AUTHORSHIP IDENTIFICATION AND WRITEPRINT VISUALIZATION

STEVEN H. H. DING

A THESIS

IN

THE CONCORDIA INSTITUTE FOR INFORMATION SYSTEMS ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF APPLIED SCIENCE IN INFORMATION SYSTEMS SECURITY

CONCORDIA UNIVERSITY

MONTRÉAL, QUÉBEC, CANADA

APRIL 2014

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Steven H. H. Ding**

Entitled: **Authorship Identification and Writeprint Visualization**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science in Information Systems Security

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Chadi Assi	Chair
Dr. Amr Youssef	CIISE Examiner
Dr. Peter Grogono	External Examiner
Dr. Benjamin C. M. Fung	Supervisor
Dr. Mourad Debbabi	Supervisor

Approved by _____

Chair of Department or Graduate Program Director

_____ 20 _____ _____

Dr. Christopher Trueman, Dean

Faculty of Engineering and Computer Science

Abstract

Authorship Identification and Writeprint Visualization

Steven H. H. Ding

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 log 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; also, these sophisticated techniques are difficult for cyber investigators or linguistic experts to interpret. In this thesis, based on the EEDI (End-to-End Digital Investigation) Framework we propose a visualizable evidence-driven approach, namely VEA, which aims at facilitating the work of cyber investigation. Our comprehensive controlled experiment and stratified experiment on the real-life Enron email data set both 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.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Benjamin C. M. Fung, for his experienced guidance, constructive criticism, and persistent support on me throughout the research. He has been a tremendous mentor for me. His encouragement and advices on both my research and career, which enables me to grow as a researcher, have been priceless.

I also would like to express my sincere appreciation to my supervisor, Dr. Mourad Debbabi, for his patience, confidence, and continuous support on me to complete this research and thesis. He provides a great source of opportunities and encouragement. I am deeply grateful to him.

My sincere appreciation also goes to all the faculty members and staff of *Concordia Institute for Information Systems Engineering*. In addition, I am very grateful to *Concordia University* for giving me this opportunity to study and work.

Last but not least, I would like to express my boundless appreciation from the button of my heart to my warm family for their irreplaceable and unconditional love. Moreover, special thanks to my fiancée, Lynne, for accompanying me, also for her firm support and heartfelt understanding.

"The painter has the Universe in his mind and hands." - Leonardo da Vinci

To my parents and

Lynne

Contents

List of Figures

List of Tables

Chapter 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: literary science, sociolinguistic research, Psycholinguistics, social psychology, forensics, and medical diagnosis, etc. [Dae13] Especially under the globalized and decentralized nature of the Internet, the communications of malicious activities (e.g., illegal material distribution, ransom, and harassment, etc. [AC08, IBFD13]) 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 law court [BAG12].

The study of authorship attribution has a long-standing history [MW64] 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 [Sta09]. These models as black-box approaches can hardly be interpreted by an investigator and their output is too simple to use as evidence in a court of law.

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' [Dae13]. 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 [Sol13]. 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 [KSA11, LD11, NPG+12], authorship attribution accuracy is greatly and directly affected by many objective factors (e.g., 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 in order to 'exclude a bogus conclusion based on inadequate data' [Sol13] when applied to real-life legal cases.

In this thesis, 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 law court. Our approach is driven by evidence and based on the lazy learning scheme [NPG+12]. Basically, our method searches inside the anonymous document for all the writing styles of different linguistic modalities as evidence and matches them to the pre-built candidate profiles. Evidence from different linguistic modalities are combined by using confidence estimation. Finally, it visualizes all 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 (i.e., varying length of text, varying candidate size, etc.) to fully evaluate our method.

1.1 The problem

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 [MW64, Hol98, IBFD13]. In most of the previous studies, the candidate sets involved in their scenarios are mostly of size ranging from 2 to 20. Although the size of a real-life candidate set may scale up to more than ten thousand, it is more appropriate to first employ scalable methods from [KSA11] or [NPG+12] to determine a potential candidate subset, and then

use other relatively more accurate techniques to figure out the most plausible conclusion.

An open-set authorship attribution problem is a variant of the original authorship attribution problem [KSA11]. 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.

We formally define the authorship identification problem with a probability confidence value output, as mentioned above. To be consistent in terminology, in this thesis "candidates" or "candidate authors" refer to the potential authors of the anonymous message, and "author" or "actual author" refer 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 $M = \{M_1, M_2, ..., M_N\}$ 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 given problem context: $p = 0$ indicates an completely uncertain result, while $p = 1$ indicates a very confident result.

1.2 Challenges and Contributions

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 [Hol94, Sta09]. 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 [IBFD13], 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.

In order 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 EEDI (End-to-End Digital Investigation) framework [BKW12], one of the recognized forensic processes used

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

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[1] in Figure 1, where the correlations among fingerprint minutiae can be visually compared, rather than presenting a simple numeric result we devise an approach visualizing all the supporting evidence on top of our visual representation of hypotheses. 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 in order to fully evaluate our approach. Our experiments demonstrate that our approach achieves the state-of-the-art attribution accuracy, while the output evidence is visualizable, presentable, and explainable.

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

- Based on the specific context of the given authorship attribution problem, our approach is also able to estimate a confidence value and, thus, can be applied to the authorship open-set problem. 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 [KSA11], it is the first trial to combine multiple classifiers by normalizing their scoring vector 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 differing 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 non-linguistic features can be further added as additional events.

1.3 Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 reviews and discusses recent development and issues in authorship analysis. Chapter 3 presents our analysis on static stylometry. Chapter 4 elaborates our *Visualizable Evidence-driven Approach* of authorship

attribution in detail. Chapter 5 evaluates our proposed method *VEA* on the Enron real-life dataset. Chapter 6 concludes this thesis.

Chapter 2

Related Works

The history of authorship attribution backed up by computational and statistical methods can be dated from the 19th century [Sta09]. 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 few have 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 chapter we are going to 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 [Hol94, Juo06, Sta09].

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 [Sta09, BAG12]. Stylometric features can be categorized into different linguistic levels [Dae13, Sta09], or, more precisely, linguistic modalities [SSMyR13, SPRMy11]. Various features of different modalities have demonstrated their effectiveness in distinguishing human writing patterns. These modalities include lexical [KSA06, Hal07, Sav12], character-based [KSA11, KSAW12, ESMy11], syntactic [KKW$^+$11, SVS$^+$13, RKM10], semantic [HS11, SZB11, SBZ12] and application-specific modality [CRS$^+$12].

Among all 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 [LD11, KSA11]. The character n-gram model actually captures information crossing different modalities [HS06]; 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 [NPG$^+$12], solutions using these features also take the risk of capturing the context rather than the authors' writing style. Regarding the relationship between stylometric modalities, [SSMyR13] employed the word "orthogonal" to assimilate them as independent components. In fact, this word appears to be over-dramatic because correlations among modalities do exist. For example, some functional words in lexical modality have exactly one corresponding Part-of-Speech tag in the syntactic modality (e.g., 'to' to

POS tag 'TO'). We argue that correlations may exist among linguistic modalities, but they have differing 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 distinct 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 distinct feature set, candidates are given different feature sets. As shown by [AC08] and [IBFD13], 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 a similarity-based approach [PSWK03, Hal07, KSA11] and a machine-learning-based approach [SG06, LV09]. The similarity-based approach employs distance functions [Sav12] to quantify the similarity between a candidate profile and a given anonymous document, while the machine-learning-based approach builds complicated models to classify the given document. Those solutions that have the best performance on benchmark data sets are mostly machine-learning related.[1] Among the machine-learning-based approaches, the SVM-based approach [AC08] and the association-rule-based approach [IBFD13] achieve higher accuracy due to the fact that they both consider the combination of feature values among the high-dimensional space. Other machine-learning techniques are also employed,

[1] Contest organized in 2004 ALLC/ACH

involving decision tree, Artificial Intelligence [TSH96], and clustering [LWD13]. Typically, one-versus-all SVM is chosen as the standard method when comparing different stylometric features because it has a better multi-class classification capacity [DK05].

Even though a machine-learning-related approach can achieve a higher quantitative performance, most involve a complicated computational model, and it is difficult to 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 [KSA11], 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 [KS11], a co-training approach is employed by using two classifiers. Also, in [RKM10], 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 (e.g., the length of an anonymous snippet, candidate score distribution, training size, 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 it is weighted based on its demonstrated consistency among prior written samples.

2.4 Adversary Stylometry

From the perspective of the adversary, several studies are trying to circumvent authorship attribution techniques [KG06, JV10, BAG12]. The most influential study is by [BAG12]. They conducted an experiment on the effectiveness of stylometry obfuscation and imitation. By recruiting volunteers and using the Amazon Mechanical Turk[2] 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. Also 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

[2] `https://www.mturk.com/mturk/welcome`

passages are about the description of participants' neighbours while pre-existing writing samples are mostly "scholarly", and thus 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 know, word-level tokens are good at capturing contextual and thematic correlation [FWE03]. We ran our model based on pure lexical n-gram on their data set 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. Also in the study of [Juo12], 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 gram usage between pre-existing samples and obfuscated samples. The difference in the 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 Result 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 [KSA11, NPG+12]. However, due to the fact that 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 [AC06]. Nonetheless, their coordinate graph-based visual representation of the output result is still too abstract from the intuitive linguistic characteristics.

Chapter 3

Analysis of Static Stylometry

In this chapter, we present our study of the static stylometric features regarding its effectiveness when employed with the similarity-based models for authorship attribution. Firstly we discuss the schemes for data representation and analyze the distance functions, which quantify the proximity among individual candidate writing styles in the similarity-based solutions for authorship identification. After that we present two visualization methods for analyzing the variation of writing styles among candidate authors. In the end we discuss their capacities and limitations.

As shown by the following discussion, the similarity-based approaches with static stylometric feature set cannot achieve higher identification accuracy than the state-of-the-art identification techniques such as [IBFD13]. Moreover, diagram-based visualization scheme for static stylometric features can be easily interpreted and it is useful for the purpose of feature analysis. However, to visualize the writeprint for authorship identification, it fails to consider the combination of different stylometric features and in each case the

number of diagrams for user to inspect is overwhelming.

3.1 Static Stylometry and Data Representation

As mentioned in Chapter 2, stylometric features can be categorized based on their linguistic properties, more precisely, linguistic modalities. However, based on their representations, these features can also be divided into *static features* and *dynamic features* [LWD12]. Static features are chosen before the training phase, and they are independent to the dataset, while dynamic features are chosen as part of the training process [LWD12]. For example, the average length of sentences is a static feature, and the frequency value of the most frequent noun in the training corpus is a dynamic feature. To solve the problem defined in Section 1.1, initially we consider employing the static stylometric features which have been predominantly adopted in previous studies [AC08, Sta09, NPG+12, BAG12, IBFD13] until very recently [KSA11, LWD12].

In the literature of stylometry, lots of features have been developed for the purpose of modeling writing styles [Juo06, ZLCH06, Sta09]. We summarize the static stylometric features in [IBFD10] and list them in Table 1. These features cover character level modality, lexical level modality and syntactic modality. As shown in Table 1, all of these static features are of type numeric and are calculated using the frequency value or ratio value. They model the preference and behavior of an individual on using specific vocabulary and grammatic structures in his/her writings. These features have demonstrated accurate authorship identification in varying settings [AC08, BAG12, IBFD13].

Table 1: Static features summarized from [IBFD10].

Features type	Features
Character features (character-based)	1. Character count (P) 2. Ratio of digits to P 3. Ratio of letters to P 4. Ratio of uppercase letters to P 5. Ratio of spaces to P 6. Ratio of tabs to P 7. Occurrences of alphabets (A-Z) (26 features) 8. Occurrences of special characters: $<> \%\|\{\} \ldots$ (21 features)
Lexical features (word-based)	1. Token count(T) 2. Average sentence length in terms of characters 3. Average token length 4. Ratio of characters in words to P 5. Ratio of short words (1-3 characters) to T 6. Ratio of word length frequency distribution to T (20 features) 7. Ratio of types to T 8. Vocabulary richness (Yule's K measure) 9. Hapax legomena 10. Hapax dislegomena
Syntactic features	1. Occurrences of punctuations and function words (311 features).

To represent a snippet as a numeric vector using these predefined features, there are two major approaches. The first approach is to treat each snippet as a standalone sample, and the vector for this sample is calculated independently using predefined features. In this case, there are $|M_i|$ samples for candidate author C_i.

The second approach treats all the snippets written by one specific author as a text corpus, and only one vector $vector^{C_i}$ is calculated for each author. In this case, there is no need to combine vectors of written snippets for deriving a final representation for writing style. We analyze both of these two approaches and discuss their performance in the following section.

3.2 Similarity-based Approach and Distance Functions

As mentioned in Chapter 2, attribution techniques are employed to predict the most plausible author after representing text snippets as numeric vectors. In this section, we analyze similarity-based approach which employs distance functions to quantify the similarity between anonymous snippet and candidate authors' writing style, due to the fact that data mining related techniques, such as SVM, introduce complicated computational models and thus they can be hardly visualized.

For the first approach of data representation, each candidate author has $|M_i|$ vectors. To derive a final vector for candidate author C_i, the typical mean center vector is employed: $vector^{C_i} = \frac{1}{|M_i|} \sum_{doc}^{M_i} vector^{doc}$. For the second data representation approach, this step is unnecessary since each candidate already has one dedicated vector $vector^{C_i}$.

Assuming that we have SF static features in total, to quantify the proximity between $vector^{C_i}$ and $vector^{\omega}$ for anonymous snippet ω, we consider following typical distance functions:

- Euclidean distance:

$$dist(vector^{C_i}, vector^{\omega}) = \sqrt{\sum_{k=1}^{SF}(vector_k^{C_i} - vector_k^{\omega})^2}$$

- Cosine distance:

$$dist(vector^{C_i}, vector^{\omega}) = \frac{vector^{C_i} \cdot vector^{\omega}}{|vector^{C_i}| \times |vector^{\omega}|}$$

- Pearson distance:

$$dist(vector^{C_i}, vector^{\omega}) = 1 - \frac{1}{SF} \sum_{k=1}^{SF}(\frac{vector_k^{C_i} - \overline{vector^{C_i}}}{\sigma_{vector^{C_i}}})(\frac{vector_k^{\omega} - \overline{vector^{\omega}}}{\sigma_{vector^{\omega}}})$$

Table 2: Identification accuracy using different models.

	1st data representation	2nd data representation
Euclidean distance	0.49	0.22
Cosine distance	0.48	0.33
Pearson distance	0.51	0.29
Minkowski distance p = 3	0.45	0.17
Manhattan distance	0.31	0.19
Chebyshev distance	0.29	0.25

- Minkowski distance with $p = 3$:

$$dist(vector^{C_i}, vector^{\omega}) = \big(\sum_{k=1}^{SF} |vector_k^{C_i} - vector_k^{\omega}|^p\big)^{\frac{1}{p}}$$

- Manhattan distance:

$$dist(vector^{C_i}, vector^{\omega}) = \sum_{k=1}^{SF} |vector_k^{C_i} - vector_k^{\omega}|$$

- Chebyshev distance:

$$dist(vector^{C_i}, vector^{\omega}) = max_{k=1}^{SF}(|vector_k^{C_i} - vector_k^{\omega}|)$$

To analyze the effectiveness of aforementioned data representations and distance functions for the task authorship identification, we randomly sampled three scenarios where 10 candidates are involved from the Enron email dataset. We tested the combination of aforementioned data representations and distance function using 10-fold cross validation, and used the identification accuracy (ratio of samples that are correctly identified) as our evalution measure. The test result is listed in Table 2.

This small analytical test is by no mean inclusive and comprehensive, but it turns out that the second data representation, which considers each writing snippet as independent sample, outperforms the first representation. Also the $Pearson\ distance$ appears to at best

model the writing styles. However, 51% identification accuracy still is incomparable with other state-of-the-art identification approaches such as [IBFD13].

3.3 Analysis through Visualization

In this section, we present two visualization schemes for static stylometric analysis. The first scheme is spectrum based approach, which is inspired from the *information gain* theory. Four examples are shown in Figure 2, respectively based on the static feature described below the diagram. In this visual representation scheme, the spectrum stands for the ascending feature value, and different colour stands for writing style for different candidate author. For example, in the first diagram on the left, colour blue mostly gathers in the upper area on spectrum, and this indicates that the candidate author corresponding to colour blue demonstrates high number of characters per sentence in his previous writing samples. Each horizontal line on the spectrum stands for the demonstrated usage on this specific value. If the colour of this line is purely only one specific colour, it means that this specific value on this feature is only revealed on the writing samples of the candidate author corresponding to this colour. For example, if a horizontal line is separated into two parts of equal length with different colour, it means that this specific value on this feature is demonstrated equally in the writing samples of these two candidate authors.

To calculate each line, we apply Equation 1 for each candidate author. f stands for feature type, fv stands for the given specific feature value, L stands for the spectrum width and col_i stands for the colour for candidate i. By combing these $Length$ values calculated

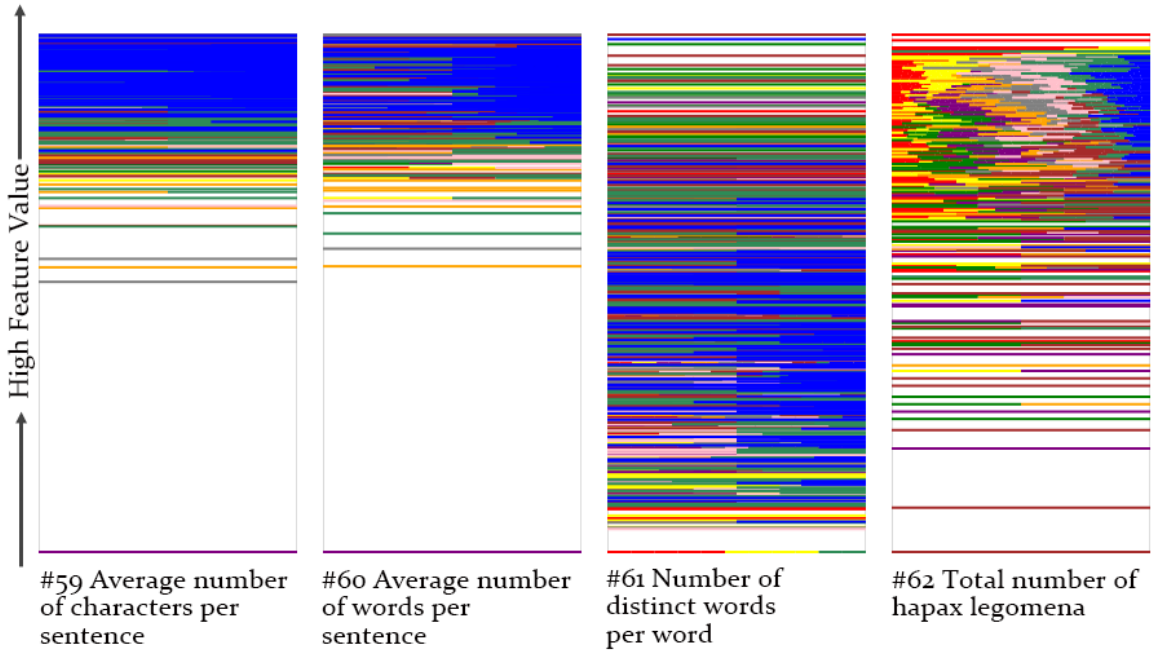

Figure 2: Examples of spectrum-based information-gain-inspired writeprint visualization scheme. Different color represents writing style of different candidate author. Spectrum value represents ascending feature value.

for all the candidate authors, a horizontal line for feature f can be obtained.

$$Length(f, fv, col_i) = L \times \frac{|\{m | m \in M_i \ \& \ m \ reveals \ fv \ on \ f\}| / |M_i|}{\sum_{k=1}^{N} |\{m | m \in M_k \ \& \ m \ reveals \ fv \ on \ f\}| / |M_k|} \quad (1)$$

This approach visualizes the variation of writing styles on one specific feature among all candidate authors. It also visualizes the discriminant power of one feature for the problem of authorship identification. The horizontal lines in the spectrum directly stands for the feature values revealed in previous writing samples, and in this way, it is easily interpretable for the user. However, this scheme only considers one feature in one spectrum, and it is hard to identify possible outliers for each candidate.

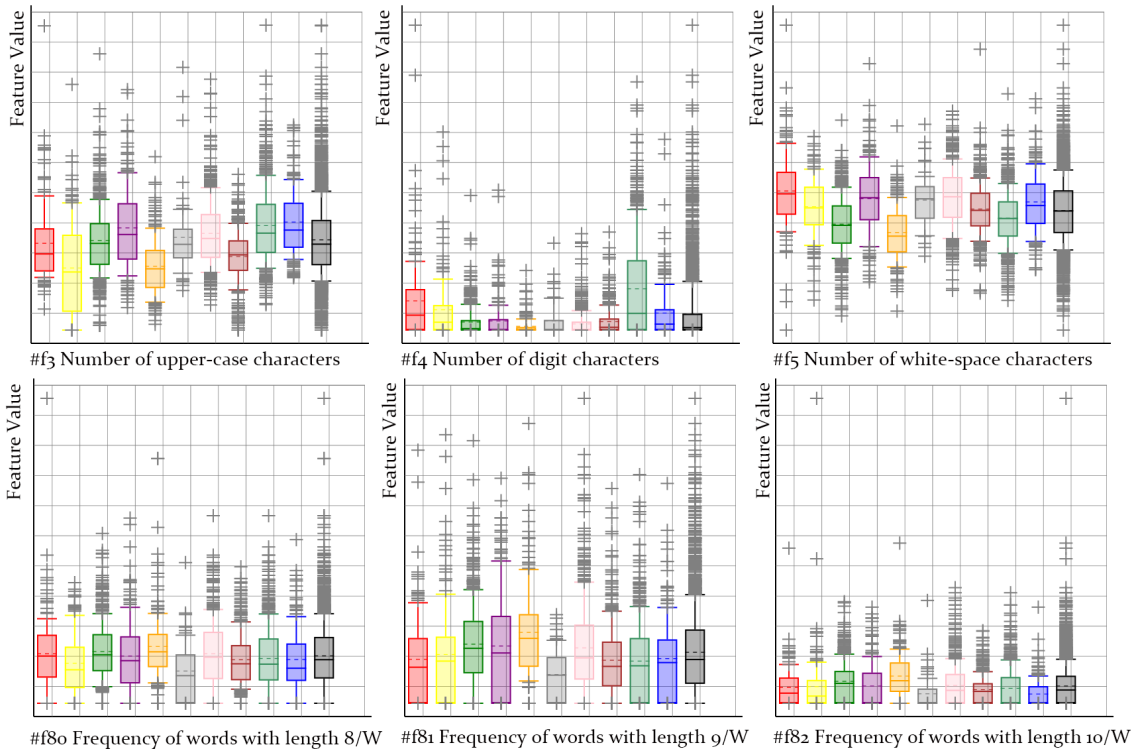

Figure 3: Examples of box-plot-based writeprint visualization scheme. Different colour (series on the diagram) stands for different candidate authors' previous writing sample M_i.

Examples of the second visualization scheme for feature analysis are shown in Figure 3. This scheme is based on the feature value distribution and it employs the box plot to scatter the demonstrated values on specific feature for each candidate author. In this scheme, different colour stands for different candidate author, and the box plot with corresponding colouring represents how previous writing samples of this candidate author distribute on this feature.

This scheme is able to demonstrate the variation writing styles among all the candidate authors on a specific stylometric feature. Differing to the first spectrum-based scheme, all the outliers as well as the mean value and the distributional variance for each author can be easily identified. However, similar to the spectrum-based scheme, this scheme also fails to

consider the combination of feature values.

These two schemes are suitable for feature analysis and individual visualization. However, for the task of authorship identification, they are impractical since they fail to combine all the features together, which assumes that the user has to inspect them one by one in each case. This assumption is unpractical since the number of features could be even more overwhelming. Also, these two types of scheme fail to answer which candidate author is more similar to the anonymous snippet ω, and thus fail to visualize the solution for the authorship identification problem.

In the next chapter, we present our evidence-driven approach, which depends on dynamic features. By combining similarity based identification approach and data mining approach, it achieves state-of-the-art identification result, at the same time its output can be visualized and interpreted. Unlike previous two types of visualization scheme, this approach aims at visualizing the identification result, and it fits all the features in one diagram in an interpretable way. Based on cumulative visual effect on the diagram, the most plausible author can be determined.

Chapter 4

A Visualizable Evidence-driven Approach for Authorship Identification

In this chapter, we present our visualizable evidence-driven approach for the authorship attribution problem, addressing the issues and problems mentioned in Chapter 1. For this approach we employ the dynamic stylometric feature set, which is different to the static feature set that applied in previous chapter. In Chapter 5, we will present the experiments that compare their performance in varying identification context.

For the purpose of promoting its interpretability and explainability, our approach is designed according to the nine processes defined by the End-to-End Digital Investigation framework (EEDI) [BKW12]. 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 of structurally organizing

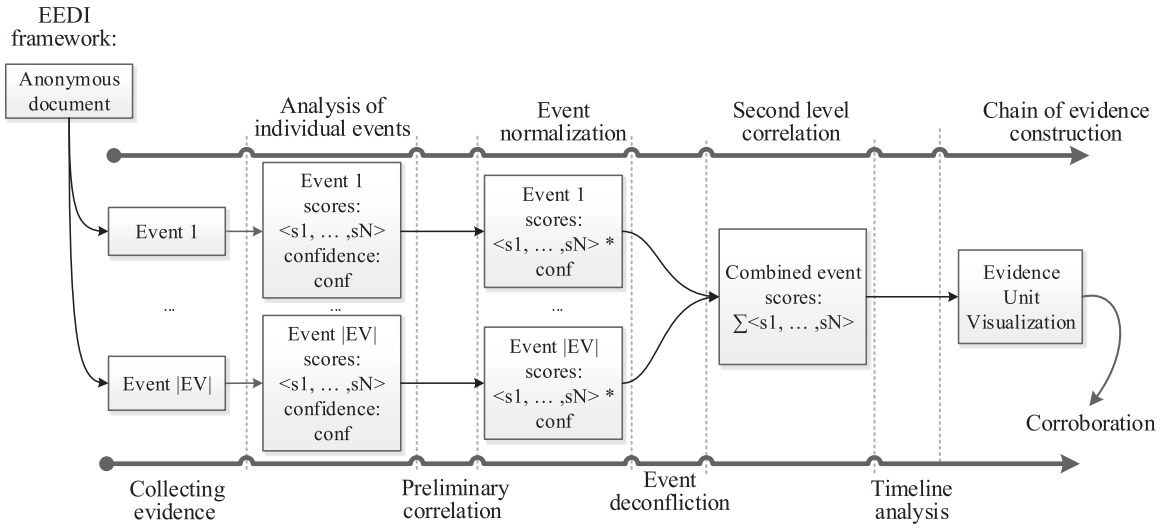

Figure 4: Overview of VEA in EEDI framework.

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. Also, the process of chain construction can be easily explained by employing the EEDI framework. The briefs of procedures employed are outlined in Figure 4.

To begin with, we formally define the term *authorship hypothesis* (see Definition 1). Basically an authorship hypothesis is a statement that claims a candidate to be the author of a given anonymous snippet ω. According to the problem defined in Sectioin 1.1, where N candidate authors are involved, N hypotheses are thus formulated, respectively targeting on each candidate in C.

Definition 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.

4.1 Collecting Evidence

The first phase in the original EEDI framework is Collecting Evidence [BKW12]. 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, etc. They required 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 the linguistic evidence. Likewise, linguistic characteristics reflected on the given snippet ω are of varying types based on their particular linguistic modalities (e.g., syntactic, lexical, and character-based, etc.), and linguistic characteristics of certain modality require specific techniques for feature extraction [Sta09]. 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 2). Evidence unit is the minimum scoring unit

Modality	Characteristics	Details	Examples
Lexical	Word Level N-gram	Length:1-8	'It is noticed', 'is noticed and appreciated'
Character	Character Level N-gram	Length:1-8	no, not, notic, tice, notice, a, an, and
Syntactic	POS N-gram	Length:1-8	PRP VBZ VBN, CC VBN, TO DT NN

Table 3: Employed lingusitic features.

and minimum visualization unit, which will be further discussed in Section 4.2.

Definition 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 $(v_1, ..., v_i, ..., v_N)$, 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 thesis 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 n-gram [Sta09]. Refer to Table 3 for detailed information and examples. The length of these grams varies from 1-8 because we can hardly find any gram present repetitively with length more than 8. We employ n-gram technique because previous studies [KSA11, Sav12, SVS+13] show its effectiveness in capturing the writing style. Also, they are comparatively easier to visualize and present as evidence units; more details will be discussed in Section 4.5.

To preserve the explainability of our approach, unlike previous research, we do not employ any feature selection techniques such as methods in [YP97]. That means we employ the full set of grams rather than an optimal top-K subset. Previous research, such as [HS06], demonstrate that such a top-K culled subset can already achieve high accuracy

in the authorship attribution problem, but it is hard 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. To avoid any exceptional circumstance, we thus employ the full set of grams to guarantee its explainability. Even though this approach introduces high 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. *(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 $\forall eu_m^{ev_n} \in EU_{ev_n}$, $f_{eu_m^{ev_n}}$ is of type T_{ev_n}. Also $\vec{V}_{ev_n} \in \mathbb{R}^N$ 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). 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 the 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

Algorithm 1 Event Construction (EC)

Input number of candidates N, linguistic type $Type$, anonymous snippet ω
Output event ev

1: $T_{ev} \leftarrow Type$ ▷ associate this event with the given type of linguistic modality
2: $features \leftarrow$ extract all linguistic characteristics of type T_{ev} from snippet ω
3: **for** $m = 1$ **to** $|features|$ **do**
4: $\vec{v}_{eu_m^{ev}} \in \mathbb{R}^N$, $\vec{v}_{eu_m^{ev}} \leftarrow \{0\}$ ▷ initialize as a zero vector
5: $f_{eu_m^{ev}} = features_m$
6: $EU_{ev} \leftarrow EU_{ev} \cup \{eu_m^{ev}\}$
7: **end for**
8: **return** ev

constructions, all the events will be passed into the next process, as shown in Figure 4. In our case, three events are created: a lexical event, a character event, and a syntactic event.

4.2 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 correlation between each event and the overall investigation [BKW12]. 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: 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: 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.

Algorithm 2 Event-based Scoring (ES)

Input event ev, writing samples M, anonymous snippet ω
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: $\quad \vec{a}[m] = \text{tf}(f_{eu_m^{ev}}, \omega)$ ▷ this vector is for anonymous snippet ω
5: **end for**
6: **for** $i = 1$ **to** N **do**
7: $\quad \vec{c} \in \mathbb{R}^{|EU_{ev}|}$, $\vec{c} \leftarrow \{0\}$ ▷ this vector is for candidate author i
8: $\quad$ **for** $m = 1$ **to** $|EU_{ev}|$ **do**
9: $\quad\quad \vec{c}[m] = \text{tf}(f_{eu_m^{ev}}, M_i) \times \text{idf}(f_{eu_m^{ev}})$ ▷ here feature $f_{eu_m^{ev}}$ is a gram
10: $\quad\quad \vec{v}_{eu_m^{ev}}[i] \leftarrow \vec{c}[m] \times \vec{a}[m]$ ▷ store intermediate result
11: $\quad$ **end for**
12: $\quad \vec{s}[i] = \vec{a} \cdot \vec{c}$
13: **end for**
14: **return** $\vec{s}$

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

Although there exist other scoring functions that may achieve higher identification accuracy [MFJP09] [LV09], we use the $tf - idf$ scheme [ZM98] for its simplicity. As in Equation 2 and Equation 3, the tf score captures the normalized frequency of a given gram, and the idf score gives weight to each gram by considering its discriminant power. The constant Θ is used to avoid the divide-by-zero problem, and it is typically chosen as 1. We set Θ as 0.1, and in this way it is in a smaller order of magnitude when compared with $|AuthorsEverUsed(gram)|$. Other scoring schemes could be employed by considering

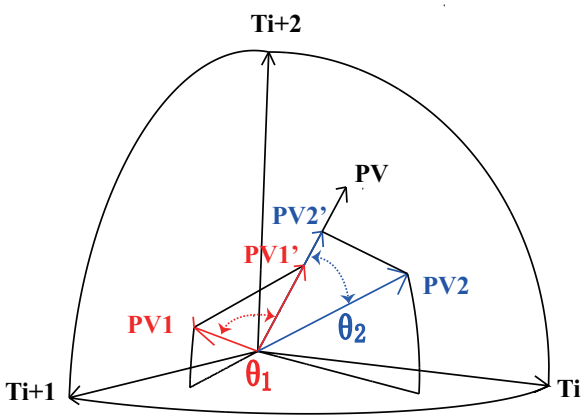

Figure 5: A sample 3-gram space.

them as separate events, which could be explored in future studies.

$$tf(gram, M_i) = \frac{frequency(gram, M_i)}{maxGramFrequency(M_i)} \tag{2}$$

$$idf(gram) = \log\left(\frac{N}{\theta + |AuthorsEverUsed(gram)|}\right) \tag{3}$$

After the construction of aforementioned $N+1$ numeric vectors, a final score is derived for each hypothesis (candidate) by comparing the similarity 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 10 in Algorithm 2.

Considering a sample 3-gram space in Figure 5, $\vec{PV_1}$, $\vec{PV_2}$, and $\vec{PV}$, respectively, are the style vectors of $candidate_1$, $candidate_2$, and the anonymous snippet ω. In previous work such as [KSA11] where n-gram related features are employed, the $cosine$ distance [SB88] 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 gram coverage and normalized

frequency of 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 of the vector in this direction. The distance function is shown in Equation 4, 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.

$$\begin{aligned} similarity(\vec{P_i}, \vec{P_\omega}) &= proj_{\vec{P_\omega}} \vec{P_i} \times \|\vec{P_\omega}\| \\ &= \|\vec{P_i}\| \times cos(\Theta_i) \times \|\vec{P_\omega}\| \\ &= \vec{P_i} \cdot \vec{P_\omega} \end{aligned} \tag{4}$$

At the end of the first analysis (see Line 13 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 4.5.

Algorithm 3 shows the second analysis. As defined in Definition 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* (e.g., anonymous snippet length, size of known-author writing, and number of candidates, etc.). Hence, we treat each event as a

Algorithm 3 Event-based Identification (EI)

Input writing samples M, candidate set C, event ev, anonymous snippet ω
Output event ev

1: $folds \leftarrow$ split(M) ▷ split M into 10 folds for cross-validation; each fold includes nine training groups and one validation group
2: $samples \leftarrow \emptyset$; ▷ create an empty set of samples; each sample follows attributes in Table 4
3: **for each** $fold$ **in** $folds$ **do**
4: $precision \leftarrow$ tests(T_{ev}, $TrainSet_{fold}$, $TestSet_{fold}$) ▷ collect precision value
5: **for each** doc **in** $TestSet_{fold}$ **do**
6: $ev' \leftarrow$ EC(N, T_{ev}, doc)
7: $scores \leftarrow$ ES(ev', $TrainSet_{fold}$, doc)
8: $sample \leftarrow$ generateSample($scores$, doc, $precision$) ▷ collect other conditions
9: $samples \leftarrow samples \cup \{sample\}$
10: **end for**
11: **end for**
12: $Model_{ev} \leftarrow$ buildModel($samples$) ▷ build a model for this event ev using precision as target attribute
13: $\vec{V}_{ev} \leftarrow$ ES(ev, M, ω) ▷ collect sample from current classification context
14: $Conf_{ev} \leftarrow Model_{ev}$.predict($\vec{V}_{ev}$, ω) ▷ estimate confidence
15: **return** ev

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 is listed in Table 4. 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 the available writing samples from M into ten groups of roughly equal size (Line 1 in Algorithm 3). Of these ten groups, one group is selected as a validation set, then the remaining nine groups are used to build events following Algorithm 1 and to predict the author of samples from the validation set by using Algorithm 2. The candidate with the highest score output (Line 7 in Algorithm 2) will be the predicted result. The next step is to construct a sample (Line 8 in Algorithm 3): the resulting precision value in this fold (i.e.,

Table 4: 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

percentage of instances that are correctly identified) will be collected as the target attribute; other attributes shown in Table 4 will be used to construct a sample. This validation process is repeated ten times and each group is used as the validation set exactly once. Based on the collected samples, a linear regression model is built for each event (Line 12 in Algorithm 3).

In Line 13, 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 with attributes in Table 4, and it is fed into the built model to derive the predicted precision value, which will be used as the confidence value (Line 14 in Algorithm 3).

Regarding the attributes used to model the identification context, in addition to using the 'gap statistic' that describes the gap between max score and the runner-up in [NPG+12, KSA11, KSA06], 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 5.4 shows that these attributes are all 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 (Line 3 to Line 12 in Algorithm 3), the intercept value in the built linear model already reflects its effect as baseline.

4.3 Event Normalization

Algorithm 4 Confidence-based Normalization (CN)

Input event ev, anonymous snippet ω
Output event ev

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

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 [BKW12]. For example, different events from different sources may have varying timing formats or different time zone settings; in order to chain them together, these formats must be normalized.

Accordingly, in our approach, after the previous process each event now has a scoring vector, while they have 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 in Algorithm 4). Also, correspondingly, we update the numeric vectors stored inside all evidence units of each event by multiplying the original score with the confidence value (Line 4 to 8 in Algorithm 4). After normalization, all the events are passed into the next process.

4.4 Secondary-level Correlation

Under the EEDI framework, this process is to examine the correlation between events and to consider ways of combining the evidence into an evidentiary chain [BKW12]. In our case, accordingly, all the events from previous process are correlated and 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.

Algorithm 5 Event Combination (EC)

Input writing samples M, candidate set C, set of event EV, anonymous snippet ω
Output $author$, confidence value p

$\vec{fs} \in \mathbb{R}^N$, $\vec{fs} \leftarrow \{0\}$ ▷ initialize final scoring vector with 0
$\mathbf{conf} \in \mathbb{R}^{|EV|}$, $\mathbf{conf} \leftarrow \{0\}$ ▷ a vector of confidence values
for $n = 1$ to $|EV|$ **do**
    for $i = 1$ to N **do**
        $\vec{fs}[i] = \vec{fs}[i] + \vec{V}_{ev_n}[i]$
    end for
    $\mathbf{conf}[n] = Conf_{ev_n}$
end for
$prediction \leftarrow$ IndexOfMaxValue($\vec{fs}$) ▷ determine the prediction result
$author \leftarrow C[prediction]$
$\mathbf{agreedConf} \in \mathbb{R}^{|EV|}$, $\mathbf{conf} \leftarrow \{0\}$
for $n = 1$ to $|EV|$ **do**
    if ev_n agrees $prediction$ **then**
        $\mathbf{agreedConf}[n] = \mathbf{conf}[n]$
    else
        $\mathbf{agreedConf}[n] = -1$
    end if
end for
p = max($\mathbf{agreedConf}$) ▷ estimate the final confidence value
return $author$, p

The procedure for evidence combination is shown in Algorithm 5. Since in previous process all the 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, Line 1 to 8 combine scoring vectors from all input events, and Line 9 determines the prediction result as the candidate author that achieves the highest score.

$$
\begin{aligned}
p &= \max_{ev_n}^{EV} P(predicted\ author \mid\ ev_n) \\
&= \max_{ev_n}^{EV} \begin{cases} Conf_{ev_n}, & \text{if } ev_n \text{ agrees on final } predicted\ author \\ -1, & \text{otherwise} \end{cases}
\end{aligned} \tag{5}
$$

To combine multiple confidence values of different classifiers, typical approaches include Product Rule, Max Rule, Min Rule, and Majority Vote Rule, etc. [KHDM98] Here we combine the Max Rule and Majority Vote Rule to derive our final estimated confidence value. As Line 12-19 in Algorithm 5 shows, the final confidence value is determined as the maximum estimated confidence value among all the events that agree on the final output candidate (also see Equation 5).

Previous research [KSA11, NPG+12] 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 6). Our experiment demonstrates that this approach can achieve higher accuracy.

$$
\vec{fs}[k] = \sum_{ev_n}^{EV} \vec{V}_{ev_n}[k] \times Conf_{ev_n} \tag{6}
$$

4.5 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 [BKW12]. 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.

$$\vec{fs}[k] = \sum_{ev_n}^{EV} \sum_{eu_m^{ev_n}}^{EU_{ev_n}} \vec{v}_{eu_m^{ev_n}}[k] \tag{7}$$

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 7 by employing the intermediate results stored in evidence units according to Algorithm 2 and Algorithm 4. $\vec{fs}[k]$ refers to the final score for candidate k in Algorithm 5, which is also the same variable in Equation 6 but is calculated using different intermediate results.

The task of this process is to visualize all the evidence units with respect to their distance 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 4.5.1. *(proportionally visualizable) Given a set of hypotheses H, we say 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 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.

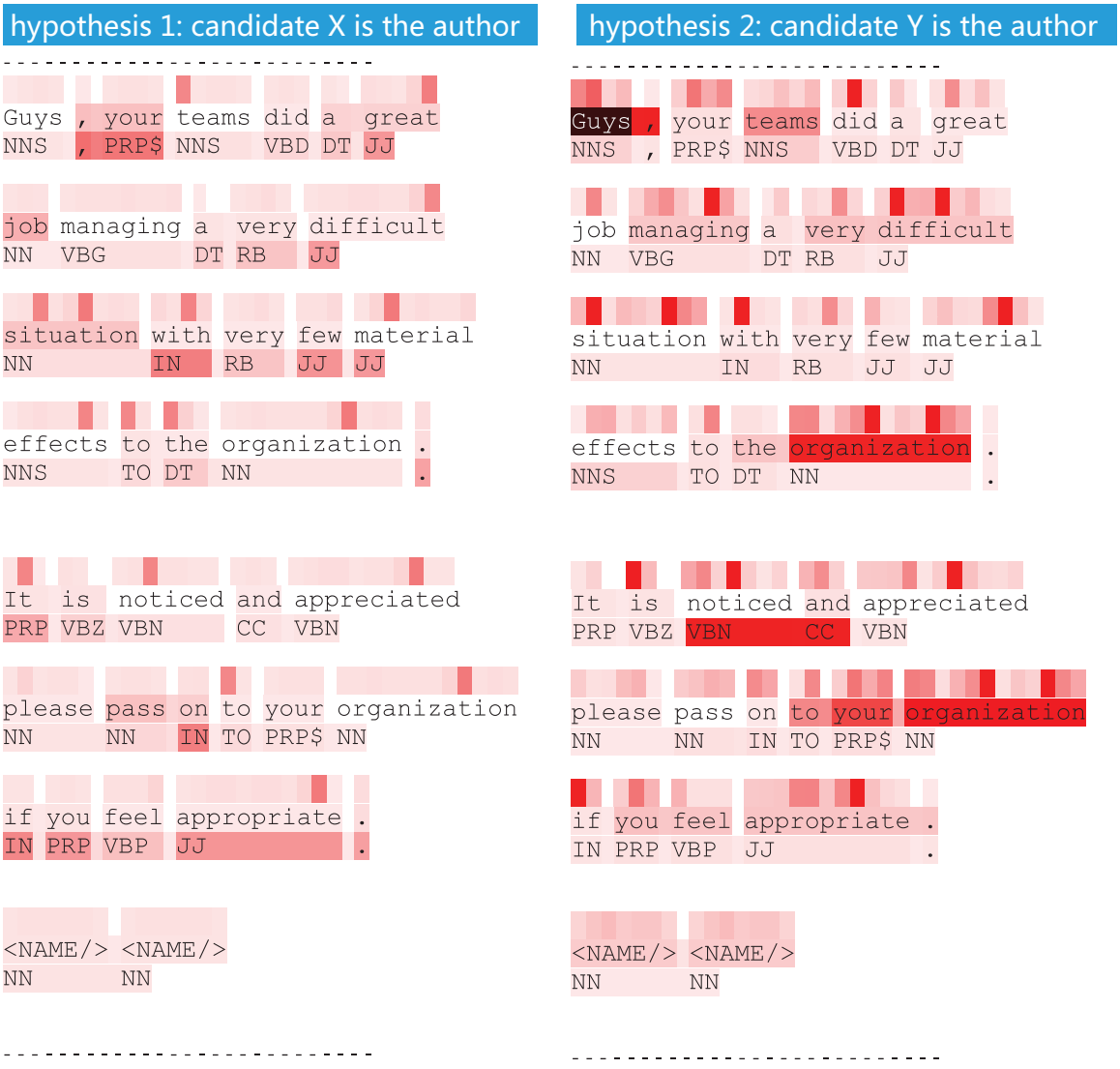

Figure 6: Evidentiary chain visualization: hypothesis representations and the visualized evidence units.

As shown in Figure 6, 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 colour 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 evident units (defined in Definition 2) by colouring each evidence unit's tokens in the above representations of hypotheses. The colour is determined by how affiliated an evidence unit is to the given hypothesis. An evidence unit hereby is our smallest visualization unit.

To colour the tokens the HSL colour scheme is employed because it is more intuitive than the RGB colour scheme [ÇLB12]. The HSL scheme encodes colour 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 colourfulness (from 0 to 100), and Lightness measures how much light should be reflected from this colour, ranging from 0 (appears as black) to 100 (appears as white); 50 is *normal* [ÇLB12]. Lightness is visually suitable as a quantitative/sequential data representation. *Dark equals more* is a standard cartographic convention [HB03] and the difference of lightness can still be perceived by people with red-green colour vision impairments [HB03]. 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 8, which represents how the original score differs from the minimum score in that scoring vector. The constant $\alpha > 1$ is used to magnify the range, avoiding assigning a blank background on $eu_m^{ev_n}$ for hypotheses k when $\vec{v}_{eu_m^{ev_n}}[k]$ equals $min(\vec{v}_{eu_m^{ev_n}})$, because if $\vec{v}_{eu_m^{ev_n}}[k] \neq 0$, $eu_m^{ev_n}$ still contributes to the overlapping effect in the colouring process, which will be discussed later.

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 first three equations in Equation 8. 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).

$$
\begin{aligned}
range'(eu_m) &= max(\vec{v}_{eu_m}) \times \alpha - min(\vec{v}_{eu_m}) \\
maxR_{ev_n} &= max(\{eu_m^{ev_n} \in EU_{ev_n} \mid range'(eu_m^{ev_n})\}) \\
maxR &= max(\{ev_n \in EV \mid maxR_{ev_n}\}) \\
dif(eu_m^{ev_n}, k) &= \frac{\vec{v}_{eu_m^{ev_n}}[k] \times \alpha - min(\vec{v}_{eu_m})}{maxR}
\end{aligned}
\tag{8}
$$

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 coloured in an overlapping manner.

$$L^{H_k}_{token_n}(eu^{ev_n}_m) = \begin{cases} L^{H_k}_{token_n} - \eta \times dif(eu^{ev_n}_m, k), & \text{if } eu^{ev_n}_m \text{ stem from } token_n \\ L^{H_k}_{token_n} & \text{otherwise} \end{cases} \quad (9)$$

Given a visual representation of hypothesis H_k, we start by initializing all tokens' backgrounds with a maximum lightness value (i.e., the background colour reflects 100% light and appears to be blank), and then we enumerate tokens in the hypotheses representation to apply Equation 9. Given a $token_n$ in H_k, for each previously extracted evidence unit $eu^{ev_n}_m$, if $f_{eu^{ev_n}_m}$ stems from $token_n$ then the token's lightness value degrades by the multiplication of degradation factor η and its normalized variant score $dif(eu^{ev_n}_m, k)$. 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. $eu^{ev_n}_m$ stems from $token_n$ means that the evidence units $eu^{ev_n}_m$ partially or completely originates from the $token_n$. For example, evidence unit "your organization" can stem from token "your" in phase "to your organization" but not from the token "your" in phase "your teams".

Since this "stem" mapping between tokens and evidence units is identical for all the hypotheses, given the same evidence unit the lightness value of a token is inversely proportional to the score $dif(eu^{ev_n}_m, k)$ of the hypothesis. In this way, it is also inversely proportional to the original score $\vec{v}_{eu^{ev_n}_m}[k]$ (see Equation 10).

$$
\begin{aligned}
L^{H_k}_{token_n}(eu^{ev_n}_m) &\propto dif(eu^{ev_n}_m, k)^{-1} \\
&\propto (\vec{v}_{eu^{ev_n}_m}[k] - min(\vec{v}_{eu^{ev_n}_m}))^{-1} \\
&\propto \vec{v}_{eu^{ev_n}_m}[k]
\end{aligned}
\tag{10}
$$

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 11.

$$
\begin{aligned}
vf_{VEA}(H_k) &= GD(H_k) \\
&\propto GL(H_k)^{-1}
\end{aligned}
\tag{11}
$$

It can be shown that this visual function satisfies Property 4.5.1 as follows: First, the global lightness function $GL(H_k)$ for hypothesis k is formulated as the cumulative lightness of all tokens (see Step 1 in Equation 12). By combining Equation 10, the $GL(H_k)$ function is inversely proportional to the final score of hypothesis k (see Step 2-5 in Equation 12).

$$
\begin{aligned}
GL(H_k) &= \sum_{token_n}^{tokens(H_k)} \sum_{ev_n}^{EV} \sum_{eu_m^{ev_n}}^{EU_{ev_n}} L_{token_n}^{H_k}(eu_m^{ev_n}) \\
&\propto \Big(\sum_{token_n}^{tokens(H_k)} \sum_{ev_n}^{EV} \sum_{eu_m^{ev_n}}^{EU_{ev_n}} dif(eu_m^{ev_n}, k) \Big)^{-1} \\
&\propto \Big(\sum_{token_n}^{tokens(H_k)} \sum_{ev_n}^{EV} \sum_{eu_m^{ev_n}}^{EU_{ev_n}} \vec{v}_{eu_m^{ev_n}}[k] \Big)^{-1} \\
&\propto \Big(\sum_{ev_n}^{EV} \sum_{eu_m^{ev_n}}^{EU_{ev_n}} \vec{v}_{eu_m^{ev_n}}[k] \Big)^{-1} \\
&\propto \Big(\vec{fs}[k] \Big)^{-1}
\end{aligned}
\tag{12}
$$

In this way, by combining Equation 11, the visual function $GD(H_k)$ is proportional to the final score of hypothesis k (see Equation 13). Thus, our selected presentation of hypothesis and evidence unit satisfies Property 4.5.1 over visual function $GD(H_k)$, which indicates that the darker the hypothesis representation's holistic colour is, the higher final score this hypothesis possesses.

$$
\begin{aligned}
vf_{VEA}(H_k) &= GD(H_k) \\
&\propto GL(H_k)^{-1} \\
&\propto \vec{fs}[k]
\end{aligned}
\tag{13}
$$

After all the aforementioned colouring is done, one can conclude that the hypothesis with the most holistically darkest colouring representation is the most plausible one. As the example in Figure 6 demonstrates, representation of *hypothesis 2* is more holistically

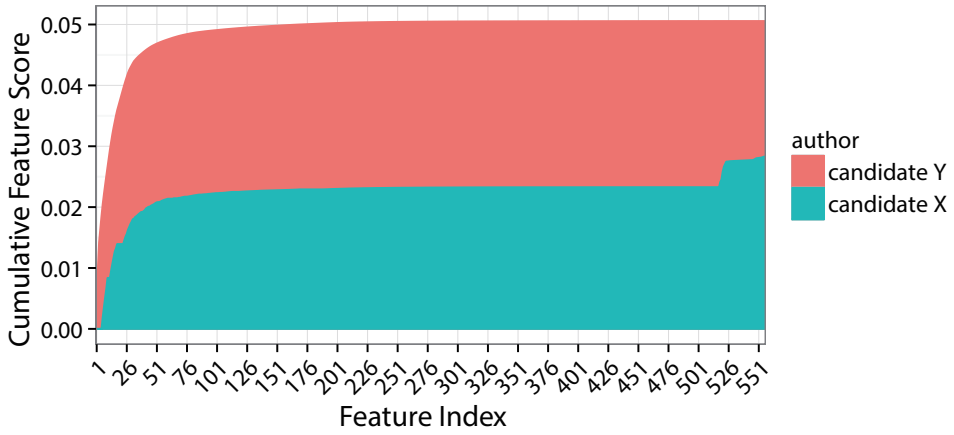

Figure 7: Cumulative evidence unit scoring diagram: the serial that achieves the highest score at the end of x-axis is for the most plausible candidate.

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 7. An area with a different colour 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 6, while this scoring diagram 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 also list all the estimated confidence values in Table 5. In this example, since all three events agreed on same plausible hypothesis, the overall confidence value is simply the maximum: one.

Table 5: 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

4.6 Corroboration

Note that linguistic evidence is only one kind of event, other non-linguistic 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 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 all the linguistic and non-linguistic events to corroborate their final hypothesis on the incident.

4.7 Implementation of Forensic Software for Authorship Identification

To implement the aforementioned approach into a forensic software, we firstly design our software architect from the perspective of *Object Oriented Programming* (OOP) [CS11] and *Aspect Oriented Programming* (AOP) [KLM+97] for the purpose that other authorship identification or verification techniques can be implemented and directly loaded into this

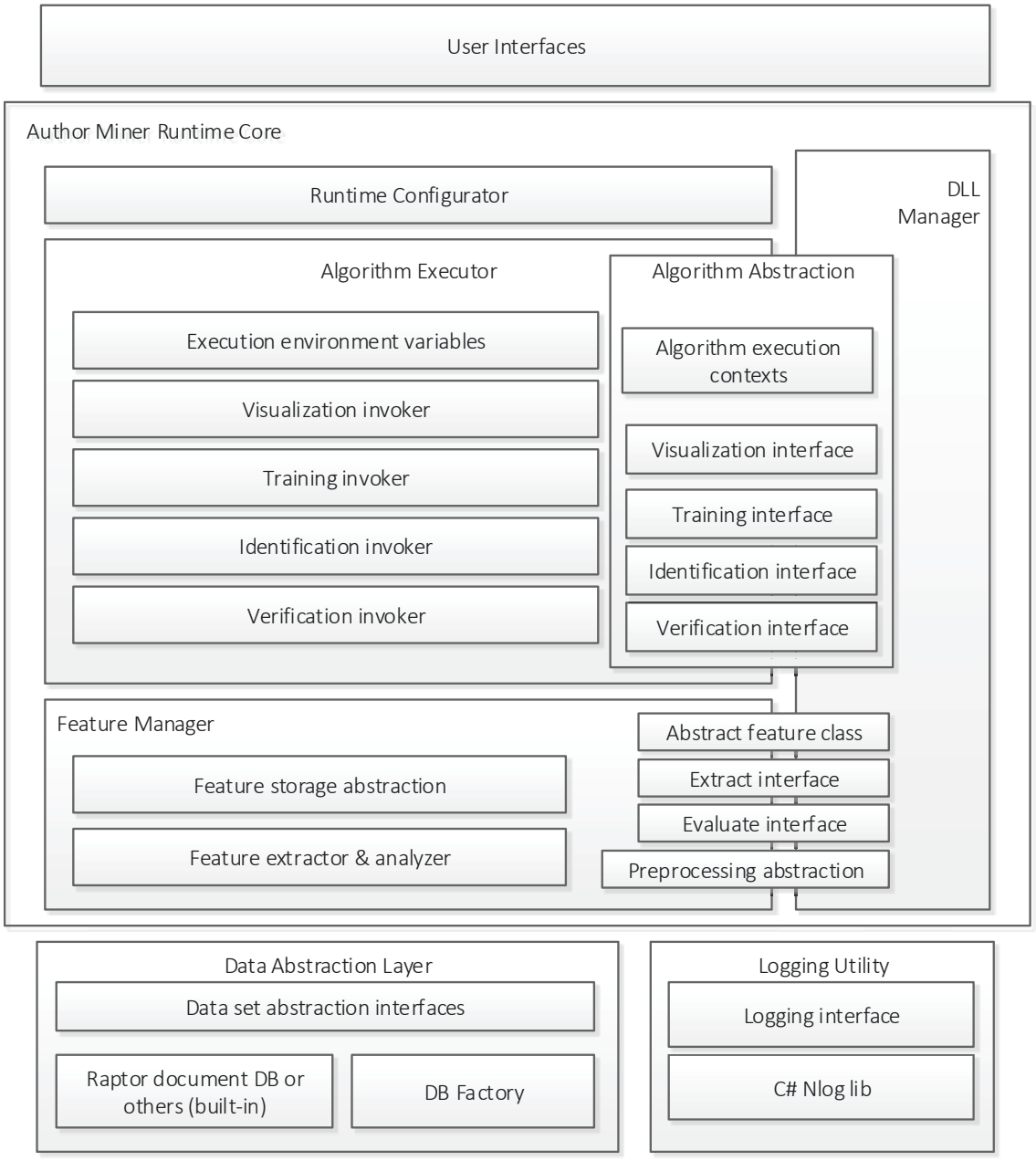

Figure 8: The architect design of the forensic software for authorship analysis.

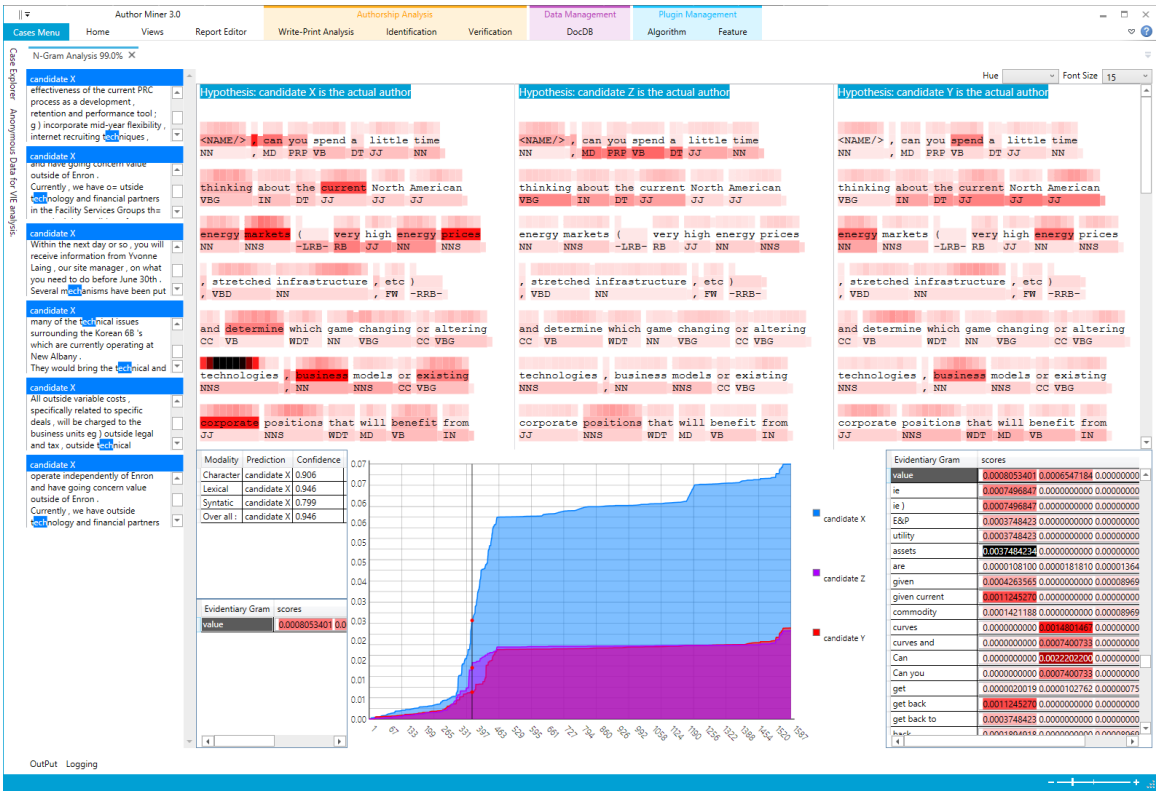

Figure 9: Implemented forensic software.

software in the future. This software is based on the Windows .Net Framework[1] 3.5 platform or its upper versions. The architect design of this software is shown in Figure 8. This software enables users to create cases, import text samples, conduct analyses, and generate reports.

This software can be divided into four critical components: logging utility, document data abstraction and database, runtime core, and user interface. Logging utility we employ the open source NLog[2] library. Document database we choose persistent Raptor[3] document database. The data abstraction layer and the logging layer are implemented as interceptors

[1].NET Framework `http://msdn.microsoft.com/en-us/vstudio/aa496123`

[2]NLog lib for .Net available at `http://nlog-project.org/`

[3]Raptor document database available at `http://www.codeproject.com/Articles/375413/RaptorDB-the-Document-Store`

in AOP, and in these ways they are cross-cutting concerns separated to the core concern of authorship analysis. For the user interfaces of software workspace we employ open source project AvalonDock[4] and Fluent[5] ribbon control. The main execution and thread managing component is the runtime library, which controls the execution of algorithm, manages the hot plugging DLL components, and coordinates the data flow.

The user interface of this software can be divided into three part: identification result and cumulative scoring diagram panel, visualized linguistic evidence panel, and evidence search panel. The first panel elaborates the prediction results and the reported confidence values from all events, and displays the cumulative scoring diagram. By clicking on the curve in this diagram, in the two panels on the both sides, the user is able to see which pieces of evidence contribute to the slope of the clicked point on this curve. The visualized evidence panel showcases our hypothesis representations and visualized linguistic evidence. The user is able to browse and see the difference in the matched evidence among all the candidate authors. Also, by clicking on the evidence (i.e., grams in our case), the user is able to see how this piece of gram is used by all candidates in their previous writing samples. All the snippets containing this clicked gram are listed on the left search panel, and the corresponding grams in all these snippets are highlighted with the candidates' colours.

In this software, the user is able to manage different authorship analysis scenarios in a case-based manner. All the imported writings and analytic results can be saved in a user specified folder. The flexibility provided by the OOP and AOP design enables the hot plugging of newly implemented authorship analytic approach. Also the document database

[4]AvalonDock, a docking control for WPF, hosted at `http://avalondock.codeplex.com/`
[5]Fluent, a ribbon control for WPF, hosted at `https://fluent.codeplex.com/`

can be changed into any other kinds through the data abstraction layer, which provides possibility of storing data in other media with encryption rather than simply on the local folder.

Chapter 5

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. The dataset that we adopted is the Enron Email dataset, which was made public by the Federal Energy Regulatory Commission [SA04]. 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 (i.e., the available samples, and available hypotheses/candidates, etc.) of the authorship attribution problem strongly affects the solution's performance, while 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.

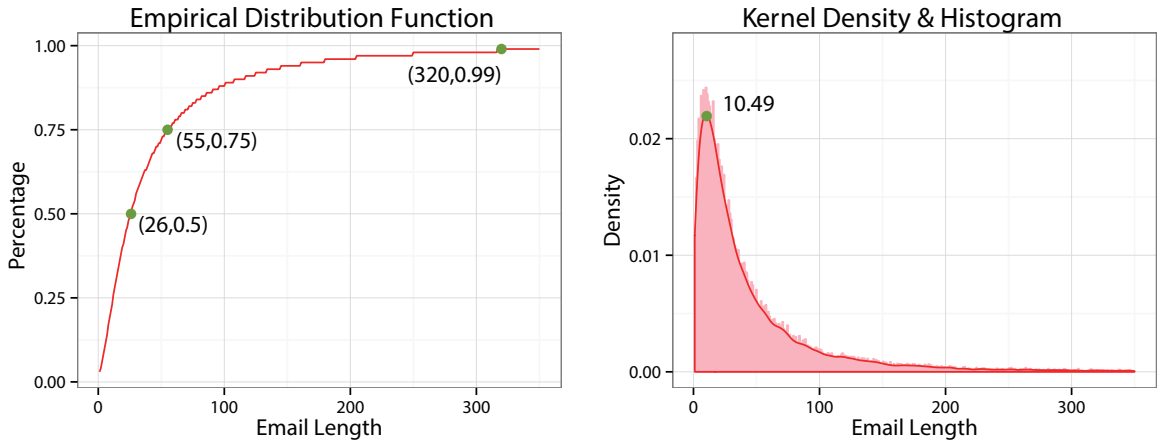

Figure 10: Dataset analysis.

5.1 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 the header information. To remove the identity-related information is relatively more complex. We completed this procedure by employing following steps:

- We utilized the regular expressions to replace URL links with the '$< link/ >$' tag.
- We utilized the Name Finder in OpenNLP[1] project to replace all the found name entries with the '$< name/ >$' tag.
- We fetched the employee information file from the data set and generated a list of

[1]available at `http://opennlp.apache.org`

first names and a list of last names. We found all the tokens that were exactly the same, case ignored, as the names in these two lists and replaced them with the '$< name/ >$' tag.

- Based on the above name lists, we found all the tokens that had exactly 1 string editing distance [Lev66] 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 data set. As plotted in Figure 10, 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. According to the criteria concluded in [Bur07], at least 1000 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 comprise 50% of the total, and 75% of the total can be categorized into emails of length ranging from 1 to 55. 99% of the total are emails of length ranging from 1 to 320.

In order to systematically test our approach, we designed two experiments: a controlled experiment and a stratified randomized sampling experiment. The first experiment is 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 is 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 multi-class 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., writing samples of each candidate author). We evaluate our approach with respect to the classification accuracy measure, which indicates the percentage of anonymous snippets that are correctly classified.

For all the experiments described below, 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.

5.2 Controlled Experiment

In this experiment, we sampled documents randomly multiple times under controlled conditions and systematically tested our approach with respect to its identification accuracy.

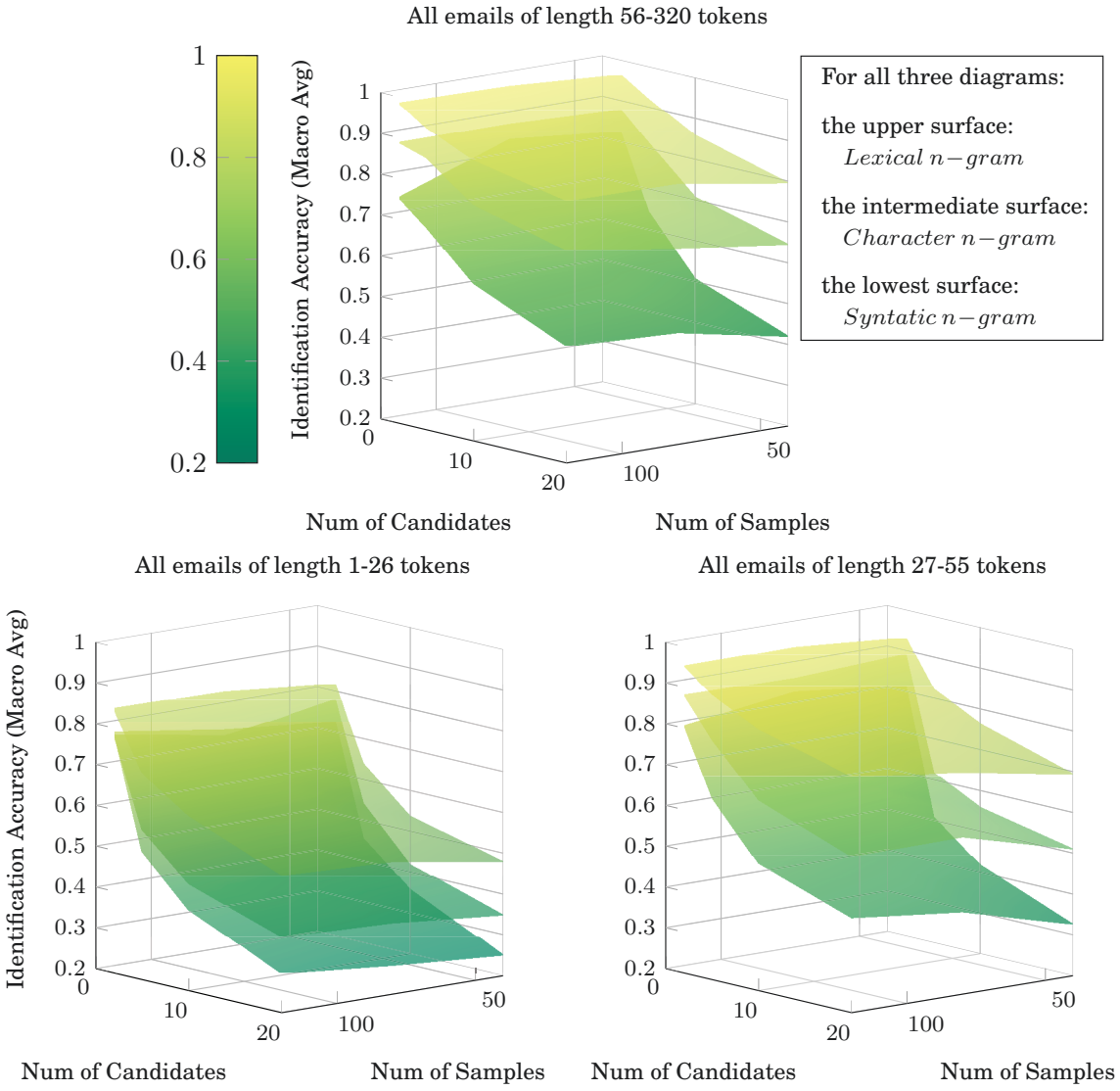

Figure 11: Performance comparison between isolated events. For all the diagrams, the upper surface is lexical n-gram event, the intermediate surface is character n-gram event and the lowest surface is POS n-gram event.

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 1-26 (50%), emails of length 27-55 (25%), and emails of length 56-320 (24%).
- For the size of samples for each author, we selected 20, 40, 80, and 120.
- For the size of candidate set, we chose the typical values: 2, 5, 10, 20.

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 [Sav12] to calculate the overall accuracy value in each round. Macro average accuracy is simply the average of all accuracy, where all the 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 in order to systematically test

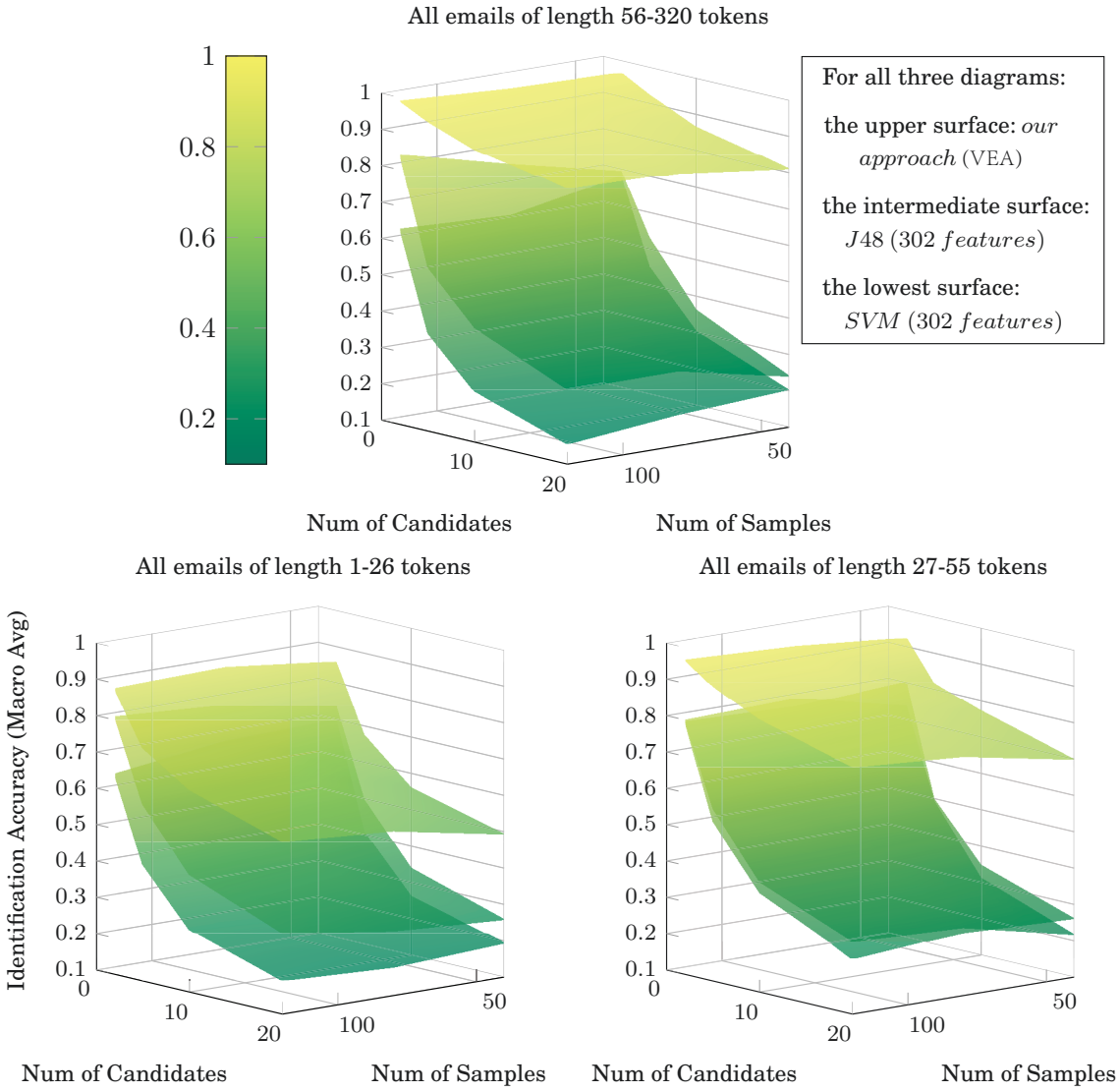

Figure 12: Performance comparison between approaches. For all the diagrams, the upper surface is VEA, the intermediate surface is the stylometric J48 and the intermediate surface is the stylometric SVM.

the difference between the events with respect to their identifying accuracy. The second one was conducted by employing the complete VEA approach in Chapter 4 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 11 shows the experimental result of test one, 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 the macro average accuracy under the given values of x and y. Also, the colour of the gradient surface indicates the accuracy value: the brighter the colour, the higher accuracy value the point has. For all three diagrams in this figure, the upper surface is the event for lexical n-gram, which means it achieves the best identifying accuracy across all given conditions, and the intermediate surface is the event character n-gram, also on the bottom, the lowest surface is for the event Part-Of-Speech n-gram.

The three diagrams in Figure 11 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 the given conditions, but it is significantly affected by the length of the given anonymous document, while the POS n-gram event appears to suffer less from this condition even though it achieves at most around 80% accuracy. Also, as the size of candidate increases, performance of the event Lexical n-gram appears to drop more slowly than the other two surfaces.

Table 6: Employed stylometric features.

Stylometry	Number of Features
lexical features	105
function words	150
punctuation marks	9
structural features	15
domain-specific features	13
gender-preferential features	10
total	302

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 how reliable this result is, a distinct model should be built for each linguistic modality. Also, when combining evidence from these modalities, they should be weighted accordingly.

Figure 12 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 302 stylometric features, as shown in Table 6. These features are used and discussed in [IBFD13]. To have a comparable result, we did not adopt any n-gram related dynamic feature. Two attribution techniques were selected: SVM and J48, which demonstrated the most comparable performance in [IBFD13]. We choose the libSVM [CL11] for SVM implementation and J48 decision tree implementation in weka[2].

As indicated in Figure 12, which is the same diagram representation used in Figure 11, our VEA approach consistently outperforms the other two typical approaches. Even though the given anonymous document is only of length 1-26, it can still achieve more than 85%

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

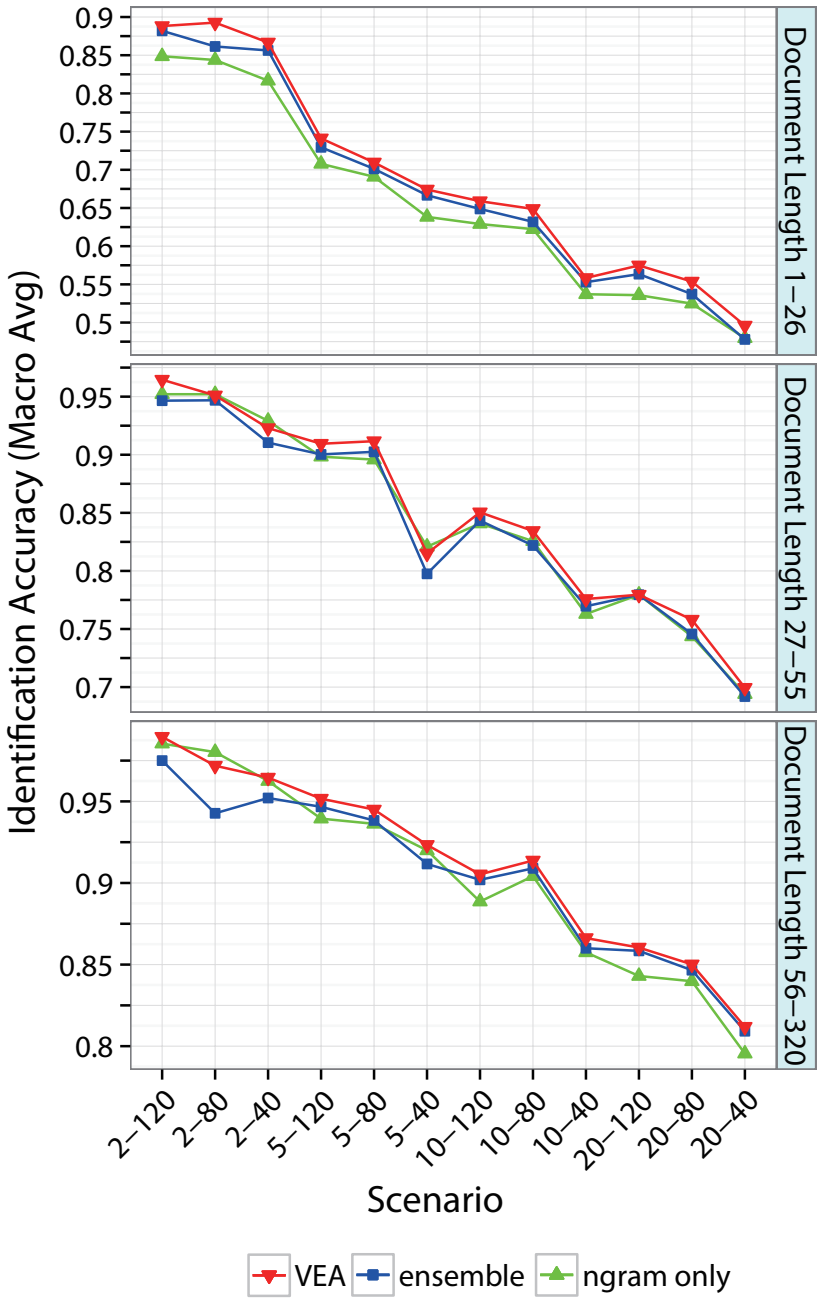

Figure 13: Performance comparison between VEA, voting ensemble, and lexical n-gram event. X axis indicates different scenario, for example *2-120* stands for a 2 candidates scenario with 120 writing samples for each of them. Y axis indicates the identification accuracy.

accuracy in a two-candidate scenario. Also, 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 a typical voting-based ensemble method and the lexical n-gram-event-only approach. The experimental result is shown in Figure 13. 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 the others in almost all cases, 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 3 scenarios.

5.3 Stratified Randomized Sampling Experiment

In this section, we describe the second experiment. In order 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 for each candidate depends on how many emails this candidate actually has 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.

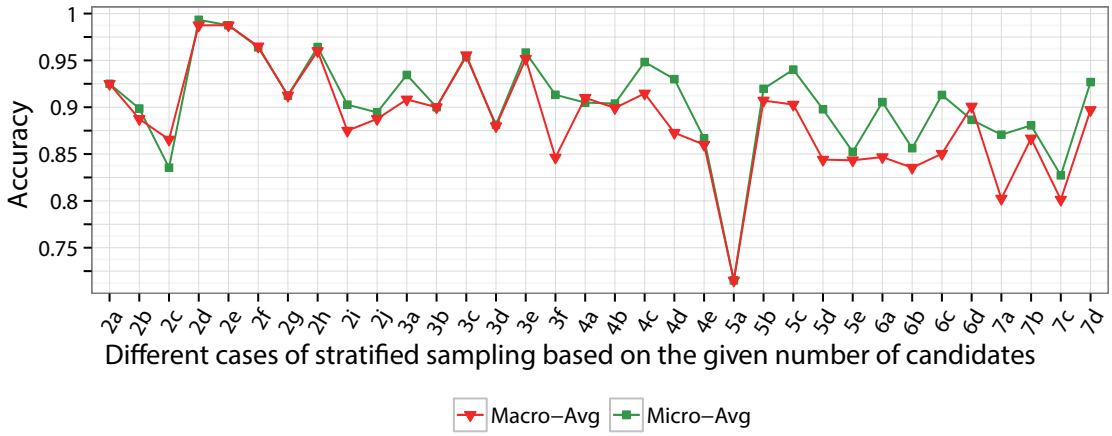

Figure 14: Performance of VEA on unbalanced-class problem.

Both the Macro Average and Micro Average accuracy measures are employed in this experiment. As mentioned above, 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 multi-class classification [Sav12]. 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 2-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 $(1/10 + 19/20)/ = 0.525$ but the Micro Average is $(1 + 19)/(10 + 20) = 0.667$.

The experimental result is shown in Figure 14. The labels on the x axis indicate the given scenario. For instance, '2a' means a stratified sampling on two random authors while '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

Table 7: 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			

class problems and achieve good identifying accuracy with respect to both Macro Average and Micro Average.

5.4 Confidence Estimation

In this section, we present our confidence estimation results. To verify how well our selected features can model the identification accuracy value, we first collected the input samples for building an estimation model from all previous runs of the VEA approach in the above experiments. Specifically, these samples were collected from Line 11 in Algorithm 3 based on the features in Table 7, along with their validation accuracy value. This test is to evaluate whether the features we selected can model the output accuracy value. The modeling result is in the first 6 rows in Table 7, which includes the estimated coefficients and the standard z-test for each coefficient. In this table, all the z values indicate that on our selected features all significantly affect the target accuracy attribute. Note that the gap statistic $dist_{m_r}$ [KSA06] does affect the prediction result but the distribution related features in scores (i.e., $socre_{avg}$, $score_{max}$) play relatively more important and stable roles.

Also, in order to verify whether our estimation model can actually predict the accuracy

value of the unseen data (unseen scenarios), we collect all the estimated confidence values from VEA in all of the above experimental runs. Specifically, these predicted values come from Line 22 in Algorithm 5. We also gather the corresponding actual accuracy value in the testing phase in all 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 and R^2 statistics are shown in the last row in Table 4. The MAE value indicates that on average our predicted confidence value has a 5% difference to the actual accuracy value, and the R^2, which closes to 1, indicates a very good prediction result.

Chapter 6

Conclusion and Future Work

In this thesis, we present our Visualizable Evidence-driven Approach (VEA) for the authorship attribution problem. To facilitate its interpretability and explainability, it is designed according to the EEDI (End-to-End Digital Investigation) framework and it is able to visualize and corroborate the linguistic evidence supporting our output attribution results. Also, we conducted comprehensive experiments to fully evaluate our VEA approach and have shown that it can achieve state-of-art authorship attribution accuracy. We have noticed the scalability issues of this method; when dealing with a scenario with more than 20 candidates, it is more suitable to identify a small subset of candidates using other scalable methods, and after that employ our method to construct cumulative visualized evidence. In general, the presented approach achieves a higher attribution accuracy than traditional stylometry, while its output is visualizable and presentable.

Future study for improving the VEA approach has the following directions:

- Include stylometric features (e.g., features in Table 1) other than the employed n-gram features into VEA approach as additional events. It is very possible for the VEA approach to achieve comparable or even better identification accuracy, since there is an increased amount of input information and VEA weights the events by considering their demonstrated discriminant power in the training set. Also, taking stylometric features other than n-gram into the model can reduce the chance that VEA approach models the themes rather than the actual writing style.

- Design a new hypothesis visualization scheme that is able to represent not only n-gram features but also other stylometric features (e.g., features in Table 1). Currently, VEA is only able to visualize n-gram features due to the limited hypothesis representation. It is possible to modify the representation to incorporate and visualize other features. For example, the feature $ratioofdigitstocharactercount$ in Table 1 can be possibly represented as digits in the anonymous snippet and then their highlighted colours represent the degree of proximity to corresponding candidate authors.

- To design authorship analysis solution, an other direction can be exploiting the power of language model. Recently, the research of language model has been developing in an increasing pace. The most promising one is the work from [MSC$^+$13] on learning vector representation of words. The learned vector space models the semantic relationship between words. As different people have varying writing styles and distinct individual vocabulary, the learned vector space describes different semantic relationship and it is possible to distinguish writing styles based on this learned model.

Authorship Attribution (AA) studies have a history longer than 120 years, while it is still not as reliable as other successful biological idiosyncrasies to be widely accepted by the public. Distinct from biological idiosyncrasy, the plain text data with its unstructured property and variant quality introduces larger fluctuation in individual style under changing scenarios. It is true that the studies on AA is calling for a widely acceptable systematic standard for conducting AA evaluational experiment, and that the purely quantitative measure of attribution result with poor presentability and interpretability is practically insufficient as evidentiary proof from the perspective of forensic science. We believe that more studies on Authorship Analysis considering these factors are needed to make AA techniques more reliable and practical.

Bibliography

[AC06] Ahmed Abbasi and Hsinchun Chen. Visualizing authorship for identification. In *Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI)*, volume 3975 of *Lecture Notes in Computer Science (LNCS)*. Springer, 2006.

[AC08] Ahmed Abbasi and Hsinchun Chen. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems (TOIS)*, 26(2), 2008.

[BAG12] Michael Brennan, Sadia Afroz, and Rachel Greenstadt. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Transactions on Information and System Security (TISSEC)*, 15(3), 2012.

[BKW12] Seymour Bosworth, Michel E. Kabay, and Eric Whyne. *Computer Security Handbook*. Wiley, 2012.

[Bur07] John Burrows. All the way through: testing for authorship in different frequency strata. *Literary and Linguistic Computing*, 22(1), 2007.

[CL11] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), 2011.

[ÇLB12] Tantek Çelik, Chris Lilley, and L David Baron. Css color module level 3. 2012.

[CRS+12] Marco Cristani, Giorgio Roffo, Cristina Segalin, Loris Bazzani, Alessandro Vinciarelli, and Vittorio Murino. Conversationally-inspired stylometric features for authorship attribution in instant messaging. In *Proceedings of the 20th ACM International Conference on Multimedia*. ACM, 2012.

[CS11] Dan Clark and Jeff Sanders. *Beginning C# object-oriented programming*. Springer, 2011.

[Dae13] Walter Daelemans. Explanation in computational stylometry. In *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, volume 7817 of *Lecture Notes in Computer Science (LNCS)*. Springer, 2013.

[DK05] Kai-Bo Duan and S Sathiya Keerthi. Which is the best multiclass svm method? an empirical study. In *Proceedings of the 6th International Workshop on Multiple Classifier Systems (MCS)*, volume 3541 of *Lecture Notes in Computer Science (LNCS)*. Springer, 2005.

[ESMy11] Hugo Jair Escalante, Thamar Solorio, and Montes-y-Gómez. Local histograms of character ngrams for authorship attribution. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2011.

[FWE03] Benjamin C.M. Fung, Ke Wang, and Martin Ester. Hierarchical document clustering using frequent itemsets. In *Proceedings of the 3rd SIAM International Conference on Data Mining (SDM)*. SIAM, 2003.

[Hal07] Hans Van Halteren. Author verification by linguistic profiling: An exploration of the parameter space. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1), 2007.

[HB03] Mark Harrower and Cynthia A Brewer. Colorbrewer.org: an online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1), 2003.

[Hol94] David I Holmes. Authorship attribution. *Computers and the Humanities*, 28(2), 1994.

[Hol98] David I Holmes. The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing*, 13(3), 1998.

[HS06] John Houvardas and Efstathios Stamatatos. N-gram feature selection for authorship identification. In *Proceedings of the 12th International Conference*

on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA), volume 4183 of *Lecture Notes in Computer Science (LNCS)*. Springer, 2006.

[HS11] Steffen Hedegaard and Jakob Grue Simonsen. Lost in translation: Authorship attribution using frame semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2011.

[IBFD10] Farkhund Iqbal, Hamad Binsalleeh, Benjamin Fung, and Mourad Debbabi. Mining writeprints from anonymous e-mails for forensic investigation. *Digital Investigation*, 7(1), 2010.

[IBFD13] Farkhund Iqbal, Hamad Binsalleeh, Benjamin Fung, and Mourad Debbabi. A unified data mining solution for authorship analysis in anonymous textual communications. *Information Sciences*, 231, 2013.

[Juo06] Patrick Juola. Authorship attribution. *Foundations and Trends in Information Retrieval*, 1(3), 2006.

[Juo12] Patrick Juola. Detecting stylistic deception. In *Proceedings of the Workshop on Computational Approaches to Deception Detection*. Association for Computational Linguistics, 2012.

[JV10] Patrick Juola and Darren Vescovi. Empirical evaluation of authorship obfuscation using jgaap. In *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security*. ACM, 2010.

[KG06] Gary Kacmarcik and Michael Gamon. Obfuscating document stylometry to preserve author anonymity. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*. Association for Computational Linguistics, 2006.

[KHDM98] Josef Kittler, Mohamad Hatef, Robert PW Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3), 1998.

[KKW+11] Sangkyum Kim, Hyungsul Kim, Tim Weninger, Jiawei Han, and Hyun Duk Kim. 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, 2011.

[KLM+97] Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Lopes, Jean-Marc Loingtier, and John Irwin. *Aspect-oriented programming*. Springer, 1997.

[KS11] Ioannis Kourtis and Efstathios Stamatatos. 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)*, 2011.

[KSA06] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 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, 2006.

[KSA11] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Authorship attribution in the wild. *Language Resources and Evaluation*, 45(1), 2011.

[KSAW12] Moshe Koppel, Jonathan Schler, Shlomo Argamon, and Yaron Winter. The "fundamental problem" of authorship attribution. *English Studies*, 93(3), 2012.

[LD11] Kim Luyckx and Walter Daelemans. The effect of author set size and data size in authorship attribution. *Literary and Linguistic Computing*, 26(1), 2011.

[Lev66] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10, 1966.

[LV09] Maarten Lambers and Cor J Veenman. Forensic authorship attribution using compression distances to prototypes. In *Proceedings of the 3rd International Workshop on Computational Forensics (IWCF)*, volume 5718 of *Lecture Notes in Computer Science (LNCS)*. Springer, 2009.

[LWD12] Robert Layton, Paul Andrew Watters, and Richard Dazeley. Recentred local profiles for authorship attribution. *Natural Language Engineering*, 18(3), 2012.

[LWD13] Robert Layton, Paul Andrew Watters, and Richard Dazeley. Automated unsupervised authorship analysis using evidence accumulation clustering. *Natural Language Engineering*, 19(1), 2013.

[MFJP09] Justin Martineau, Tim Finin, Anupam Joshi, and Shamit Patel. Improving binary classification on text problems using differential word features. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. ACM, 2009.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. pages 3111–3119, 2013.

[MW64] Frederick Mosteller and David Wallace. *Inference and disputed authorship: The Federalist*. Addison-Wesley series in behavioural science. Addison-Wesley, 1964.

[NPG+12] Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. On the feasibility of internet-scale author identification. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP)*, 2012.

[PSWK03] Fuchun Peng, Dale Schuurmans, Shaojun Wang, and Vlado Keselj. Language independent authorship attribution with character level n-grams. In *Proceedings of the 10th conference on European chapter of the Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics,

2003.

[RKM10] Sindhu Raghavan, Adriana Kovashka, and Raymond Mooney. Authorship attribution using probabilistic context-free grammars. In *Proceedings of the ACL 2010 Conference Short Papers*. Association for Computational Linguistics, 2010.

[SA04] Jitesh Shetty and Jafar Adibi. The enron email dataset database schema and brief statistical report. *Information Sciences Institute Technical Report, University of Southern California*, 4, 2004.

[Sav12] Jacques Savoy. Authorship attribution based on specific vocabulary. *ACM Transactions on Information Systems (TOIS)*, 30(2), 2012.

[SB88] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 1988.

[SBZ12] Yanir Seroussi, Fabian Bohnert, and Ingrid Zukerman. Authorship attribution with author-aware topic models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, 2012.

[SG06] Conrad Sanderson and Simon Guenter. 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*. Association for Computational Linguistics, 2006.

[Sol13] Lawrence M Solan. Intuition versus algorithm: The case of forensic authorship attribution. *Brooklyn Journal of Law and Policy*, 21(551), 2013.

[SPRMy11] Thamar Solorio, Sangita Pillay, Sindhu Raghavan, and Manuel Montes-y-Gómez. Modality specific meta features for authorship attribution in web forum posts. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, 2011.

[SSMyR13] Upendra Sapkota, Thamar Solorio, Manuel Montes-y-Gómez, and Paolo Rosso. The use of orthogonal similarity relations in the prediction of authorship. In *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, volume 7817 of *Lecture Notes in Computer Science (LNCS)*. Springer Berlin Heidelberg, 2013.

[Sta09] Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3), 2009.

[SVS+13] Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. Syntactic dependency-based n-grams: More evidence of usefulness in classification. In *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, volume 7816 of *Lecture Notes in Computer Science (LNCS)*. Springer, 2013.

[SZB11] Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. Authorship attribution with latent dirichlet allocation. In *Proceedings of the 15th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2011.

[TSH96] Fiona J Tweedie, Sameer Singh, and David I Holmes. Neural network applications in stylometry: The federalist papers. *Computers and the Humanities*, 30(1), 1996.

[YP97] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, 1997.

[ZLCH06] Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology*, 57(3), 2006.

[ZM98] Justin Zobel and Alistair Moffat. Exploring the similarity space. 32(1), 1998.