
14: return $\vec{s}$
```

At the end of the first analysis (see line 10 of Algorithm 2), each evidence unit's scoring vector $\vec{v}$ is updated with the corresponding score v_i that describes the correlation between candidate i and this given linguistic feature. This updated value will be used in the visualization process elaborated in Section 3.5.

Algorithm 3 shows the second analysis. As defined in Definition 3.3, each event is represented as a set of linguistic features. The goal of this analysis is to evaluate features of a given event with respect to their demonstrated consistency and discriminant power among the known author writing samples M. Such properties vary for different linguistic modalities under the given *identification context* (anonymous snippet length, size of known author writing, and number of candidates, etc.). Hence, we treat each event as a stand-alone similarity-based classifier. Then a confidence value is estimated for each event in an isolated manner by building linear models. The features used to model an identification context are listed in Table II. In this way, an event is the minimum confidence estimation unit.

To proceed with this analysis, a 10-fold cross-validation test is conducted by partitioning all available writing samples from M into 10 groups of roughly equal size (line 1 of Algorithm 3). Of these 10 groups, 1 group is selected as the test set, then the remaining 9 groups are used to build events following Algorithm 1 and to predict the author of samples from the test set by using Algorithm 2 (lines 7 and 8 of Algorithm 3).

ALGORITHM 3: Event-Based Identification (EI)

```
Input known author writing samples $M$, candidate set $C$, event $ev$, anonymous snippet $\omega$
Output event $ev$
 1: $folds \leftarrow$ split($M$)        ▷ split $M$ into 10 folds for cross-validation; each fold includes nine
                                          training groups and one testing group
 2: $samples \leftarrow \emptyset$;       ▷ create an empty set of samples; each sample follows attributes in
                                          Table II
 3: for each $fold$ in $folds$ do
 4:     $foldSamples \leftarrow \emptyset$;    ▷ an empty set of samples following attributes in Table II
 5:     $correctGuess \leftarrow 0$
 6:     for each $doc$ in $TestSet_{fold}$ do
 7:         $ev' \leftarrow$ EC($N,\ T_{ev},\ doc$)                    ▷ corresponds to Algorithm 1
 8:         $scores \leftarrow$ ES($ev'$, $TrainSet_{fold}$, $doc$)       ▷ corresponds to Algorithm 2
 9:         $index$ =IndexOfMaxValue($scores$)
10:         $predictedAuthor = C[index]$
11:         if $predictedAuthor$ is ActualAuthor($doc$) then
12:             $correctGuess = correctGuess + 1$
13:         end if
14:         $sample \leftarrow$ GenerateSample($scores$, $doc$)    ▷ collect feature values (Table II)
15:         $foldSamples \leftarrow foldSamples \cup \{sample\}$
16:     end for
17:     $precision \leftarrow \frac{correctGuess}{|TestSet_{fold}|}$        ▷ calculate the precision value for this fold
18:     for each $sample$ in $foldSamples$ do
19:         $sample \leftarrow$ pad the vector $sample$ with $precision$ as target attribute.
20:     end for
21:     $samples \leftarrow samples \cup foldSamples$
22: end for
23: $Model_{ev} \leftarrow$ buildModel($samples$)    ▷ build a prediction model for this event $ev$ using
                                               precision as target attribute
24: $\vec{V}_{ev} \leftarrow$ ES($ev$, $M$, $\omega$)    ▷ collect sample from current classification context
25: $Conf_{ev} \leftarrow Model_{ev}$.predict($\vec{V}_{ev}$, $\omega$)    ▷ estimate confidence
26: return $ev$
```

The candidate with the highest score output (lines 9 and 10 of Algorithm 3) will be the predicted result. The next step is to collect vector values for the attributes listed in Table II as a sample (line 14 of Algorithm 3). After iterating documents from the test set, the precision value of the test is calculated (line 17 of Algorithm 3) and padded to each sample vector in this fold as values of target attribute (lines 18 through 20 of Algorithm 3). This process is repeated 10 times, and each group is used as the test set exactly once. Based on the collected samples, a linear model is built for each event (line 23 of Algorithm 3).

In line 24, the event derives a scoring vector for given candidates based on the anonymous snippet ω by using Algorithm 2. Based on this scoring vector, a sample is created following attributes in Table II, and then it is fed into the built model to derive the predicted precision value, which will be used as the confidence value (line 25 of Algorithm 3).

Regarding the employed attributes to model the identification context, in addition to using the "gap statistic" that describes the gap between the max score and the runner-up in [Narayanan et al. 2012; Koppel et al. 2006, 2011], we also include more attributes that describe the scoring distribution, including the maximum, the minimum, the average, and the length of testing document. Our experiment in Section 4.4 shows that all of these attributes are significantly important for confidence estimation. However, we do not include the size of known author writings, because when we conduct the

10-fold cross-validation process (lines 3 through 22 of Algorithm 3), the intercept value in the built linear model already reflects its effect as baseline.

ALGORITHM 4: Confidence-Based Normalization (CN)

```
Input event $ev$, anonymous snippet $\omega$
Output event $ev$
1: for $i$=1 to $N$ do
2:     $\vec{V}_{ev}[i]$=$\vec{V}_{ev}[i] \times Conf_{ev}$          ▷ normalize score for this event
3: end for
4: for $m = 1$ to $|EU_{eu}|$ do
5:     for $i = 1$ to $N$ do
6:         $\vec{eu}^{ev}_{m}[i] = \vec{eu}^{ev}_{m}[i] \times Conf_{ev}$          ▷ normalize the score inside each evidence unit
7:     end for
8: end for
9: return $ev$
```

3.3. Event Normalization

The event normalization process under the EEDI framework is to normalize all evidentiary data of the same type from different sources into the same measurement level and to further consider the possibility of combining them [Bosworth et al. 2012]. For example, different events from different sources may have varying timing formats or different time zone settings; to chain them together, these formats must be normalized.

Accordingly, in our approach, after the previous process each event now has a scoring vector but different confidence values, which means they have different performance levels on discriminating candidates. Before considering the combination of evidentiary data from these events, normalization of performance for each event must be done. Hence, we conduct our normalization step by multiplying the scoring vector with corresponding confidence value for each event (line 2 of Algorithm 4). Correspondingly, we update the numeric vectors stored inside all evidence units of each event by multiplying the original score with the confidence value (lines 4 through 8 of Algorithm 4). After normalization, all events are passed into the next process.

3.4. Secondary-Level Correlation

Under the EEDI framework, this process examines the correlation between events and considers ways of combining the evidence into an evidentiary chain [Bosworth et al. 2012]. In our case, accordingly, all events from the previous process are combined to derive a unidimensional score for each candidate author. The idea is to summarize the fine-grained evidence of different linguistic modalities into a single kind of evidence: the linguistic evidence.

The procedure for event combination is shown in Algorithm 5. Considering that in the previous process all events have been normalized into the same identification performance level, the final scoring vector is simply the sum of the scoring vector from each input event. In this algorithm, lines 1 through 8 combine scoring vectors from all input events, and line 9 determines the prediction result as the candidate author that achieves the highest score.

$$
\begin{aligned}
p &= \max_{ev_n}^{EV} P(\mathit{predicted\ author} \mid ev_n) \\
&= \max_{ev_n}^{EV} \begin{cases} Conf_{ev_n}, & \text{if } ev_n \text{ agrees on final } \mathit{predicted\ author} \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
\tag{4}
$$

To combine multiple confidence values of different classifiers, typical approaches include Product Rule, Max Rule, Min Rule, and Majority Vote Rule [Kittler et al. 1998].

ALGORITHM 5: Event Combination (EC)

```
Input writing samples $M$, candidate set $C$, set of event $EV$, anonymous snippet $\omega$
Output $author$, confidence value $p$
 1: $\vec{fs} \in \mathbb{R}^N$, $\vec{fs} \leftarrow \{0\}$          ▷ initialize final scoring vector with 0
 2: conf $\in \mathbb{R}^{|EV|}$, conf $\leftarrow \{0\}$          ▷ a vector of confidence values
 3: for $n = 1$ to $|EV|$ do
 4:     for $i = 1$ to $N$ do
 5:         $\vec{fs}[i] = \vec{fs}[i] + \vec{V}_{ev_n}[i]$
 6:     end for
 7:     conf$[n] = Conf_{ev_n}$
 8: end for
 9: $prediction \leftarrow$ IndexOfMaxValue($\vec{fs}$)          ▷ determine the prediction result
10: $author \leftarrow C[prediction]$
11: agreedConf $\in \mathbb{R}^{|EV|}$, conf $\leftarrow \{0\}$
12: for $n = 1$ to $|EV|$ do
13:     if $ev_n$ agrees $prediction$ then
14:         agreedConf$[n]$ = conf$[n]$
15:     else
16:         agreedConf$[n] = -1$
17:     end if
18: end for
19: $p$ = max(agreedConf)          ▷ estimate the final confidence value
20: return $author$, $p$
```

Here we combine the Max Rule and Majority Vote Rule to derive our final estimated confidence value. As lines 12 through 19 of Algorithm 5 show, the final confidence value is determined as the maximum estimated confidence value among all events that agree on the final output candidate (see Equation (4), in which *predicted author* indicates the final output prediction from line 9 Algorithm 5).

Previous research [Koppel et al. 2011; Narayanan et al. 2012] mostly combine classifiers using the ensemble method and derive the final result in a voting manner. Differently from these, we combine classifiers—or rather events, in our case—in the scoring vector level, and each scoring vector is normalized by the estimated confidence (see Equation (5), in which $\vec{fs}[k]$ is the final score for candidate k as used in line 1 Algorithm 5, and $\vec{V}_{ev_n}[k]$ is the final score for candidate k in the scoring vector of event ev_n as defined in Definition 3.3). Our experiment demonstrates that this approach can achieve higher accuracy.

$$\vec{fs}[k] = \sum_{ev_n}^{EV} \vec{V}_{ev_n}[k] \times Conf_{ev_n} \tag{5}$$

3.5. Chain of Evidence Construction

In this process, under the EEDI framework, evidences are aligned on a timeline, and based on this timeline, a coherent chain of evidence is developed [Bosworth et al. 2012]. This chain of evidence is able to connect the starting point and ending point of the criminal incident. However, in our solution, temporal priority among all linguistic evidence is nonexistent. Based on the employed *dot-point* distance, the cumulative effect of evidences is instead established from hypotheses to conclusion.

$$\vec{fs}[k] = \sum_{ev_n}^{EV} \sum_{eu_m^{ev_n}}^{EU_{ev_n}} \vec{v}_{eu_m^{ev_n}}[k] \tag{6}$$

At this point, based on the input events, the cumulative effect to derive the final unidimensional score for each hypothesis can be expressed as Equation (6) by employing the intermediate results stored in evidence units according to Algorithms 2 and 4. $\vec{fs}[k]$ refers to the final score for candidate k in line 1 of Algorithm 5, which is the same variable in Equation (5) but is calculated using different intermediate results.

The task of this process is to visualize all evidence units with respect to their proximity to each hypothesis. The visually cumulative effect of all evidence units should be able to reflect the difference between candidate scores $\vec{fs}[k]$. Formally, a visual measurement function vf should have the following property.

Property 1 (*proportionally visualizable*). Given a set of hypotheses H, we say that they are proportionally visualizable over a visual effect function vf if they satisfy $\forall H_k \in H \; vf(H_k) \propto \vec{fs}[k]$.

To begin with, hypotheses are visualized. As defined in Definition 3.1, the hypothesis is the statement that an anonymous snippet ω is authored by one specific author. Given N candidates in C, we thus have N hypotheses, and each hypothesis is represented by the raw tokens extracted from the anonymous snippet ω with the corresponding statement about one specific candidate.

As shown in Figure 4, two hypotheses are presented as examples . Each hypothesis is represented by the hypothetical statement on the title along with the following evidence extracted from anonymous snippet ω: the first row represents character-level tokens, the second row represents word-level tokens, and the third row represents Part-of-Speech tokens. To make the representation simpler and clearer, in the first row we display the character tokens with a transparent font color so that each character token can be easily matched to the lexical token beneath.

After presenting the visualizations of hypotheses, we are going to visualize all evidence units (defined in Definition 3.2) by coloring each evidence unit's tokens in the preceding representations of hypotheses. The color is determined by how affiliated an evidence unit is to the given hypothesis. An evidence unit hereby is our smallest visualization unit.

To color the tokens, the HSL color scheme is employed. The HSL scheme encodes color by using three parameters: Hue, Saturation, and Lightness. Hue represents the selected tint ranging from 0 to 360, and in most cases it is used as a qualitative representation in data visualization: the difference in kinds reflected in the difference of tint. Saturation controls its colorfulness (from 0 to 100), and Lightness measures how much light should be reflected from this color, ranging from 0 (appears as black) to 100 (appears as white); 50 is *normal* [Çelik et al. 2012]. Lightness is visually suitable as a quantitative/sequential data representation. *Dark equals more* is a standard cartographic convention [Harrower and Brewer 2003], and the difference of lightness can still be perceived by people with red-green color vision impairments [Harrower and Brewer 2003]. Thus, we adopt the lightness value representing the scores of evidence units.

Based on our observation, given an evidence unit $eu_m^{ev_n}$ and its scoring vector $\vec{v}_{eu_m^{ev_n}}$, in most cases the range of this vector $range(\vec{v}_{eu_m^{ev_n}})$ is only a small fraction of the overall score range. Simply picking up the lightness value of the given evidence unit $eu_m^{ev_n}$, for hypothesis k based on its score $\vec{v}_{eu_m^{ev_n}}[k]$, will naturally lead to the imperceptible visual discrimination among hypotheses. Hence, instead of visualizing the original scores, we visualize $dif(eu_m^{ev_n}, k)$ in Equation (7), which represents how the original score differs from the minimum score in that scoring vector. The constant α is used to shift the range, avoiding assigning a blank background on $eu_m^{ev_n}$ for hypotheses k when $\vec{v}_{eu_m^{ev_n}}[k]$ equals $min(\vec{v}_{eu_m^{ev_n}})$, to visually distinguish the absence of an n-gram (blank background)

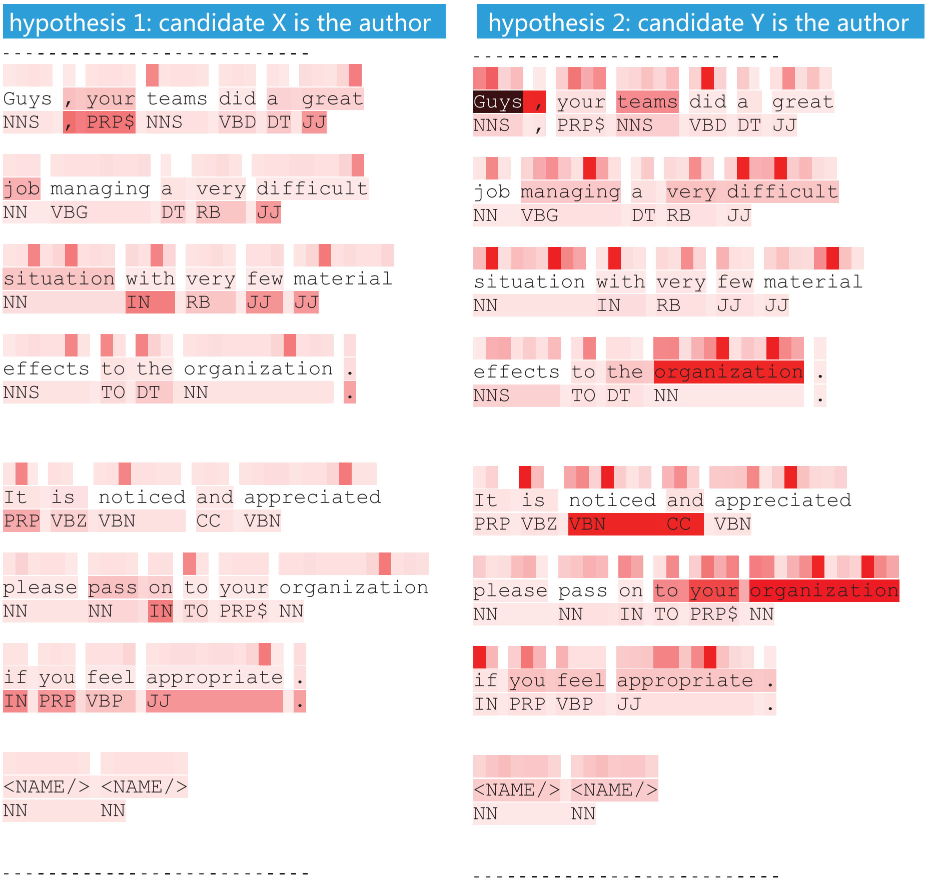
hypothesis 1: candidate X is the author
Guys , your teams did a great
NNS , PRP$ NNS VBD DT JJ
job managing a very difficult
NN VBG DT RB JJ
situation with very few material
NN IN RB JJ JJ
effects to the organization .
NNS TO DT NN .
It is noticed and appreciated
PRP VBZ VBN CC VBN
please pass on to your organization
NN NN IN TO PRP$ NN
if you feel appropriate .
IN PRP VBP JJ .
<NAME/> <NAME/>
NN NN
hypothesis 2: candidate Y is the author
Guys , your teams did a great
NNS , PRP$ NNS VBD DT JJ
job managing a very difficult
NN VBG DT RB JJ
situation with very few material
NN IN RB JJ JJ
effects to the organization .
NNS TO DT NN .
It is noticed and appreciated
PRP VBZ VBN CC VBN
please pass on to your organization
NN NN IN TO PRP$ NN
if you feel appropriate .
IN PRP VBP JJ .
<NAME/> <NAME/>
NN NN

Fig. 4. Evidentiary chain visualization: hypothesis representations and the visualized evidence units.

and the presence of an n-gram, but for a given hypothesis, it is the minimum score in the scoring vector.

To calculate the value $dif(eu_m^{ev_n}, k)$ for each hypothesis k on each evidence unit $eu_m^{ev_n}$, the global range $maxR$ of the scaled difference is first calculated by using the first three equations in Equation (7). The range of the scaled difference in scoring vectors is calculated for each event and then all ranges are combined to reach $maxR$ (globally maximum scaled difference in all scoring vectors).

$$\begin{aligned}
range'(eu_m) &= max(\vec{v}_{eu_m}) - min(\vec{v}_{eu_m}) \\
maxR_{ev_n} &= max\left(\left\{eu_m^{ev_n} \in EU_{ev_n} \mid range'\left(eu_m^{ev_n}\right)\right\}\right) \\
maxR &= max(\{ev_n \in EV \mid maxR_{ev_n}\}) + \alpha \\
dif\left(eu_m^{ev_n}, k\right) &= \frac{\vec{v}_{eu_m^{ev_n}}[k] + \alpha - min(\vec{v}_{eu_m})}{maxR}
\end{aligned} \tag{7}$$

Example 3.5. Consider a sample text ‘*your organization*’ and two candidates C_1 and C_2. Assume that we only employ the lexical event $EV = \{ev_1\}$, so we have three lexical n-grams: $eu_1^{ev_1}$ ‘*your*,’ $eu_2^{ev_1}$ ‘*organization*,’ and $eu_3^{ev_1}$ ‘*your organization*.’ In addition, we assume that their corresponding scoring vectors are $\vec{v}_{eu_1^{ev_1}} = (0.3, 0.6)$, $\vec{v}_{eu_2^{ev_1}} = (0.8, 0.5)$, and $\vec{v}_{eu_3^{ev_1}} = (0.1, 0.2)$. In this case, $range'(eu_1^{ev_1}) = 0.3$, $range'(eu_2^{ev_1}) = 0.3$, $range'(eu_3^{ev_1}) = 0.1$, and $maxR_{ev_1} = 0.3$. By setting $\alpha = 0.1$, since we only use one event, we have $maxR = 0.4$. Applying the fourth function in Equation (7) on each evidence unit's scoring vectors, we then have $dif(eu_1^{ev_1}, 1) = 0.25$, $dif(eu_1^{ev_1}, 2) = 1$, $dif(eu_2^{ev_1}, 1) = 1$, $dif(eu_2^{ev_1}, 2) = 0.25$, $dif(eu_3^{ev_1}, 1) = 0.25$, and $dif(eu_3^{ev_1}, 2) = 0.5$.

The linguistic feature we chose is based on the *n*-gram model, where each evidence unit is represented as a sequence of tokens. As such, different evidence units may share the same token in the hypothesis representation. Accordingly, each evidence unit is colored in an overlapping manner.

$$L_{token_n}^{H_k}\left(eu_m^{ev_n}\right) = \begin{cases} L_{token_n}^{H_k} - \eta \times dif\left(eu_m^{ev_n}, k\right), & \text{if } eu_m^{ev_n} \text{ stem from } token_n \\ L_{token_n}^{H_k} & \text{otherwise} \end{cases} \tag{8}$$

Given a visual representation of hypothesis H_k, we start by initializing all tokens' backgrounds with a maximum lightness value (i.e., the background color reflects 100% light and appears to be blank), and then we enumerate tokens in the hypotheses representation to apply Equation (8). Given a $token_n$ in H_k, for each previously extracted evidence unit $eu_m^{ev_n}$, if $f_{eu_m^{ev_n}}$ stems from $token_n$ then the token's lightness value degrades by the multiplication of degradation factor η and its normalized variant score $dif(eu_m^{ev_n}, k)$. Degradation factor $\eta \in (0, 100]$ controls the contrast between hypotheses and can be designated by the user or empirically as $100.0/MaxMatch$, where $MaxMatch$ indicates the maximum number of evidence units that can stem from the same token. $eu_m^{ev_n}$ stems from $token_n$, which means that the evidence unit $eu_m^{ev_n}$ partially or completely originates from the $token_n$. For example, the evidence unit '*your organization*' can stem from the token '*your*' in the phrase '*to your organization*' but not from the token '*your*' in the phrase '*your teams.*'

Since this "stem" mapping between tokens and evidence units is identical for all hypotheses, given the same evidence unit, the lightness value of a token is inversely proportional to the score $dif(eu_m^{ev_n}, k)$ of the hypothesis. In this way, it is also inversely proportional to the original score $\vec{v}_{eu_m^{ev_n}}[k]$ (Equation (9)).

Example 3.6. Continue Example 3.5. For the word '*your*,' there are two *n*-grams, $eu_1^{ev_1}$ '*your*' and $eu_3^{ev_1}$ '*your organization*' that stem from this word. By setting $\eta = 30$, the lightness of the word '*your*' for candidate C_1 (i.e., hypothesis H_1), $L_{yours}^{H_1} = 100 - 30 \times dif(eu_1^{ev_1}, 1) - 30 \times dif(eu_3^{ev_1}, 1) = 100 - 30 \times 0.25 - 30 \times 0.25 = 85$. Correspondingly, for candidate 2 (i.e., hypothesis 2), $L_{yours}^{H_1} = 100 - 30 \times 1 - 30 \times 0.5 = 55$.

$$\begin{aligned} L_{token_n}^{H_k}\left(eu_m^{ev_n}\right) &\propto dif\left(eu_m^{ev_n}, k\right)^{-1} \\ &\propto (\vec{v}_{eu_m^{ev_n}}[k] + \alpha - min(\vec{v}_{eu_m^{ev_n}}))^{-1} \ \propto \vec{v}_{eu_m^{ev_n}}[k] \end{aligned} \tag{9}$$

Our selected visual function $vf_{VEA}(H_k)$ for hypothesis k is the global darkness of its visual representation, denoted by $GD(H_k)$, which is inversely proportional to the global lightness $GL(H_k)$ function. We assume that the global lightness value is contributed by the cumulative lightness of all tokens on the representation. This assumption is reasonable when the anonymous snippet is short. $GD(H_k)$ is formulated in Equation (10).

$$vf_{VEA}(H_k) = GD(H_k) \propto GL(H_k)^{-1} \tag{10}$$

It can be shown that this visual function satisfies Property 1 as follows. First, the global lightness function $GL(H_k)$ for hypothesis k is formulated as the cumulative

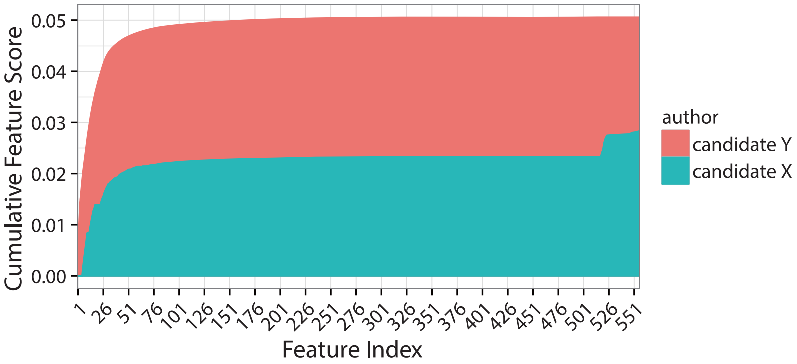

Fig. 5. Cumulative evidence unit scoring diagram: the serial that achieves the highest score at the end of x-axis is for the most plausible candidate.

lightness of all tokens (see step 1 in Equation (11)). By combining Equation (9), the $GL(H_k)$ function is inversely proportional to the final score of hypothesis k (see steps 2 through 5 in Equation (11)). $\vec{fs}[k]$ refers to the same variable in Equations (5) and (6).

$$\begin{aligned} GL(H_k) &= \sum_{token_n}^{tokens(H_k)} \sum_{ev_n}^{EV} \sum_{eu_m^{ev_n}}^{EU_{ev_n}} L_{token_n}^{H_k}\left(eu_m^{ev_n}\right) \\ &\propto \left(\sum_{token_n}^{tokens(H_k)} \sum_{ev_n}^{EV} \sum_{eu_m^{ev_n}}^{EU_{ev_n}} dif\left(eu_m^{ev_n}, k\right) \right)^{-1} \\ &\propto \left(\sum_{token_n}^{tokens(H_k)} \sum_{ev_n}^{EV} \sum_{eu_m^{ev_n}}^{EU_{ev_n}} \vec{v}_{eu_m^{ev_n}}[k] \right)^{-1} \\ &\propto \left(\sum_{ev_n}^{EV} \sum_{eu_m^{ev_n}}^{EU_{ev_n}} \vec{v}_{eu_m^{ev_n}}[k] \right)^{-1} \propto \left(\vec{fs}[k] \right)^{-1} \end{aligned} \tag{11}$$

In this way, by combining Equation (10), the visual function $GD(H_k)$ is proportional to the final score of hypothesis k (Equation (12)). Thus, our selected presentation of hypothesis and evidence unit satisfies Property 1 over visual function $GD(H_k)$, which indicates that the darker is the hypothesis representation's holistic color, the higher final score that this hypothesis possesses.

$$vf_{VEA}(H_k) = GD(H_k) \propto GL(H_k)^{-1} \propto \vec{fs}[k] \tag{12}$$

After all of the aforementioned coloring is done, one can conclude that the hypothesis with the most holistically darkest coloring representation is the most plausible one. As the example in Figure 4 demonstrates, representation of *hypothesis 2* is more holistically darker than that of *hypothesis 1*, and thus the corresponding candidate, *candidate Y*, is the plausible author.

In addition, we construct an evidence unit cumulative scoring diagram, as shown in Figure 5. An area with a different color represents a different hypothesis, and the one that achieves the highest score at the end of x-axis is the most plausible one. If many candidates are involved, or the given anonymous text is too long, the cumulative visual discrimination will be difficult to perceive in Figure 4, although this scoring diagram

Table III. Confidence Estimation

Events	Estimated Confidence
n-gram (lexical level)	0.8311
n-gram (character level)	0.9560
n-gram (syntactic level)	0.6867
Voted maximum	0.9560

is still able to show which hypothesis achieves the highest final score, and the detailed evidence can still be referred to the visualized evidence.

At the end of this phase, we listed all estimated confidence values in Table III. In this example, since all three events agreed on same plausible hypothesis, the overall confidence value is simply the maximum: one.

3.6. Corroboration

Note that linguistic evidence is only one kind of event; other nonlinguistic evidence exists related to the criminal incident and may support the authorship identification problem. Evidence may include system logs, network logs, or IP-related information from an ISP, or even the socioeconomic relationship between each candidate and this incident. By including this process, linguistic evidence for this authorship attribution problem becomes a stand-alone event, and investigators can further connect the linguistic and nonlinguistic events to corroborate their final hypothesis on the incident.

4. EXPERIMENTAL RESULTS

The objective of the experiment is to evaluate our approach with respect to the identification accuracy and robustness under varying circumstance in the authorship attribution problem. We adopted the Enron email dataset, which was made public by the Federal Energy Regulatory Commission [Shetty and Adibi 2004]. This dataset contains 517,424 emails from 151 users. Email data tend to be relatively short compared to other literature works and bring more challenges to the authorship identification problem.

As previous work demonstrated, the identification context (the available samples and available hypotheses/candidates, etc.) of the authorship attribution problem strongly affects the solution's performance, whereas most of the previous experiments by design failed to test their model systematically. To avoid other possible explanations of our experimental results, we first conducted statistical analysis of the dataset and then conducted both controlled sampling experiments and stratified randomized experiments.

4.1. Dataset Preprocessing, Analysis, and Experimental Setups

We started by conducting preprocessing procedures on this dataset. The first procedure extracted the body from each email, and the second procedure cleaned up the identity-related information. The extraction procedure was completed by using a set of regular expressions that removed the "forward" and "reply" part of the email as well as all header information. Removing the identity-related information is relatively more complex. We completed this procedure by employing the following steps:

- —We utilized the regular expressions to replace URL links with the <*link*/> tag.
- —We utilized the Name Finder in the OpenNLP[5] project to replace all of the found name entries with the <*name*/> tag.
- —We fetched the employee information from the dataset and generated a list of first names and a list of last names. We replaced all tokens that were exactly the same, case ignored, as the names in these two lists with the <*name*/> tag.

[5] Available at http://opennlp.apache.org.

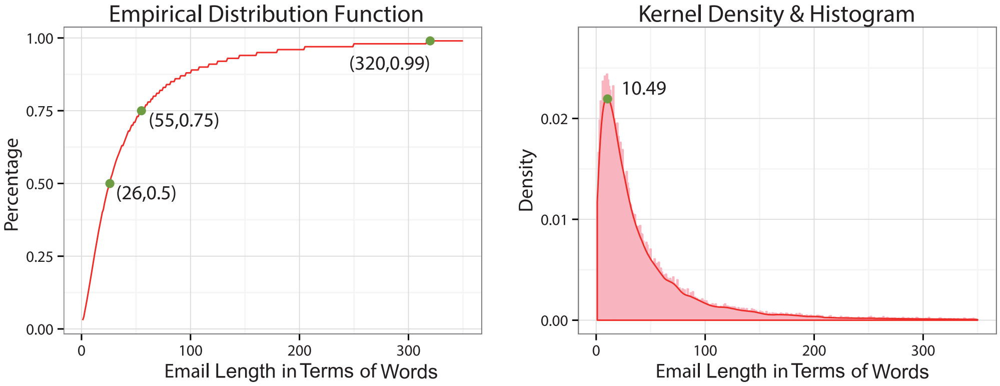

Fig. 6. Dataset analysis.

—Based on the preceding name lists, we found all tokens that had exactly one string editing distance [Levenshtein 1966] to the names, case ignored, and replaced these tokens with the <*name*/> tag. We assume that the author of a given email can only make one character mistake when typing his or another's first/last name.
—Also based on the employee information, we constructed a list of short names by concatenating the first character of a first name and that of the last name. We found these tokens and replaced them with the <*name*/> tag in the last sentence for each email.

After preprocessing, we analyzed the distribution of email length for this dataset. As plotted in Figure 6, we conducted the empirical distribution analysis, the kernel density analysis, and the histogram analysis. These diagrams show that most of the emails inside this dataset are of length less than 11 tokens. According to the criteria concluded in Burrows [2007], at least 1,000 emails per author are required to guarantee a good identification result. This introduces a great challenge to authorship identification solutions when it comes to a context with a small number of writing samples. For the length distribution, emails of length ranging from 1 to 26 tokens comprise 50% of the total, emails of length ranging from 1 to 55 tokens comprise 75% of the total, and 99% of the total are emails of length ranging from 1 to 320 tokens.

To systematically test our approach, we designed two experiments: a controlled experiment and a stratified randomized sampling experiment. The first experiment was done to evaluate the performance of our approach under different authorship attribution contexts and to evaluate its performance degradation as the available information systematically degrades. The second experiment was performed to simulate the real authorship identification scenario, where emails of varying lengths are sampled for each candidate author and, in most cases, the size of known author writing samples is unbalanced.

The authorship attribution problem can be regarded as a multiclass text classification problem: we classify the anonymous snippet into a set of predefined classes (i.e., candidate authors) based on the known samples from each class (i.e., known author writing samples). We evaluate our approach with respect to the classification accuracy, which indicates the percentage of anonymous snippets that are correctly classified.

For all experiments described next, we adopt the 10-fold cross-validation test, where the emails for each author are split into 10 groups. For a total of 10 iterations, each is used as a validation set exactly once (used as anonymous samples), and the remaining 9 groups are used as known author samples. The final accuracy measure is the average of accuracy values of these 10 iterations.

4.2. Controlled Experiment

In this experiment, we randomly sampled documents multiple times under controlled conditions and systematically tested our approach with respect to its identification accuracy. First, based on previous work, we identified the three most critical factors that significantly affect authorship attribution performance: the size of known author writings, the size of the candidate set, and the document length. We counted the document length with respect to the number of tokens that it had. The size of known author writings is measured by the number of documents (i.e., emails). We did not break a complete email or reconstruct an email by concatenation. The following are the selected factors and their selected value intervals:

- The distribution of the email length naturally leads us to conduct experiments on three different levels: emails of length 1 to 26 tokens (50%), emails of length 27 to 55 tokens (25%), and emails of length 56 to 320 tokens (24%).
- For the size of samples for each author, we selected 20, 40, 80, and 120 emails.
- For the size of candidate set, we chose the typical values: 2, 5, 10, and 20 authors.

Since each candidate author is regarded as a class in a classification problem, it has its own accuracy value (number of samples that are identified correctly) during the 10-fold validation. In this case, because each author has the same controlled number of known writing samples, our problem can be attributed to the balanced-class classification problem. Hence, we only adopted the Macro Average [Savoy 2012] to calculate the overall accuracy value in each round. Macro Average accuracy is simply the average of all accuracy, where all classes are equally weighted.

By controlling the combination of the aforementioned conditions, we conducted three tests. The first one was conducted by isolating each event to systematically test the difference between the events with respect to their identifying accuracy. The second one was conducted by employing the complete VEA approach in Section 3 to compare its performance with other typical approaches. Since our approach of combining events (i.e., linguistic modalities) can be attributed as an ensemble method, we also compared our approach with the typical voting ensemble method.

Figure 7 shows the experimental result of the first test, in which each event is tested in an isolate manner by employing Algorithm 2. In each diagram, the *Num of Candidates* axis represents the size of candidate authors, and the *Num of Samples* axis represents the size of samples for each author in the 10-fold validation. The z-axis indicates Macro Average accuracy under the given values of x and y. Additionally, the color of the gradient surface indicates the accuracy value: the brighter the color, the higher accuracy value of the point. For all three diagrams in this figure, the upper surface is the event for lexical n-gram, which means that it achieves the best identifying accuracy across all given conditions, and the intermediate surface is the event character n-gram; on the bottom, the lowest surface is for the event Part-of-Speech n-gram.

The three diagrams in Figure 7 show that as the available information decreases in the identification context, the identification accuracy for all isolated events drops significantly. Lexical n-gram performs the best across all given conditions, but it is significantly affected by the length of the given anonymous document, whereas the POS n-gram event appears to suffer less from this condition even though it achieves at most around 80% accuracy. In addition, as the size of candidate increases, performance of the event lexical n-gram appears to drop more slowly than the other two surfaces.

This result indicates that evidence of different linguistic modalities has different degrees of sensitivity to the conditions of the given investigation scenario. Hence, for a confidence estimation task, where a confidence value is part of the identification result implying reliability of this result, a distinct model should be built for each linguistic

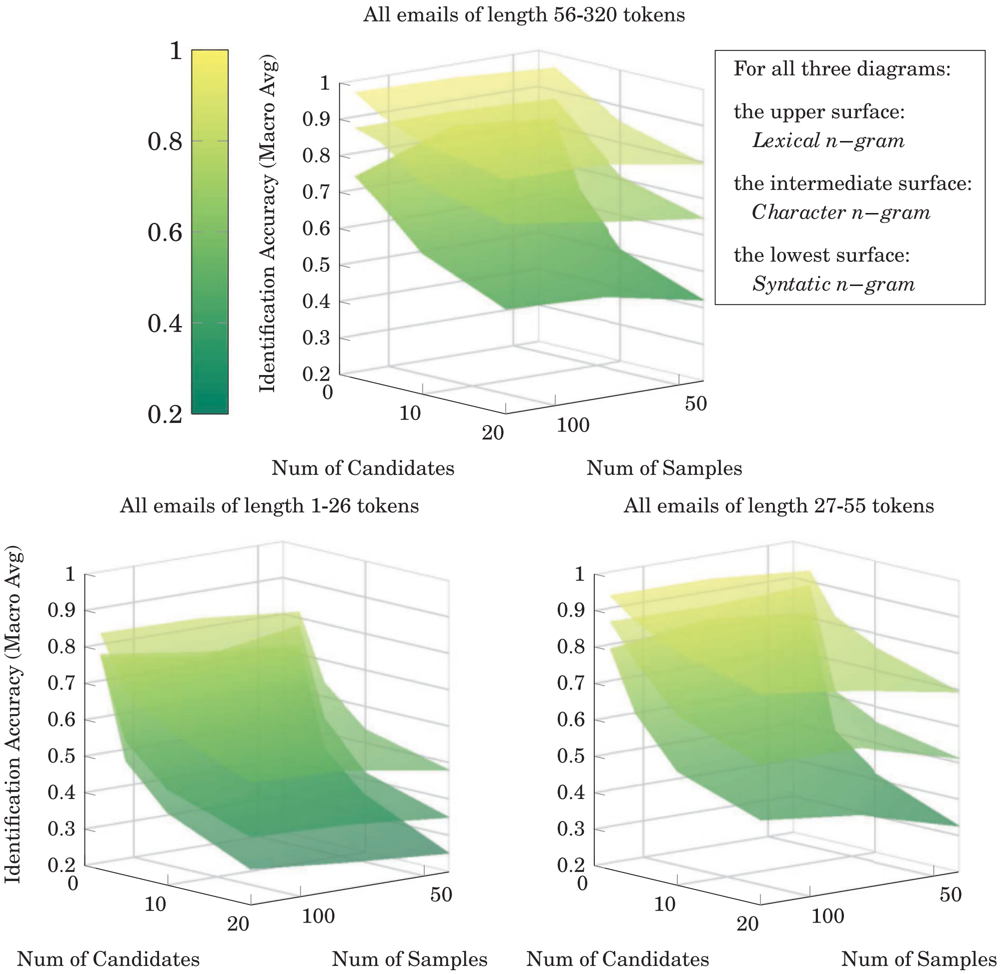

Fig. 7. Performance comparison between isolated events. For all diagrams, the upper surface is a lexical n-gram event, the intermediate surface is a character n-gram event, and the lowest surface is a POS n-gram event.

modality. As well, when combining evidence from these modalities, they should be weighted accordingly.

Figure 8 shows the experimental results of the second test. In this experiment, we compare the performance of VEA to the other two typical stylometric techniques. The selected stylometric feature set of these two approaches consists of 2,302 stylometric features, as shown in Table IV. The first 302 static features are used and discussed in Iqbal et al. [2013]. We also included the top 2,000 n-grams ranked by their occurring frequency; n ranges from 1 to 4. Previous AA research already experimentally demonstrated that frequency value carries enough stylistic information and outperforms the *information gain* scheme [Stamatatos 2009]. We also tried *information gain* for feature selection but did not notice significant difference in their performance. As we are not comparing which feature selection scheme is better, here we only show the result using frequency. Two attribution techniques were selected for comparison: SVM and J48, which demonstrated the most comparable performance in Iqbal et al. [2013]. We choose

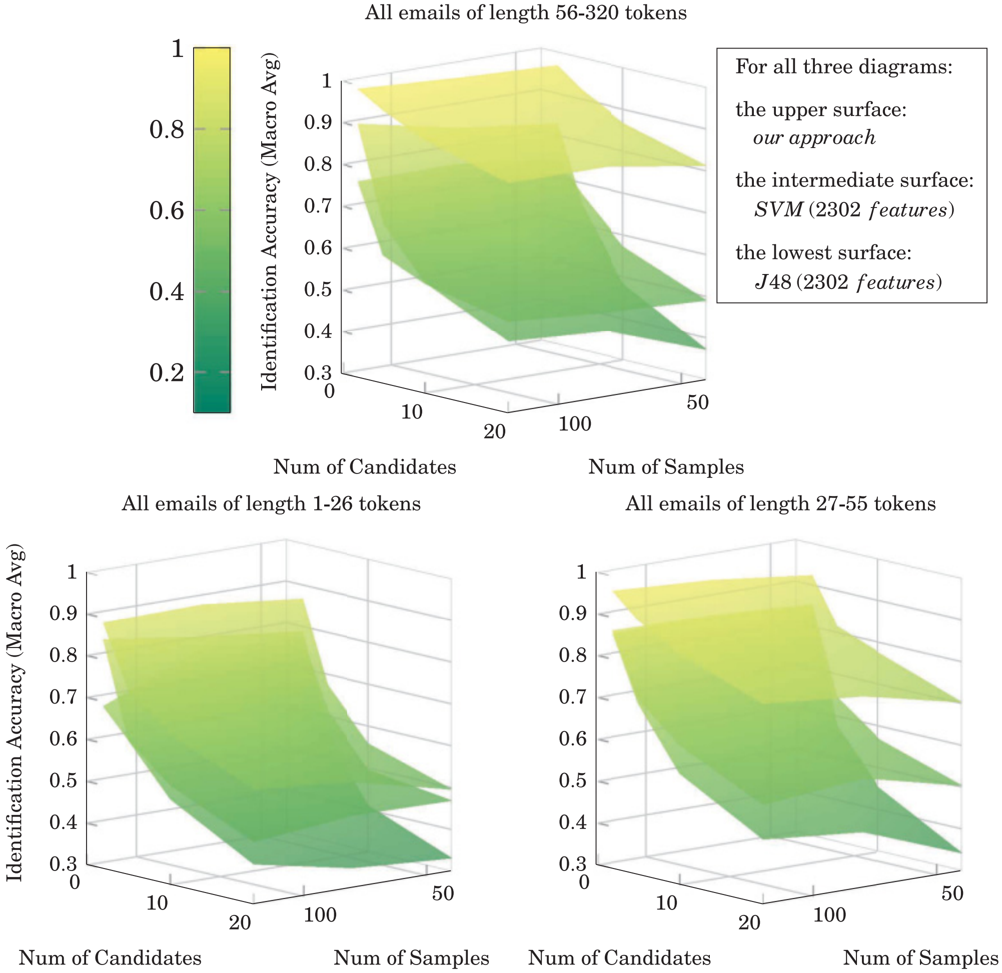

Fig. 8. Performance comparison between approaches. For all diagrams, the upper surface is VEA, the intermediate surface is the stylometric J48, and the intermediate surface is the stylometric SVM.

Table IV. Employed Features for SVM and J48 (2,302 Fatures in Total)

Feature Type	Features	Count	Example
Static	Lexical features	105	Ratio of digits and vocabulary richness, etc.
	Function words	150	Occurrence of *after*
	Punctuation marks	9	Occurrences of punctuation *!*
	Structural features	15	Presence/absence of greetings
	Domain-specific features	13	Occurrences of words *contract*, *time*, *draft*, etc.
	Gender-preferential features	10	Ratio of words ending with *ful*
Dynamic	Top 2,000 word n-grams, character n-grams, and POS n-grams ranked by the occurring frequency	2,000	'It is noticed,' 'notic,' 'tice,' 'PRP VBZ VBN,' etc.

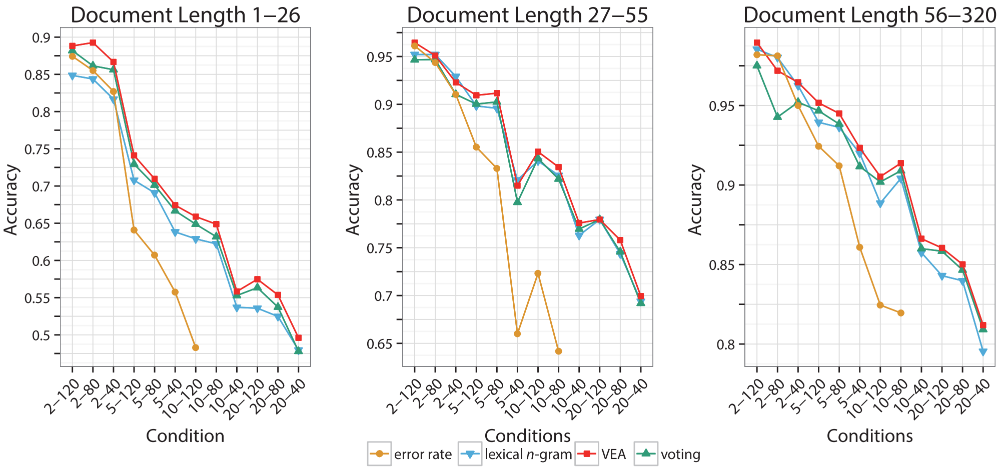

Fig. 9. Performance comparison between VEA, voting ensemble, and lexical n-gram event. The x-axis represents the combination of condition (e.g., 2–120 represents a scenario with 2 candidates and 120 emails for each of them). Part of the line for the series *error rate* is omitted, as it is below the lower bound of other series.

the libSVM [Chang and Lin 2011] for SVM implementation and the J48 decision tree *C4.5* implementation in weka.[6]

As indicated in Figure 8, which is the same diagram representation used in Figure 7, our VEA approach consistently outperforms the other two typical approaches. Even though the given anonymous document is only of length 1 to 26, it can still achieve more than 85% accuracy in a two-candidate scenario. In addition, as the diagrams show, our VEA approach is more robust against information drops with respect to the candidate size and the available known author samples.

In the third test, we compared our VEA approach with an ensemble method based on voting, an ensemble method combining events using the classification error rate, and the lexical n-gram-event-only approach. The classification error rate is collected by conducting a 10-fold cross-validation test on the known author writing samples. Thus, each of the known author writing samples is tested exactly once. We use similar equations in *AdaBoost* [Freund and Schapire 1995] to derive the weight for each event. First, the error rate is calculated using the first equation in Equation (13), and $X_{doc_i} = 1$ if the given event incorrectly classifies document doc_i; otherwise, $X_{doc_i} = 0$. Then, the weight of event ev_n is calculated using the second equation in Equation (13). If the weight is less than zero, then we treat it as zero (i.e., if the event cannot correctly classify 50% of the samples, then the weight will be zero). Finally, each classifier is combined to derive the final prediction by using weighted voting.

The experimental result is shown in Figure 9. The y-axis represents Macro Average accuracy, and the x-axis stands for the combination of conditions. For example, "2–120" stands for 2 candidate authors, each of whom has 120 writing samples. As the diagram illustrates, our VEA approach promotes the identifying accuracy and performs better than all of the others, especially when the given documents are short. It always outperforms the voting ensemble approach, and it performs better than the pure lexical n-gram approach, except in three scenarios. By using the pairwise t-test, it also turns

[6] http://www.cs.waikato.ac.nz/ml/weka/.

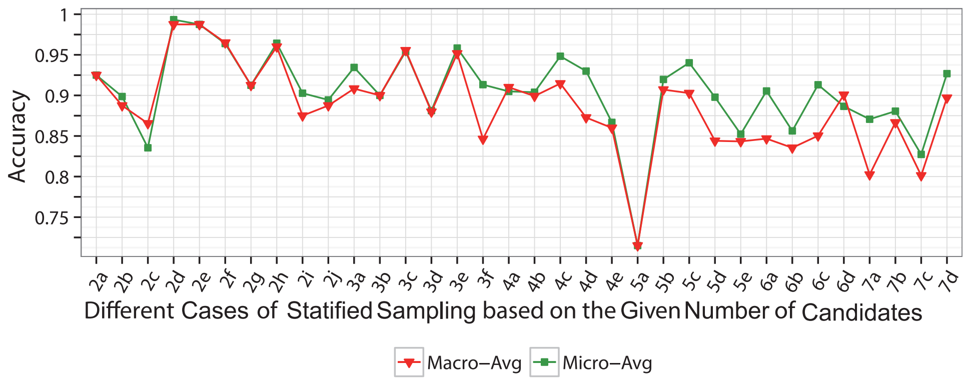

Fig. 10. Performance of VEA on unbalanced-class problem. The x-axis represents the case of sampling (e.g., 2a is a sampling with two authors, whereas 2b is another sampling with two different authors).

out that the VEA approach outperforms the others.

$$error(ev_n) = \frac{1}{10 \times |M|} \sum_{doc_i}^{M} X_{doc_i} \qquad weight_{ev_n} = log \frac{1 - error(ev_n)}{error(ev_n)} \tag{13}$$

4.3. Stratified Randomized Sampling Experiment

In this section, we describe the second experiment. To simulate the actual authorship identification task, we conducted the stratified randomized experiment, where the sample size for each author is unbalanced and the variant in document length of the samples is much larger. In this experiment, the number of emails that we randomly sampled (without replacement) for each candidate depends on how many emails this candidate actually had in the whole dataset. We also manually examine and conduct preprocessing steps for each email with respect to its identity-related information to avoid the explanation that the high accuracy is simply attributed to the capture of identity-related information rather than the writing style.

Both the Macro Average and Micro Average accuracy measures are employed in this experiment. As mentioned earlier, Macro Average is simply the average of accuracy value from each author (i.e., class in classification problem). Micro Average accuracy employs the confusion matrix to calculate the accuracy value for multiclass classification [Savoy 2012]. Typically, Micro Average will yield better results in an unbalanced classification problem because it gives more weight to the class that has more samples. For example, in a two-class classification problem, if for the first class 1 sample is correctly classified out of 10, and for the second class 19 are correctly classified out of 20, the Macro Average accuracy is simply $(1/10 + 19/20)/2 = 0.525$, but the Micro Average is $(1 + 19)/(10 + 20) = 0.667$.

The experimental result is shown in Figure 10. The labels on the x-axis indicate the given scenario. For instance, "2a" means a stratified sampling on two random authors, whereas "2b" is another stratified sampling on two random authors. The y-axis represents the accuracy value, and two serials in the diagram respectively stand for the Macro Average and the Micro Average. As shown in this diagram, our VEA approach can still handle unbalanced class problems and achieve good identifying accuracy with respect to both Macro Average and Micro Average.

Table V. Confidence Estimation Result

Variable	Coefficient	z-Value	$Pr(>\mid z \mid)$
$score_{avg}$	$1.204e+01$	7.429	$1.10e-13$
$score_{max}$	$-4.234e+00$	-5.747	$9.07e-09$
$score_{min}$	$-7.368e+00$	-4.333	$1.47e-05$
$dist_{max-runnerup}$	$2.032e+00$	4.818	$1.45e-06$
$test_{length}$	$4.775e-04$	5.004	$5.63e-07$
$tokens_{common}$	$5.811e-04$	4.378	$1.20e-05$
MAE : 0.057536618 R^2 : 0.90564199			

4.4. Confidence Estimation

In this section, we present our confidence estimation results. To verify how well our selected features can model the identification precision value, we first collected the input samples for building the estimation model from all previous runs of the VEA approach in the preceding experiments. Specifically, these samples were collected from line 23 of Algorithm 3 based on the features in Table V. These samples have been padded with the prediction precision on test set (see lines 17 through 20 of Algorithm 3). This test is to evaluate whether the features that we selected can model the output precision value. The regression modeling result is on the first six rows of Table V, which includes the estimated coefficients and the standard z-test for each coefficient. In this table, the z-values indicate that our selected features all significantly affect the target precision attribute. Note that the gap statistic $dist_{m_r}$ [Koppel et al. 2006] affects the prediction result, but the distribution-related features in scores (i.e., $socre_{avg}$, $score_{max}$) play relatively more important and stable roles.

$$MAE = \frac{1}{|AllTest|} \sum_{i}^{|AllTest|} |EstimatedConfidence_i - ClassificationPrecision_i| \qquad (14)$$

To verify whether our estimation model can actually predict the accuracy value of the unseen data (unseen scenarios), we collect all estimated confidence values from VEA in all of the preceding experimental runs. Specifically, these predicted values come from line 19 of Algorithm 5. We also gather the corresponding actual accuracy value in the testing phase in all of our 10-fold cross-validation experiments. By comparing these predicted accuracy values and actual accuracy values, its performance on the unseen data can be evaluated. Both mean absolute error (MEA) and R^2 statistics are shown in the last row in Table V. The MAE value, calculated using Equation (14), indicates that on average our predicted confidence value has a 5% difference to the actual accuracy value, whereas the R^2, which closes to 1, indicates good prediction.

5. CONCLUSIONS

In this article, we present our VEA for the authorship attribution problem. To facilitate its interpretability and explainability, it is designed according to the EEDI framework and is able to visualize and corroborate the linguistic evidence supporting our output attribution results. Additionally, we conducted comprehensive experiments to fully evaluate our VEA approach and have shown that it can achieve state-of-the-art authorship attribution accuracy. We have noticed the scalability issues of this method; when dealing with a scenario that includes more than 20 candidates, it is more suitable to identify a small subset of candidates using other scalable methods, then after that employ our method to construct cumulative visualized evidence.

Our future research will focus on the following directions. First, to evaluate the performance of the proposed approach regarding the precision versus recall measure (e.g., F-measure and ROC cure) on the open-set authorship problem, systematic experiments are required to be carefully designed and conducted. Second, tremendous works already exist in the literature of machine learning for classifier combination, and we will explore more deeply for higher accuracy in the future. In addition, the proposed visualization scheme is not very applicable to long anonymous documents. One possible solution is to remove *n*-grams that share similar lightness among hypotheses.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers and the editor for the thorough reviews and valuable comments, which significantly improve the quality of this article.

REFERENCES

Ahmed Abbasi and Hsinchun Chen. 2006. Visualizing authorship for identification. In *Intelligence and Security Informatics*. Lecture Notes in Computer Science, Vol. 3975. Springer, 60–71.

Ahmed Abbasi and Hsinchun Chen. 2008. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems* 26, 2, Article No. 7.

Seymour Bosworth, Michel E. Kabay, and Eric Whyne. 2012. *Computer Security Handbook*. Wiley.

Michael Brennan, Sadia Afroz, and Rachel Greenstadt. 2012. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Transactions on Information and System Security* 15, 3, Article No. 12.

John Burrows. 2007. All the way through: Testing for authorship in different frequency strata. *Literary and Linguistic Computing* 22, 1, 27–48.

Tantek Çelik, Chris Lilley, and L. David Baron. 2012. CSS Color Module Level 3.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 3, Article No. 27.

Marco Cristani, Giorgio Roffo, Cristina Segalin, Loris Bazzani, Alessandro Vinciarelli, and Vittorio Murino. 2012. Conversationally-inspired stylometric features for authorship attribution in instant messaging. In *Proceedings of the 20th ACM International Conference on Multimedia*. ACM, New York, NY, 1121–1124.

Walter Daelemans. 2013. Explanation in computational stylometry. In *Computational Linguistics and Intelligent Text Processing*. Lecture Notes in Computer Science, Vol. 7817. Springer, 451–462.

Kai-Bo Duan and S. Sathiya Keerthi. 2005. Which is the best multiclass SVM method? An empirical study. In *Multiple Classifier Systems*. Lecture Notes in Computer Science, Vol. 3541. Springer, 278–285.

Hugo Jair Escalante, Thamar Solorio, and Manuel Montes-y-Gómez. 2011. Local histograms of character *n*-grams for authorship attribution. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 228–298.

Yoav Freund and Robert E. Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the 2nd European Conference on Computational Learning Theory*. 23–37.

Benjamin C. M. Fung, Ke Wang, and Martin Ester. 2003. Hierarchical document clustering using frequent itemsets. In *Proceedings of the 3rd SIAM International Conference on Data Mining (SDM)*.

Hans Van Halteren. 2007. Author verification by linguistic profiling: An exploration of the parameter space. *ACM Transactions on Speech and Language Processing* 4, 1, Article No. 1.

Mark Harrower and Cynthia A. Brewer. 2003. Colorbrewer.org: An online tool for selecting colour schemes for maps. *Cartographic Journal* 40, 1, 27–37.

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

David I. Holmes. 1994. Authorship attribution. *Computers and the Humanities* 28, 2, 87–106.

David I. Holmes. 1998. The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing* 13, 3, 111–117.

John Houvardas and Efstathios Stamatatos. 2006. *N*-gram feature selection for authorship identification. In *Artificial Intelligence: Methodology, Systems, and Applications*. Lecture Notes in Computer Science, Vol. 4183. Springer, 77–86.

Farkhund Iqbal, Hamad Binsalleeh, Benjamin Fung, and Mourad Debbabi. 2013. A unified data mining solution for authorship analysis in anonymous textual communications. *Information Sciences* 231, 98–112.

Patrick Juola. 2006. Authorship attribution. *Foundations and Trends in Information Retrieval* 1, 3, 233–334.

Patrick Juola. 2012. Detecting stylistic deception. In *Proceedings of the Workshop on Computational Approaches to Deception Detection*. 91–96.

Patrick Juola and Darren Vescovi. 2010. Empirical evaluation of authorship obfuscation using JGAAP. In *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security*. ACM, New York, NY. 14–18.

Gary Kacmarcik and Michael Gamon. 2006. Obfuscating document stylometry to preserve author anonymity. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*. 444–451.

Sangkyum Kim, Hyungsul Kim, Tim Weninger, Jiawei Han, and Hyun Duk Kim. 2011. Authorship classification: A discriminative syntactic tree mining approach. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 455–464.

Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas. 1998. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 3, 226–239.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2006. Authorship attribution with thousands of candidate authors. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 659–660.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2011. Authorship attribution in the wild. *Language Resources and Evaluation* 45, 183–94.

Moshe Koppel, Jonathan Schler, Shlomo Argamon, and Yaron Winter. 2012. The "fundamental problem" of authorship attribution. *English Studies* 93, 3, 284–291.

Moshe Koppel, Jonathan Schler, and Elisheva Bonchek-Dokow. 2007. Measuring differentiability: Unmasking pseudonymous authors. *Journal of Machine Learning Research* 8, 1261–1276.

Ioannis Kourtis and Efstathios Stamatatos. 2011. Author identification using semi-supervised learning. In *Proceedings of the 2011 CLEF Conference on Multilingual and Multimodal Information Access Evaluation (Lab and Workshop Notebook Papers)*.

Maarten Lambers and Cor J. Veenman. 2009. Forensic authorship attribution using compression distances to prototypes. In *Computational Forensics*. Lecture Notes in Computer Science, Vol. 5718. Springer, 13–24.

Robert Layton, Paul Andrew Watters, and Richard Dazeley. 2013. Automated unsupervised authorship analysis using evidence accumulation clustering. *Natural Language Engineering* 19, 195–120.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady* 10, 707.

Kim Luyckx and Walter Daelemans. 2011. The effect of author set size and data size in authorship attribution. *Literary and Linguistic Computing* 26, 135–55.

Justin Martineau, Tim Finin, Anupam Joshi, and Shamit Patel. 2009. Improving binary classification on text problems using differential word features. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. ACM, New York, NY, 2019–2024.

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

Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. 2012. On the feasibility of Internet-scale author identification. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP)*. 300–314.

Fuchun Peng, Dale Schuurmans, Shaojun Wang, and Vlado Keselj. 2003. Language independent authorship attribution with character level language models. In *Proceedings of the 10th Conference on the European Chapter of the Association for Computational Linguistics—Volume 1*. 267–274.

Sindhu Raghavan, Adriana Kovashka, and Raymond Mooney. 2010. Authorship attribution using probabilistic context-free grammars. In *Proceedings of the ACL 2010 Conference Short Papers*. 38–42.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24, 5, 513–523.

Conrad Sanderson and Simon Guenter. 2006. Short text authorship attribution via sequence kernels, Markov chains and author unmasking: An investigation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. 482–491.

Upendra Sapkota, Thamar Solorio, Manuel Montes-y-Gómez, and Paolo Rosso. 2013. The use of orthogonal similarity relations in the prediction of authorship. In *Computational Linguistics and Intelligent Text Processing*. Lecture Notes in Computer Science, Vol. 7817. Springer, 463–475.

Jacques Savoy. 2012. Authorship attribution based on specific vocabulary. *ACM Transactions on Information Systems* 30, 2, Article No. 12.

Yanir Seroussi, Fabian Bohnert, and Ingrid Zukerman. 2012. Authorship attribution with author-aware topic models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers—Volume 2*. 264–269.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2011. Authorship attribution with latent Dirichlet allocation. In *Proceedings of the 15th Conference on Computational Natural Language Learning*. 181–189.

Jitesh Shetty and Jafar Adibi. 2004. *The Enron Email Dataset Database Schema and Brief Statistical Report*. Information Sciences Institute Technical Report, University of Southern California.

Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. 2013. Syntactic dependency-based *n*-grams: More evidence of usefulness in classification. In *Computational Linguistics and Intelligent Text Processing*. Lecture Notes in Computer Science, Vol. 7816. Springer, 13–24.

Lawrence M. Solan. 2013. Intuition versus algorithm: The case of forensic authorship attribution. *Brooklyn Journal of Law and Policy* 21, 551, Paper No. 342.

Thamar Solorio, Sangita Pillay, Sindhu Raghavan, and Manuel Montes-y-Gómez. 2011. Modality specific meta features for authorship attribution in Web forum posts.. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*. 156–164.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology* 60, 3538–556.

Fiona J. Tweedie, Sameer Singh, and David I. Holmes. 1996. Neural network applications in stylometry: The Federalist papers. *Computers and the Humanities* 30, 1, 1–10.

Yiming Yang and Jan O. Pedersen. 1997. A Comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*. 412–420.

Justin Zobel and Alistair Moffat. 1998. Exploring the similarity space. *ACM SIGIR Forum* 32, 1, 18–34.

Received March 2014; revised October 2014; accepted November 2014