Universiteit
Leiden

Master Computer Science

Assessing the interpretability of word-based authorship verification methods for forensic applications

Name:	Sophie van der Bliek
Student ID:	s2380243
Date:	20/07/2024
Specialisation:	Data Science
1st supervisor:	Suzan Verberne
2nd supervisor:	Niki van Stein

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Dedication

Dedicated to the loving memory of Matthew Wilson Loughridge
1995 - 2023

I guess I am now the nerdiest between the two of us. I hope I made you proud.

Acknowledgements

At the risk of sounding as if I am accepting an Oscars award, I want to thank the people around me. I knew this program would be a challenge, but it ended up being more difficult than I imagined. After many hardships, I came out the other end. This would not have happened if I had to do it all by myself. I am grateful to those who helped me, and I would like to give them credit.

I would like to thank my friends and family for supporting me, listening to my endless frustrations and speaking words of encouragement all the way, for all those years. To my parents for helping me finance this masters, and especially for my mother for always being there for me and believing in my abilities despite what I told myself. Thank you for instilling your can-do attitude in me, and not ever allowing me to be given ”a certificate of incapacity”, in your words. I would also like to thank my best friends for supporting me through this five-year masters ordeal, who probably still have no idea what I study or what this thesis is about.

I would like to thank my supervisors, Suzan and Niki, for providing me with structure, excellent and quick feedback, for your patience and for providing me with a second chance to finally get through this last part of my masters degree. It means a lot to me you put so much time and effort into supervising this project to ensure I could finally get it done.

I would like to thank my friends Jasper and Bartosz in particular, for helping me when I was completely stuck. You are both some of the smartest people I know and really admire, and I would not have been where I am today without your help. Sharing your knowledge and enthusiasm led me to pursue this field, and your unwavering support and patience has gotten me in this programme, and through it. I will never forget everything you did to help me, and hope I can someday repay the favour somehow.

Lastly, I would like to thank Alexandra Blank for pulling me from the trenches and motivating me to keep going and not give up. Your empathy and ability to pull someone from a dark place is a gift that not many people possess. I hope you know that I cherish the support you gave me, and I hope to follow in your footsteps with my own students.

> *Put people in a position to*
> *succeed, and they just might.*
>
> -Kelsey Hightower

Contents

Abstract

Authorship verification aims to determine whether two pieces of text are written by the same author or not, which can be analyzed through their writing style. Authorship analysis is especially of interest to forensic linguists, who can use these methods to analyze evidence and present forensic analysis in court. Although there has been an increase in using machine learning for research in this field, many developed machine learning methods are unsuitable for forensic analysis. This is mostly due to both their complexity and corresponding lack of interpretability, and the experiments being conducted on data that is not forensically feasible. Transparent and shallow machine learning methods using word-based and descriptive features using short minimally pre-processed text achieved an F1 of 0.85 and AUC of 0.84 in this experimental set up. Despite the limitations of this experimental set-up, the results suggest that a transparent model could potentially be useful as a tool to quantify some parts of analysis for forensic linguists, rather than fully replace forensic linguistic experts. In order to test these methods more thoroughly, the models should be tested on real forensic data and evaluated by end users.

1 Introduction

Authorship verification is a subfield of applied linguistics that aims to analyze whether two pieces of text were written by the same person. In forensics, there can be several cases where determining authorship of a text can aid in the prosecutive process, such as threats, suicide notes, or text messages [33]. Forensic linguistics and authorship analysis has played a big part in several high-profile cases, such as the Hummert “stalker/serial killer” case, the Coleman triple homicide case, and the case of Ted Kaczynski, known as the Unabomber [28].

Kaczynski successfully evaded the law authorities for almost two decades, during which he sent untraceable bombs to several targets, killing three and injuring nearly two dozen more. In 1995, seventeen years after he sent his first bomb, he sent a 35,000 word manifesto to the FBI explaining his motives and views of modern society. After his manifesto was published, his estranged brother and his wife recognized his style of writing and provided the authorities with letters from Kaczynski. Linguistic analysis determined that the author of those letters and the manifesto were highly likely to be the same author. This break, together with other evidence, led to a search warrant, which led to the arrest of Kaczynski in April 1996 [1].

Today, most communication is electronic, and forum posts, blogs, and texts may even be anonymously posted from online accounts or 'handles' that do not reflect the author's true identity [33]. Establishing authorship is becoming more difficult, but also more relevant than ever, especially in forensic applications. In the past two decades, research has increased in using computational applications of linguistic analysis, including machine learning to determine authorship. Koppel & Winter [26] developed the impostor's method, and Potha & Stamatos [35] the profile-based method, which inspired many other variations in later years. In 2013, the first compression method was developed by Veenman & Li [41], using only a compression algorithm and a threshold to determine authorship. Halvani [18] also developed compression methods, as did Hürlimann et al. [23]. In more recent years, deep learning models have been dominating many subfields of natural language processing (NLP), such as the work of Bagnal [3], using a Recurrent Neural Network (NN) to identify authorship using character n-grams, and Boeninghoff et al [7], where they used an attention-based similarity network for explainable authorship.

However, applying authorship analysis to forensic cases comes with several constraints. First, almost all of the aforementioned methods were applied to long, clean text with proper grammar and spelling such as essays, novels and papers, ensuring a lot of text for the model to train on to detect linguistic patterns. This text is often also pre-processed to optimize model performance, as punctuation and irregularities will introduce noise, which often compromises model performance [14]. Whereas realistically, forensic text will be brief, contain non-standardized grammar and spelling and other irregular use of language, which are actually powerful discriminative features in forensic linguistics [12]. Second, many of these models suffer from a lack of interpretability. When forensic evidence is presented to the court in a criminal case, the method with which the evidence was obtained must be transparent and easy to interpret for the members of the court, such that they can evaluate the value of this evidence. Although forensic experts go through extensive training and their expertise is often taken at face value in court [12], the methods themselves should be understandable to someone who might be facing a lifechanging sentence, and for those who convict them [2]. The research on explainable AV applicable to forensics is sparse, likely due to the difficult trade-off between good model performance and being constrained by the strict requirements of valid forensic evidence.

Therefore, the aim of this thesis is to provide an overview of currently existing methods in AV, and evaluate the challenges of several methods from both the perspective of a forensic expert and the perspective of a machine learning expert. Then, an experiment of authorship verification will be run with lexical features, using widely known and well-understood models Logistic Regression, Support Vector Machines (SVM), and XGBoost on short text using microblogging data (tweets) with minimal preprocessing of text. An example use case will be presented of how a forensic linguist may use these models to analyze lexical features in text, using coefficients from Logistic Regression. The experiments in this thesis show that relatively simple and easily interpretable methods such as Logistic Regression can obtain an F1 of 0.85 and ROC-AUC of 0.84 using unigrams, bigrams and descriptive features on this dataset. The strongest coefficients of the models can be used to determine to which degree specific features contributed to a same-author or different-author classification, making the results directly interpretable. Considering the strict standards of evidence for forensic methods and the many forms of linguistic analysis that may be applicable to a case, these methods might be used as complementary part of forensic linguistic analysis in cases where it might be of use, rather than working towards a method that might replace a forensic linguist entirely.

The following research questions are addressed in this thesis:

- What is the current state of authorship verification research, focusing on research that may be applicable to forensics?
- What are the challenges in AV regarding forensic applicability?
- Can word-based simple machine learning methods and their coefficients using short minimally pre-processed text form an effective yet interpretable method?
- How might the outcome of this experiment be useful to a non-technical end user, i.e., a forensic linguist, and to which degree?

2 Authorship Verification

Authorship attribution (AA) can be described as a single-label multi-class text categorization task. Given a text with unknown authorship and sample texts written by a set of candidate authors, the objective is to determine which of the candidate authors, if any, is most likely the author of the text under investigation [8]. This is usually determined by their similarities in writing style (e.g. word use, vocabulary range, punctuation, and so on). Determining personal style is a difficult task, especially because writing habits are easily influenced by the particular topic or genre of the text. For example, one may speak about very different topics with their families than with their colleagues, and stylistic elements such as word choice may differ greatly in a formal email as opposed to texting [33]. This distinction is essential when evaluating methods for an AA task, as results on similar-topic or similar-genre can often not be extrapolated for cross-topic or cross-genre tasks.

Authorship verification (AV) can be described as a specific subset problem of open-set authorship attribution, where the objective is to determine whether, given a text under investigation and sample texts by one author, these texts were written by that same author or not [36]. Therefore, this can be seen as a one-class classification problem, since the outcome is binary (i.e. written by the same author, or not written by the same author) [16]. AV, as opposed to AA, is a relatively young and interdisciplinary field, but has received considerable attention in the past years. PAN at CLEF organizes a challenge with different tasks every year related to digital text forensics and stylometry, including authorship verification, authorship attribution, style change detection, and sentiment analysis. These shared tasks have resulted in increased attention and significant scientific contributions to these fields [15]. Many new AV approaches have been proposed in the context of the PAN challenge, such as using compression for AV by Veenman & Li [41]. The first deep learning model was proposed in 2015 by Bagnall [3] for the author identification challenge. Hernández-Castañeda & Calvo [20] proposed a method using a semantic space model through Latent Dirichlet Allocation (LDA) in 2017 using data from PAN 2014 and 2015.

Authorship verification is especially interesting for forensic applications. For example, AV can be useful to determine the ownership of a phone with incriminating evidence on it, by comparing texts on a phone of unknown ownership with the texts from a suspect whose phone has been seized for investigation [5]. However, real-world forensic cases pose another layer of difficulty concerning the length of the text, as the amount of data seized is often low to very low. Especially for complex modeling techniques, having little data available may severely impact the reliability and applicability of machine learning analysis. For applications such as forensics, reliability and validity of the method is crucial in order to adhere to standards of admissible evidence.

2.1 Definitions of authorship verification

Authorship verification can be approached as both a similarity detection and a classification problem. Given an unknown document D_u and a known document D_a, the task is to determine whether both documents were written by the same author, A, focusing on the writing style rather than the topic or genre [16]. Applications of authorship verification range from plagiarism detection to forensic analysis [19].
AV methods can be formally identified into three categories, as described by Halvani et. al. [16]:
Unary AV determines the classification model solely on the basis of D_a, assuming D_u to be written by author A if D_u is stylistically similar to the documents in D_a.

Binary-intrinsic AV methods build a classification model using a training corpus that includes verification cases labeled with a ground truth as either Y (same-author) or N (different-author). Methods in this category treats known and unknown documents as a single unit X (such as a feature vector). If X is more similar to Y, accept A as the author of D_u, if X is more similar to N, assume the author is not A.

Binary-extrinsic method determine classification on the basis of external documents (e.g. from a search engine), and is often referred to as the *Impostor Method* (IM). The remainder of the method is much like the binary-instrinsic method. The document of D_a are Y, and the impostor documents as N. D_u is assumed to be written by A if it is most similar to documents in D_a. If D_u is more similar to the impostor document, A is rejected as the true author of D_u. The experiments in this thesis will be conducted with binary-intrinsic methods, as we are using a classification model trained on ground-truth labeled data, using feature vectors and a distance measure to determine authorship.

2.2 AV in forensic settings

According to Chaski [12], forensic linguistics can provide answers to four categories of inquiry in legal settings:

1. Identification of author, language, or speaker;
2. Intertextuality, the referencing or alluding to another text that might be of impact on the original text such as reference to film, books, or music;
3. Classifying the text type such as threats;
4. Linguistic profiling of an author to assess their demographic background (i.e., age, gender, education level).

In forensic investigations, courts often rely on forensic linguists to conduct authorship analysis. They apply linguistic knowledge to examine documents under investigation, such as blackmail letters, other types of threats, emails and chats. Their analyses are based on domain expertise, and the result is usually presented in a report. Legal standards for evidence dictate their admissibility in court and scientific standards for research. In both common law systems and civil law systems, this includes aspects such as the expert being able to demonstrate an error rate, the technique being empirically tested and being falsifiable and refutable, and straightforwardness, i.e. the technique must be explained clearly enough to be understood by the court and/or jury [12].

Especially in the past, forensic linguistics also incorporated processes such as handwriting extracted from notes, diaries and letters. This type of text required both graphic and stylistic analysis, and the interaction of the two. Nowadays, as commucation is often typed rather than handwritten, the focus has shifted to writing style rather than handwriting style [33]. Style markers are often based on stylometric features, which can be categorized into several different groups, as defined by Stamatatos [38]:

Lexical features: Lexical features pertain to an author's use of vocabulary, which can range from vocabulary richness to word frequencies or word n-grams, to errors and token-based quantifiable entities such as word length or sentence length. A significant advantage is that these techniques are largely language-independent before the model is trained, after which the model becomes specific to that language. Lexical features also simply require tokenization of a text, i.e. separating words or sentence by spaces, periods or other punctuation.

Character features: Character features refer to the type of characters an author uses, such as alphabetic or numeric character counts, uppercase or lowercase characters, punctuation mark counts or character n-grams. Compression methods also fall under this category.

Semantic features: Semantic features refer to the underlying meaning and thematic essence of words and sentences used by the author. This can be the analysis of synonyms and contextual use of language, which can be uncovered with NLP with part-of-speech tagging, text chunking and semantic parsing. For AV, identifying words with low semantic content such as pronouns, articles, and prepositions are of more interest than identifying words with high semantic content such as nouns, verbs, and adjectives [2]. The latter often depends on the topic of the text, and are less useful to identify an author. If one is speaking about their vacation in one text, and describing details of their work in another, it is unlikely they will be using a lot of the same nouns in both texts. This presents a problem when the documents under investigation are cross-topic. However, studies have shown authors often use the same low semantic content-words, regardless of what they are writing about. [2].

Syntactic features: Syntactic features refer to the structural and grammatical composition of sentences, and patterns such as passive versus active voice. Use of function words that serve little lexical significance, can play a crucial role in the grammatical framework of sentences. The analysis of these words, due to their frequent yet subconscious use by authors can be a potential indicator of authorship.

Application-specific features: These features are tailored to the particular domain of the text that is analysed. For social media content, this might be incorporating the use of emojis, hashtags, mentions and internet-specific words and abbreviations for analysis.

Function words: Another feature that is often used are function words. These are commonly used words (e.g. “and”, “the”, “of”) that have little lexical meaning but serve a grammatical purpose. The frequency and distribution of these words can be a powerful indicator of authorship, as these are often used subconsciously.

Patterns of punctuation, spelling and spacing can reflect an author’s idiolect. Style markers can be assess qualitatively (how unusual they seem), or quantitatively (how many times they occur in the text) [2]. The analysis of common word usage in texts for authorship determination has been widely studied, with emerging consensus that it is both accurate and precise. Prior research has shown that the one hundred most frequently used words can be used to effectively distinguish between authors [2].

Forensic science is subject to much scrutiny, as studies have shown up to 20% of erroneously convictions due to admission of inaccurate or unreliable forensic evidence. Science-based evidence is crucial for a fair justice system, and forensic expert testimony should rely on replicable, well-researched, and transparent methods that help resolve issues in litigation that cannot be proven otherwise [2]. Chaski [12] describes this phenomenon as the problem of expert testimony as a “hired gun”, where the expert runs a method in whatever way to obtain a desired result, rather than following established protocol. Especially in countries that follow a civil law system, where the court appoints an expert and the court itself is involved in uncovering the facts about a case, the government is responsible for the quality of the admitted evidence.

Chaski [12] outlines best practices for admissible forensic linguistic evidence in court, following scientific standard for research. She describes acceptable methods for AV, such as development of methods and testing for accuracy of said methods independent of litigation, that are tested on ground truth data (where the correct answer is known), that is tested for known limits to specific accuracy levels, and that these methods are able to work reliably on forensically feasible data. She notes that many AV methods are tested on 'clean' texts such as novels or essays, which are fundamentally different from data that is typically admitted in a forensic case. Clean data can not function as a proper ground truth, as forensic text is often anything but clean – it is often cross-topic, cross-genre, and will most likely contain spelling mistakes and be ungrammatically structured. She also points out that the brevity of textual data is an inherent challenge in forensic authorship verification, one that cannot be addressed by research focused on texts containing tens or hundreds of thousands of words in a single document. Furthermore, text that is pre-processed such as eliminating extra spaces, standardizing words (thus not taking into account the misspelling of words), which are often key markers that are useful to determine authorship. Chaski [12] defines forensically feasible data as "documents which are short, in several types of genre and register, and without any correction to spelling, grammar or prescriptive conventions about writing". Above all, both Ainsworth & Juola [2] and Chaski [12] note that forensic evidence must be replicable, related to generally accepted and well-understood techniques, and able to be interpreted by laymen (i.e. those without technical background).

2.3 Explainability

As models become more complex, especially in subfields where the problem is not always easily quantifiable such as language, it also becomes increasingly important in high-stakes applications to know how a model came to a certain performance or decision. As a result, researchers have been focusing more on explainable artificial intelligence (XAI) in recent years.

2.3.1 Definitions and relevance

Although there is currently no established technical definition, Barredo Arrieta et al. [4] define XAI as "given an audience, an explainable AI is one that produces details or reasons to make its functioning clear or easy to understand". Terms such as interpretability, transparency and explainability are often used interchangeably, but there are subtle differences. Whereas interpretability is more related to the ability to explain or provide meaning in understandable terms to a human, transparency is rather related to the characteristics of the model itself. Certain models, such as logistic regression or decision trees, are transparent because the model itself is understandable. External XAI techniques however, can be used for models that are intrinsically more opaque such as Support Vector Machines (SVMs), ensemble models or deep learning models to communicate understandable information of how the predictions were produced considering the input [4]. Barredo Arrieta et al. [4] distinguish between shallow ML models, i.e. supervised models that do not have layered neural processing structures such as SVMs and tree ensembles, from deep models. Generally, explainability approaches can also be divided into global explainability and local explainability. Global explanations address the interpretation of the behaviour of the model as a whole, whereas local explanations aim to explain the output regarding a specific instance [6].

Understanding how and why a model makes certain predictions is essential to building user trust in high-stakes applications such as healthcare, legal applications and fraud detection. Besides trustworthiness, explainability is also key in informativeness regarding decisionmaking, confidence in models used in specific applications, and fairness regarding ethical use of models, among others. These considerations will soon be required for machine learning systems in the EU with newly introduced regulatory frameworks such as the EU Artifical Intelligence Act, which came into force August 1st 2024. The EU AI Act is the first comprehensive AI law and regulatory framework to ensure that companies and governmental agencies adhere to strict standards regarding the transparency of their systems, especially in high-risk applications such as fraud detection and automated screening tools that may lead to discrimination [27]. As models become more advanced, such as complex deep learning systems, the inner workings of the model become more difficult to understand for both experts and end users with little technical knowledge. If unchecked, this can result to models being implemented without fully understanding their inner mechanism and possible limitations, which can lead to potentially catastrophic consequences such as racial profiling in automated decision systems such as credit scoring, or the childcare benefit scandal in the Netherlands which led to the collective resignation of the government in 2021 [34].

Barredo Arrieta et al. [4] list three components for transparent models, i) simulability, or the extent to which a model can be simulated strictly by a human, ii) decomposability, the ability to explain input, parameter and calculation of the model, and iii) algorithmic transparency, which mostly considers the user's ability to comprehend how a model processes input to produce a specific output, and requires that the model's internal working are fully analyzable through mathematical methods. Of the latter, a linear model such as linear or logistic regression is considered transparent because its error surface is comprehensible, which enables end users to understand how the model produced a certain result. This contrasts with complex deep architectures, where the loss landscape can be opaque and the solution often relies on heuristic optimization methods like Stochastic Gradient Descent, making it difficult to comprehend the decision-making process [4].

Post-hoc explainability methods can be model-agnostic or model-specific. Model-agnostic methods can be used for any model, regardless of its inner processing to uncover information about its predictions. Feature relevance estimation, visualization and explanations by simplification are popular choices, whereof the latter, Local Interpretable Model-Agnostic Explanations (LIME) is arguably one of the most well-known. LIME aims to explain predictions for individual instances by building an interpretable model around perturbed samples and the corresponding predictions from an opaque model [37]. An example of this could be to build a linear regression model for a specific instance, and measure how close the explanation is to the predictions of XGBoost. Another well-known technique for local explanations of feature relevance explanation, is SHapley Additive exPlanations (SHAP). SHAP aims to explain the functioning of a black-box model by ranking the influence of each feature on the prediction output [30].

Besides the aforementioned methods, there are several other methods specifically developed that are used for post-hoc explanations for deep learning, including model simplification methods, local explanations and feature relevance. A notable technique in NLP research has been the use of attention mechanisms, which was initially introduced for increasing performance of the encoder-decoder model in machine translation. Attention mechanisms use a weighted combination of all encoded inputs vectors, where the most relevant vectors are ascribed the highest weights determined by the model [6]. These weights can be visualized to show which words or phrases the model focused on, such as showing which words such as “excellent” or “bad” contributed to a decision in sentiment analysis. However, the application of attention mechanisms as an explainability method is a matter of debate, which will be further discussed in chapter 3.3.1.

2.3.2 Challenges in explainability

Due to the complexity and lack of transparency in these models, explainability is especially challenging when it comes to providing accessible explanations for non-expert audiences such as policy makers and end users. Deep learning is often favoured by researchers due to their performance, but the trade-off between interpretability and performance is becoming increasingly important when these models are deployed and have to adhere to interpretability standards. Although deep learning models are often able to approximate more complex functions compared to more transparent and simpler models, more complex is not necessarily always better. Well-defined problems with appropriate constraints, such as in certain parts of industry with well-defined physically controlled environments, can benefit from a more inflexible model that models the distribution well enough without added complexity [4]. Furthermore, if the model requires interpretation by a human to ensure validity before it can be deployed for its intended task, the level of complexity necessary for a well-performing model may defeat the purpose of using a machine learning model entirely.

Explanations can be i) faithful, referring to how close the explanation is to the inner workings of the model, or (ii plausible, whether the explanation is understandable and satisfactory to human users [32]. In this chapter, the focus has been faithfulness, which is often the focus for a data scientist or someone with a technical background. Much of the discussion around explainability in technical papers mainly focuses on faithfulness as well – as long as the mechanism is clear to ‘technical people’ (i.e. researchers), that is good enough for them. However, the end user is not rarely someone without a technical background at all. In the case of forensic linguistics, even if we accept that attention mechanisms are faithful explanations to those with technical training, how plausible are concepts such as attention mechanisms in bidirectional LSTMs to end users such as lawyers, judges and defendants? Even if the model works very well, how desirable is it that judges must trust a data scientist, someone who conversely likely has very little knowledge of how and why legal frameworks are set up, fully in their expertise? Even if a forensic expert claims that attention mechanisms are a faithful explanation to them, it is hardly as easily interpreted as a coefficient from a transparent model. As noted previously, mechanisms regarding explainability in deep learning applications are still topic of debate.

As Chaski [12] noted, it is not desirable that an expert becomes a ‘hired gun’, and it is favourable to develop systems that have at least some degree of transparency to all end users and that have been verified and well-understood by several experts in that domain. As shown in cases such as the Dutch benefits scandal, leaning too much on expertise of domain experts can lead to exploitation of loopholes in the law, but it does not absolve the end users, companies or – in this case – the government, of responsibility and handling of consequences. The estimated cost of reparations of the benefits scandal is, at the moment of writing, at least 7.2 billion euros and is expected to increase [34].

3 Related Work

Several methods of approaching AV have been proposed, and there are different categorizations as proposed by Halvani [19] and Stamatatos [39]. However, for the sake of brevity, the foundation regarding the type of models used of these methods can be roughly divided into three types: i) 'shallow' ML methods (i.e. lacking deep architectures), which make use of well-known ML models such as SVMs or tree ensembles, ii) compression methods, and iii) deep learning models. Almost all of these methods use stylometric features such as the ones outlined in chapter 2.2 in some way to determine authorship. Several explainable AV methods will be discussed in section 3.4.

3.1 Shallow machine learning methods

Hernández-Castañeda & Calvo [20] proposed a method using a semantic space model through Latent Dirichlet Allocation (LDA) in 2017 using data from PAN 2014 and 2015, in which documents are presented a mix of topics. Brocardo et al. [11] used stylometric features with their own supervised learning technique based on thresholds combined with n-gram analysis for short text. Using the Enron dataset, they achieved an EER of 14.35% for message blocks of just 500 characters. The authors state n-gram features are considered noise tolerant and effective, which is especially useful for unstructured short text such as email.

3.1.1 Impostors method

The best-performing AV methods of the PAN competition of 2013 and 2014 are based on the Impostor's Method (IM), where the method was initially proposed by Koppel & Winter [26]. IM requires two stages, (1) gather documents according to a data collection procedure that serve as impostors, (2) use feature randomizations to iteratively calculate the cosine similarity between pairs of documents. If a suspect is selected from the among the impostor set above a set certainty, then the suspect is marked as the author of D_a. Potha & Stamatatos [39] introduced an improved version of the impostors method that focuses on better selection of impostor documents by choosing impostors that are more representative and challenging, and enhanced comparison techniques. Although IM is considered a succesful method, the method has trouble distinguishing same-author and different-author pairs if they differ in genre and topic. IM is also an extrinsic method (i.e. it requires external documents for construction of impostors), meaning that it must be certain that the impostor documents are not authored by either of the individuals involved in the verification task. Another disadvantage is that the method is computationally expensive [17].

3.1.2 Profile-based method

The profile-based method was introduced by Potha & Stamatatos [35] in 2014, that cumulatively considers all samples by one author for verification rather than instance-based approaches where all samples from an author are considered individually. In the profile-based method, all documents of a single author are combined to create a single profile, which is then used to verify the authorship of an unknown document. This approach uses cumulative textual features from multiple documents to create a general representation of the author's writing style. Using the 2013 PAN corpora, consisting of various texts written by different authors across different genres and languages, they achieved one of the highest F1 scores with the exception of the

English corpus. Stamatatos [40] later used an improved version of the profile-based method to conduct authorship analysis in a setting with varying topics.

3.2 Compression Methods

Compression-based AV methods have its roots in information theory, using a compression distance method and classifier to discern between authors. Veenman & Li [41] proposed compression-based methods for the PAN challenge in 2013, following a supervised learning paradigm. They used 1000-word documents from engineering textbooks by 46 authors, resulting in 2-75 documents per textbook. The first method uses a 1-nearest neighbour approach and obtains a 10% error rate. The second method uses a two-class classifier and achieves a 26% error rate on 10 cases. The third approach aims to improve the second method's robustness and achieves a 16% error rate. All methods use a Compression Dissimilarity Measure (CDM) with the Prediction by Partial Matching (PPMd) text compressor. Longer texts and better compressors are noted to be key for accurate predictions.

Another compression method was proposed by Hürlimann et al [23], where they used a compression feature as part of a set of different features for the PAN 2015 challenge, using the compression ratio instead of the length of compressed documents in an ensemble named GLAD. Using a binary SVM classifier, they found that the compression feature showed only moderate performance in the full feature set. Nonetheless, the overall performance was one of the best methods proposed in the PAN 2015 competition for three languages, except English.

Halvani et. al. [18] also proposed a compression-based AV method. Their method requires only a compression algorithm, a (dis)similarity measure and a threshold. They used five different compressors and compression distance similarity measures, and achieved a similar performance to the Veenman & Li [41] method and several state-of-the-art methods, including neural networks, at the time of publishing. The authors note this method is very fast and easily implemented, as there is no feature engineering or natural language processing methods necessary.

However, as the authors also note, compression suffers from a lack of interpretability and no explainability methods have been applied. As no feature engineering or tokenization is used, the texts are compressed and set to be assessed as same author or different author by the threshold only, which was arbitrarily set. Although good results can be obtained, even cross-genre and cross-topic, it is unclear why these methods perform so well. Therefore, the applicability of these methods in criminal forensics is likely limited. All aforementioned compression methods used long, pre-processed text as well.

3.3 Neural Network-based Models

In the past years, deep learning methods have dominated research in AV methods. Various deep architectures have been proposed to tackle AV problems, from deep combinations of stylometric features to attention-based similarity learning using siamese networks.

Boumber et al. [10] proposed a method using a a deep LSTM-based language model called ULMFit on an augmented training set. They trained an ensemble of RNN and QRNN classifiers using the forward and backward encodings generated by the ULMFit model, then evaluated the test data while applying test-time data augmentation. Both document manipulation and adversarial noise injection by LM are employed as data augmentation techniques to improve model generalization. Document pairs of known and unknown authorship are encoded as embeddings, which are used to train the ensemble of RNN and QRNN. The experiments were conducted with four AV datasets, and the authors demonstrate high robustness and stability as well as high performance in AUC and accuracy.

Litvak [29] proposed a method for AV tasks of short email messages, implementing binary classification with a seq2seq model, training a CNN on D_a and D_u. This multilayer neural network approach is trained in an end-to-end fashion on minimally pre-processed "raw" text, and learns several layers of feature extraction.

Techniques such as deep learning have a considerable disadvantage of needing a lot of data, which is difficult for AV as there are few corpora available suitable for this task specifically [9]. Furthermore, the fundamental need for large datasets makes deep learning extremely difficult to use in forensic setting, as there is typically very little data available in a specific criminal case.

3.4 Explainable AV methods

Several methods have attempted to use prompting for Large-Language Models (LLMs) for AV, and using explanations provided by the model itself. Hung et al. [22] proposed PromptAV, which uses LLMs to determine authorship with step-by-step prompts which are explainable, according to the authors. They report a superior accuracy score of 0.587 over other prompting methods, and a "meticulous analysis of linguistics features". The zero-shot PromptAV method does provide confidence score for several different linguistic markers, but it is not clear from this paper where that confidence is based on or what exact differences the model detected. Huang et al. [21] used LLMs for both AA and AV tasks, and assessed their ability to provide explanations. They tested LLMs with varying levels of 'guidance' in the prompts, with a specific list of linguistic features that the model should focus on. They found that a GPT-4 model with very specific prompts achieved highest performance for AV tasks, but that its performance was highly dependent on the type of prompt guidance. They generate word clouds as an explainability method, but no analysis of the mechanism of the models itself is provided.

Boenninghoff et al. [7] propose an explainable AV method using attention-based similarity learning within a Siamese neural network, called AdHominem (Attention-based Deep Hierarchical cOnvolutional siaMese bIdirectional recurreNt nEural-network Model). The AdHominem model operates in three stages: first, text preprocessing; second, feature extraction, where characters, words, and sentences are encoded into a single neural feature vector, with a character-to-word layer capturing prefixes, suffixes, and spelling errors. The attention layers are used to highlight significant words and sentences. In the third stage, a nonlinear metric learning module measures document similarity. The model significantly outperformed stylometric-feature-based systems, cutting error rates by half, even in challenging same-author/cross-topic and different-author/same-topic scenarios. AdHominem can handle unknown tokens, like misspelled words, and learns atypical expressions. However, its explainable features require manual expert analysis, and the authors also note the lack of a comprehensive quantitative analysis of explainability.

For large-scale datasets and especially forensic analysis, this would become complicated. The authors also note that they were not able to conduct a comprehensive quantitative analysis of explainability, but that this framework would be efficient for big datasets in forensic linguistics.

Besides techniques such as these needing a lot of data which makes it an unrealistic option for forensic settings, the use of attention weights as explanations is topic of debate as well. Jain & Wallace [24] found poor correlations between attention weights and other explanation methods, and that shuffling prediction weights in a neural model do not affect the final prediction. Several other studies as outlined by Bibal et al. [6] have countered some of their claims and showing that although attention weights are not necessarily *the* explanation, they can serve as *an* explanation, especially in NLP tasks such as part-of-speech tagging and syntactic tagging.
However, as Lyu et al. [32] also remark, there is a lot of ongoing discussion and no consensus on faithfulness nor plausibility of these mechanisms. Although the role of attention mechanisms is a valuable scientific question, it is still quite far from being an established form of explainability [32]. Especially in domains with very specific and narrow constraints such as authorship verification, and even more specifically high-stakes applications such as forensics or even plagiarism detection, white-box models are preferable until that time if the degree of explainability in both faithfulness and plausibility must be easily verified.

Lyu et al. [32] mention that white-box evaluations can be used as a transparent method in NLP tasks. White-box models in this context are models such as Logistic Regression or Decision trees, where the explanation of important features can be directly obtained from their predictions. Models such as these can be evaluated directly as their evaluation can be compared to ground-truth feature importances (for example, consensus on stylometric features determined by a forensic linguist). Lyu et al. [32] recommend white-box models for faithful evaluation methods in NLP specifically. However, they note since the transparent set-ups are simplified, a white-box model does not always automatically faithfully generalize to real-world applications. In order to generalize methods, they remark it is important to define faithfulness in advance rather than ad-hoc, state the assumptions of the evaluation, and to distinguish clearly between the capacity of the model and the quality of the explanation [32]. This is in line with Chaski's [12] recommendation of defining the requirements prior to experimentation for forensic linguistic analysis, in order to avoid tweaking the parameters for a desired result.

Keeping this in mind, and the increasingly high standards for applicability of machine learning for high-stakes applications such as forensics, it might be rewarding to shift attention to exploring and describing the limits and possibilities of transparent models first. With transparent models as a starting point, more complex methods can be developed from there that still obtain adequate performance. Although it is unlikely that a rather simple transparent model could conduct a thorough and complete authorship verification analysis, it might be worth uncovering whether a transparent model picks up the same patterns as a forensic linguist, and whether those evaluations are faithful and/or plausible. As Halvani [14] noted, straightforward SVM models have obtained some of the best performing author profiling methods, and as Barredo Arietta et al. [4] noted, more complex models do not always necessarily perform better. To the best of our knowledge, there is no prior work using token n-grams and descriptive features such as punctuation, word length statistics and vocabulary richness on short minimally pre-processed informal text using transparent and shallow ML models, and using the coefficients as a transparent white-box evaluation method for an AV task.

Therefore, the experiments conducted in this thesis aim to assess whether Logistic Regression can serve as a suitable transparent mode for this task, as well as as linear kernel SVM, and XGBoost. Although the latter two are not transparent models, their performance might serve as a valuable reference point for the performance of Logistic Regression, as they are able to model more complex patterns in the data due to being non-parametric models. If they do perform substantially better, other well-established explanation methods such as LIME could still be applied.

4 Methods

This research will focus on using lexical features, unigrams and bigrams, using varying text length and descriptive features using widely used and well-understood models to assess their use in forensics.

4.1 Data description

The dataset used for this task is de PAN @ CLEF Profiling Hate Speech Spreaders on Twitter dataset from 2021. Although this dataset was not designed for an authorship verification task, this dataset was chosen because it contains 200 training cases/authors, with 200 tweets per author, which ensured there were enough tweets per author to conduct reliable experiments. Other reasons for choosing this dataset was that it served as an adequate proxy for real forensic data as tweets are short-text data, and it was available upon request, whereas real forensic datasets are naturally not accessible to the public. The dataset contained 200 XML files, each file representing a unique author, with 200 tweets per author. Each tweet is denoted as a document in the XML structure.

The dataset, for its original purpose, consists of 200 different authors with 200 crawled tweets from their feed each that either contain or do not contain hate speech towards a person or group on the basis of characteristics such as race, colour, ethnicity, gender, sexual orientation, nationality, religion, or other. If an author's timeline contains hate speech, it can contain hate speech towards multiple groups, and not every tweet of that author contains hate speech. Therefore, the tweets can be considered cross-topic text. Although the original dataset contains datasets for both Spanish and English tweets, only the English tweets were used for this experiment.

Some excerpts of the tweets in the files are shown below in Figure 1.

As the dataset was composed in 2021 or before that, one tweet can contain a maximum of 280 characters and can include emojis. Tweets are notoriously difficult datasets for text analysis purposes – tweets often contain spelling or grammar errors, slang, abbreviations or other abnormalities. However, for the purpose of testing models for forensic applications, short text with abnormalities is precisely the type of text needed to assess whether a model can perform adequately.

4.2 Data preparation

4.2.1 Pre-processing

The original dataset had already marked elements such as user handles or hashtags as #USER# or #HASHTAG#. Therefore, raw tweets were pre-processed to remove the marked elements

```
Weird vibes today. They're strong
though. I'll make the most of it.

spirited away is GOATED !!

Scored a pair of golf clubs.. Who tryna
go go golfing??

I need sumn to watch, Anime preferably

it's chewsday innit

My friend is not having fun in ATL
:neutral-face:
```

Figure 1: Tweet excerpts

of user tags, hash tags, and URLs, as these elements can be used by many users and only contain non-specific information that might introduce noise, since the actual hashtag or user mention is not visible. Then, retweets were removed so only the tweets written by the author were left in. The tweets were also filtered by removing duplicate tweets, as accounts that have very many duplicate tweets often indicate that it can be a bot rather than a person.

The maximum number of characters allowed in a tweet has changed over the years. As this dataset was created in 2021 or prior to 2021, tweets are a maximum of 280 characters. Considering a single tweet would yield too little text for one instance, tweets are concatenated with N tweets per sample. An author was mapped to N amount of tweets per data sample, where N amount of tweets per data sample were randomly selected from the author's list of tweets, without replacement. Each lexical feature, word or special symbol, was tokenized and converted to lowercase. Punctuation and contractions (e.g. "i'm" instead of "I am") were preserved during tokenization, because punctuation proportion is used as one of the descriptive features and contractions can be a discriminative feature. Also, as using symbols to denote a word rather than writing out a word entirely, such as using the ampersand '&' instead of the word 'and', can be a powerful discriminative feature, all entities separated by a space are considered a token. This is in line with Chaski's [12] recommendations to minimize text preprocessing, and to be able to assess non-prescriptive ways of writing concerning spelling and grammar.

The top *k* most frequent unigrams and bigrams were extracted from all tweets over the entire dataset. The tweets were then vectorized using the TfidfVectorizer from the sklearn library, passing the list of most frequent words and bigrams to the vocabulary parameter. If a particular's author's tweets are represented in both the training set and test set, model performance may be overestimated as it has already seen the tweets of that specific author before. Therefore, the authors, rather than the tweets, were randomly split into a train and test set to prevent data leakage, with an 80-20 ratio for training and testing.

One data point then consists of a pair of two feature vectors, where vector 1 represents tweet A, and the vector 2 represents tweet B, accompanied with a label. The label was true if tweet A and tweet B were written by the same author, false if tweet A and tweet B were written by different authors. This then results in same-author feature vectors, and different-author feature vectors.

The dimensions of the feature vectors would be doubled if the vectors were concatenated, which would add complexity for the chosen methods. Therefore, to obtain a new feature vector, the absolute value of the difference between each feature vector was taken using an element-wise measure, as shown in equation 1. This method resembles the Manhattan distance measure.

$$\delta(x,\ y)_i\ =\ |x_i\ -\ y_i|,\ where\ x,\ y\ \in\ \mathbb{R}^n \tag{1}$$

Other distance measures, such as cosine or Euclidian, return a single value representing a comparison between two feature vectors, where the similarity between the vectors is compressed into a one-dimensional score. This leads to a loss of information, and this interferes with the possibility to examine the feature vectors in terms of interpretability. If the dimensions of the feature vector remain intact, it enables us to assess exactly which word or descriptive feature is represented in the feature vector.

This approach results in same-author feature vectors, and different-author feature vectors as shown below in Figure 2.

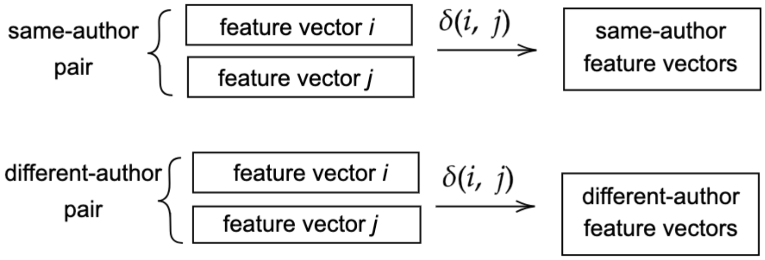


Figure 2: Feature vectors used for modeling

Naturally, with creating pairs of text by random sampling, there will be many more different-author pairs than same-author pairs, which can lead to class imbalance. When there is class imbalance in a dataset, this adds significant bias during the modeling phase. The model may then consistently predict the most frequent class, or not be able to distinguish well between classes at all. Therefore, same-author feature vectors and different-author feature vectors were randomly sampled from all possible pairs to ensure the training set and test set both consisted of 50% of same-author feature vectors, and 50% of different-author feature vectors, such that both classes are equally represented.

4.2.2 Author descriptive features

Several descriptive features that have been found as distinctive style markers in literature were added to the model as well [33, 2, 28]. For the author descriptive features, five different features were calculated over each sample containing N tweets of each author: the type/token ratio (TTR), the proportion of punctuation, the mean word length, the standard deviation of word length, and the fraction of words in a tweet sample that are function words.

The TTR is the ratio obtained by dividing the total number of different words (types) occurring in a text by the total number of words (tokens) [31]. The TTR can be used as a measure for lexical variation in the text. A high TTR, closer to 1, indicates a high degree of lexical variation, whereas a low TTR, closer to zero, indicates a low degree of lexical variation.
The proportion of punctuation is calculated as the ratio of number of punctuation characters to the total number of characters in the sample, to assess the density of punctuation marks in a tweet sample.
The mean word length calculates the average word length in the sample. Then, the standard deviation of this average word length is measured to indicate the spread of this average value. The mean word length can be an important style marker, as text with higher word lengths are often more complex or formal. The standard deviation helps to understand the consistency of word usage in terms of their length within the text.
Lastly, **the proportion of function words** per tweet sample was calculated. Function words typically have little lexical meaning but serve a grammatical purpose. The full list of function words used can be found in Appendix A.

The descriptive features are added at the end of the feature vector of the most frequent words vector, such that their respective contribution to the model can be easily assessed in the coefficients analysis after fitting the models. To ensure proper interpretability of the ranking of the coefficients, a scaler was applied to the vectors. This changes the one unit of increase to one standard deviation increase.

5 Results

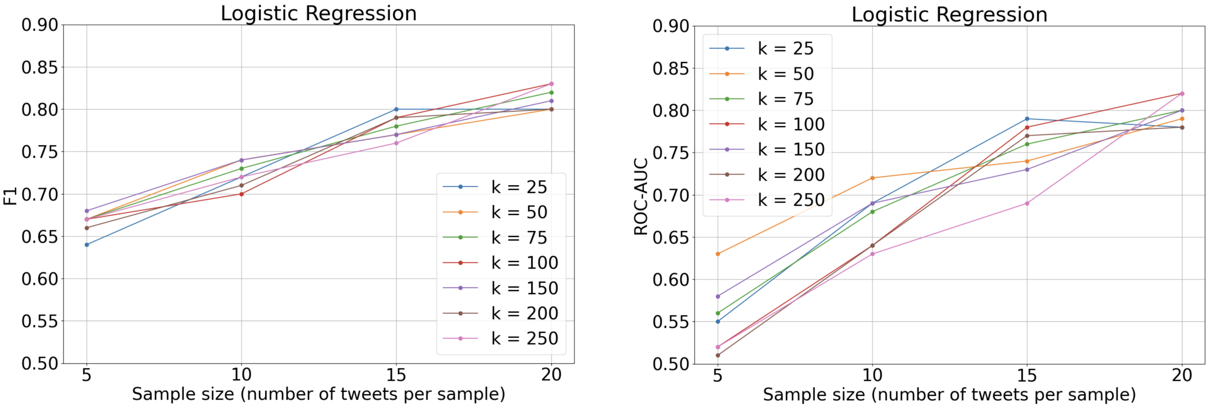
The results of the experiments are displayed below. These experiments consist of running the experiment with different values for several parameters, based on word frequency. The parameters currently included are the top *k* most frequently occurring words, the number N tweets per sample, with or without idf, and emojis included. The training set size is set at 10.000, and the test set size at 2000.

5.1 Varying text length per sample

The variation of top *k* words is included in the legend, and the sample length varied from $N = \{5, 10, 15, 20\}$ tweets concatenated per sample. As shown in Figure 3, 20 tweets per sample yielded the highest F1 and AUC across all values of *k*. This is in line with expectations following the literature, as more text per sample is usually necessary for a reliable result. As shown in Figure 3, just including about 5 tweets per sample results in an AUC around 0.60, which amounts to the model only being slightly better than random guessing. At about 15-20 tweets per sample, the model is able to achieve an AUC between 0.75-0.85, which indicates the model becomes better at distinguishing between classes. The graphs for SVM and XGBoost can be found in Appendix B.

		F1	*ROC-AUC*
With idf	SVM	0.84	0.82
	Logistic Regression	0.83	0.82
	XGBoost	0.83	0.81
Without idf	SVM	0.82	0.80
	Logistic Regression	0.81	0.79
	XGBoost	0.80	0.77

Table 1: Results with most frequent words with $k = 100$ and $N = 20$ tweets per sample, with and without tf-idf.



(a) F1

(b) ROC-AUC

Figure 3: Performance of the Logistic Regression classifier with varying values of N and k.

5.2 Precision and recall

As shown in Table 2, there is a gap between precision and recall performance in the different-author class and same-author class. For the different-author class, the precision is higher compared to the lower recall. This result suggests that the models are somewhat conservative in predicting different-author pairs. Although the models produce few instances where they are incorrectly classified as texts by different authors, it fails to detect a number of different-author cases. Conversely, for the same-author pairs, precision is lower and recall is higher. This result suggests that the models sometimes produce false positives, i.e. incorrectly predict that two texts are written by the same author while they are not, but that the models identify most cases where the texts are by the same author correctly. In this experimental setting, same-author pairs are usually correctly identified, but the models misclassify a number of different-author pairs.

5.3 Assessing model performance stability over 10 runs

The following figures compare the performance (AUC) of the models over multiple runs for varying text length samples.

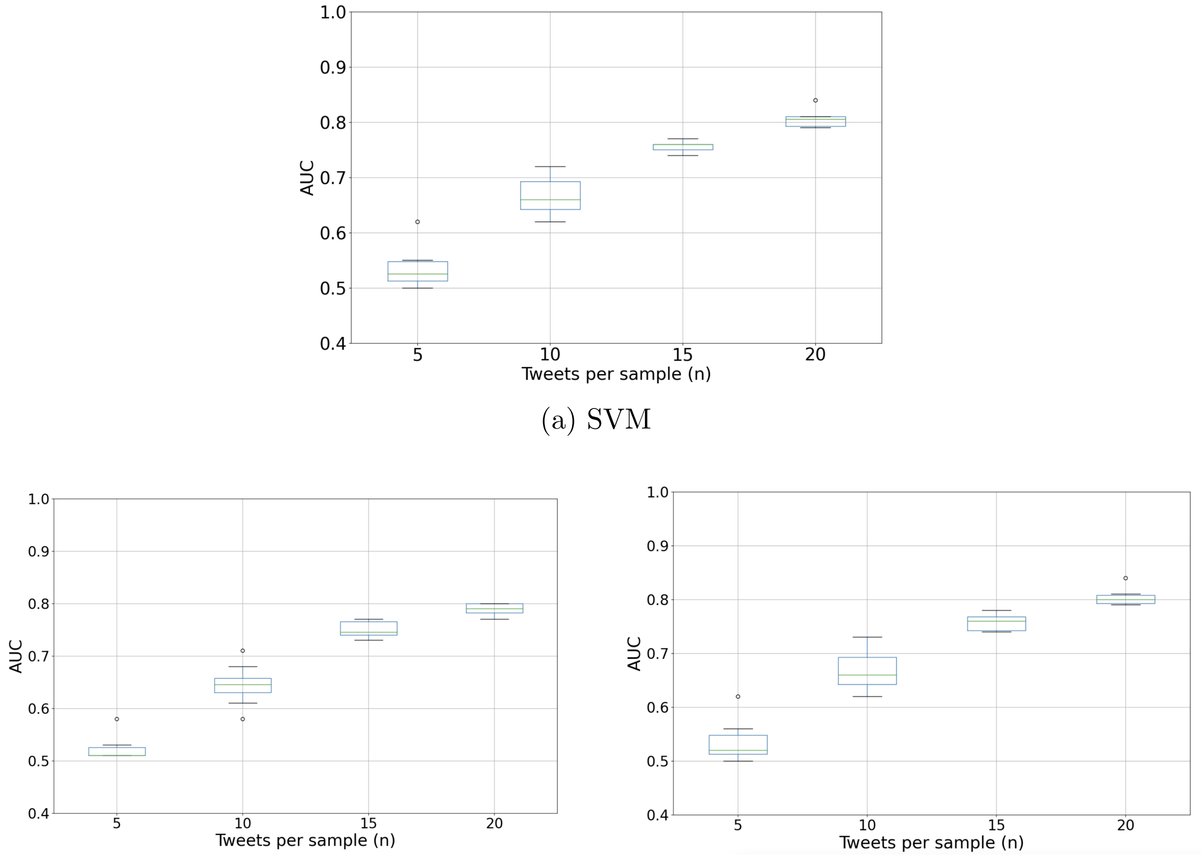
As shown in Figure 4, the model shows an even distribution of AUC scores around larger text samples as well, indicating that the model is more stable in its ability to distinguish between

	Model	Precision	Recall	F1
Different-author class	SVM	0.89	0.77	0.82
	Logistic Regression	0.87	0.72	0.79
	XGBoost	0.87	0.69	0.77
Same-author class	SVM	0.81	0.90	0.84
	Logistic Regression	0.76	0.89	0.83
	XGBoost	0.77	0.90	0.83

Table 2: Precision and recall for different-author class (0) and same-author class (1), with $N = 20$ and $k = 100$.

classes when the text sample size increases. With a low amount of tweets per sample, between 5-10 tweets per sample, the variability in performance is much larger. The top strongest coefficients, i.e. ranking of the words, also varied much more with 5-10 tweets per sample across runs, and became stable with 15-20 tweets per sample.

This is in line with expectations following the literature. For a robust performance, there needs to be a sufficient amount of text to train on to use ML models. It stands to reason that just a few words or sentences do not allow most ML models to uncover any patterns.



(a) SVM

(b) XGBoost

(c) Logistic Regression

Figure 4: AUC performance of the classifiers across 10 runs for each $k = 100$

5.4 Model results with both unigrams and bigrams

The following table displays the results of incorporating both unigrams and bigrams in the model. As shown in Table 3, integrating both unigrams and bigrams into the model increases overall model performance. However, it should be noted that bigrams are fairly sparse in this dataset, as the first top *k* bigrams did not come up until $k = 100$ and above. However, since there is a slight increase in performance, this result suggests that considering both unigrams and bigrams leads to the model becoming better at distinguishing between classes correctly.

	F1	*ROC-AUC*
SVM	0.85	0.83
Logistic Regression	0.85	0.83
XGBoost	0.84	0.81

Table 3: Results of using both unigrams and bigrams with k = 150 and $N = 20$ tweets per sample.

5.5 Adding descriptive features of authors

The following descriptive features were added: 1) type/token ratio to measure vocabulary richness, 2) punctuation proportion, 3) average word length, 4) word length distribution and 5) function word frequency. These features were added to the models as features to determine their relative contribution to the model's prediction in addition to unigrams, based on their coefficients. Several descriptive features consistently presented in the top 10 coefficients or weights across all models, with the average word length and punctuation proportion as the top two respectively. Whether function words were in the top thirty was largely dependent on how many function words were included. Notably, the more function words were included, the less powerful the contribution was to the model.

As shown in Table 4, model performance slightly increased when descriptive features were added. Especially the AUC increased across all models, indicating that the model became better at distinguishing between classes. The overall results suggest that including counts of not only unigrams and bigrams, but also descriptive features result in a more powerful model for authorship verification.

		F1	*ROC-AUC*
Word-based features only (unigrams, bigrams)	SVM	0.85	0.83
	Logistic Regression	0.85	0.83
	XGBoost	0.84	0.81
Word-based features + descriptive features	SVM	0.86	0.84
	Logistic Regression	0.85	0.84
	XGBoost	0.84	0.83

Table 4: Results with most frequent unigrams and bigrams only, and unigrams and bigrams plus the descriptive features, with $k = 150$ and $N = 20$ tweets per sample.

5.6 Examining coefficients and weights for interpretability

5.6.1 Global explainability: comparing the coefficients and weights between the models

Table 5 shows the top 10 features for each of the models. As SVM is implemented with a linear kernel, the coefficients can be easily compared to that of Logistic Regression ('LR' in Table 5). Although Logistic Regression uses statistical (probabilistic) measures and SVMs is based on a geometrical approach, if the SVM is implemented with a linear kernel such as in this experiment, their performance is expected to be similar with a sufficiently large dataset size [25]. As shown in Table 5, the ranking of the top features is almost exactly the same, except for the last two features. In contrast, after the first four features, XGBoost provides a different ranking from Logistic Regression and SVMs. Although XGBoost performs well with complex non-linear problems, it is more prone to overfitting than linear models and usually requires fine-tuning, which can also be seen in its comparative performance in Table 4. This result suggests that linear models can do quite well on this task in this experimental setting, and using a model that is able to handle more complexity does not result in a better performance. Regarding interpretability, this is a satisfying result – this means we can use the more transparent model Logistic Regression for this task, without having to resort to post-hoc explainability methods.

	LR		***SVM***		***XGBoost***
feature	***coef***	***feature***	***coef***	***feature***	***weights***
MEAN_WORD_LEN	-1.238	*MEAN_WORD_LEN*	-0.875	*MEAN_WORD_LEN*	0.361
PUNCTUATION	-0.878	*PUNCTUATION*	-0.641	*PUNCTUATION*	0.124
biden	-0.593	biden	-0.421	biden	0.054
u	-0.409	u	-0.278	u	0.040
i'm	-0.361	i'm	-0.251	*FUNC_WORD*	0.030
and	-0.341	and	-0.239	*TTR*	0.029
i	-0.312	i	-0.205	i'm	0.017
the	-0.289	the	-0.197	and	0.015
this	-0.285	this	-0.192	is a	0.012
&amp;	-0.269	an	-0.179	of	0.012
TTR	-0.240	are	-0.164	this	0.011

Table 5: Top ranked coefficients (*coef*) or weights of all models. The words are listed, and the descriptive features are capitalized and italicized to distinguish them from the words in this list: *mean_word_len* refers to the average word length, *func_word* refers to the proportion of function words, *TTR* refers to the the type/token ratio and *punctation* to the punctuation proportion.

5.6.2 Logistic Regression coefficients interpretation

Table 6 shows the top 10 words and descriptive features that resulted from the Logistic Regression model. Although the SVM model performed very slightly better, the logistic regression model performed almost as well and its coefficients are very easily interpreted. For the sake of interpretability, we can assess the meaning of these tables more easily in context using logistic regression coefficients compared to SVM.

top k position	***feature***	***coefficient***	***odds ratio***
152	*MEAN WORD LENGTH*	-1.238	0.290
151	*PUNCTUATION*	-0.878	0.415
107	biden	-0.593	0.553
70	u	-0.409	0.665
38	i'm	-0.361	0.697
6	and	-0.341	0.711
2	i	-0.312	0.732
0	the	-0.289	0.749
10	this	-0.285	0.752
54	*&amp; (ampersand)*	-0.269	0.765
150	*TTR*	-0.240	0.786
61	an	-0.235	0.790
52	why	-0.231	0.793

Table 6: Logistic Regression top coefficients and odds ratio

Table 5 shows the top 10 negative coefficients. The negative sign in front of the coefficient might seem counterintuitive, but keeping in mind that what we measure is the distance between two feature vectors and a positive prediction (1) refers to predicting that it was a same author-prediction, this means that as the distance between two vectors increases, this feature decreases the likelihood of predicting a same-author pair for that feature if all other variables remain fixed. To make it easier to interpret, if we take the odds ratio of the coefficient with exp*(coefficient)*, we see that for instance, for every unit increase in distance between two vectors for the feature mean word length from Table 6, we see a $1 - 0.290 \approx 71\%$ decrease in odds in predicting a same-author pair. This is in line with expectations that if two texts contain very different average word lengths, it is less likely they are written by the same author. The same holds for punctuation, which is in line with the literature that punctuation use is a powerful discriminative feature to distinguish between authors.

A few of these in the top 10 words (unigrams) stand out. First, For example, the features 'u' and the ampersand '&'. In informal short text such as tweets or text messages, the letter 'u' is often used to denote the word 'you', and the ampersand is used to denote the word 'and' instead of writing it out. Intuitively, it would make sense that someone who consistently writes the word 'you' as 'u' is a powerful discriminative feature when comparing two texts to assess whether they have been written by the same author. Likewise for using the ampersand, instead of writing out the word 'and'. Manual inspection of several tweets indeed confirms that the ampersand is very often used to denote the word 'and'.

Conversely, table 7 shows features that with every unit increase of the distance between two vectors for that feature, there is an increase of likelihood for a same-author prediction. For example, for every unit increase in distance between two vectors for the word 'say', there is an increase of 1.257 in odds for a same-author prediction, with all other variables fixed. It is unclear why exactly a bigger difference between the use of the words in Table 7 would result in a same-author prediction. It could be that due to the small dataset size or that these words are difficult to write in a different way. It is also shown in Table 7 that all of these features are very close in score, and the differences between them are very small, with only 0.147 difference

between the first and last feature in the table.
They are also very close to 1, which means that odds of being either class are almost equally likely. Note that the negative coefficients in Table 6 are all stronger than the positive coefficients Table 7, suggesting the influence of these features is more impactful to the model compared to the top 10 positive coefficients. The difference between them is greater as well, and further from 1.

top k position	***feature***	***coefficient***	***odds ratio***
115	say	0.229	1.257
153	*STD WORD LENGTH*	0.179	1.196
42	as	0.179	1.196
121	too	0.158	1.171
129	here	0.146	1.158
78	them	0.143	1.154
71	go	0.142	1.152
57	know	0.130	1.139
101	still	0.130	1.139
62	want	0.129	1.138
127	being	0.124	1.132
58	got	0.104	1.110

Table 7: Logistic Regression top 10 positive coefficients

5.6.3 Use case example

Consider the following text boxes in Figure 5, each of which contain 20 concatenated tweets, from the same user. For reference of what the tweets originally looks like, the example refers to the original text in the tweets before the preprocessing steps described in chapter 4.2.1. The Logistic Regression model predicted this correctly as being written by the same author, with a predicted probability of 0.903 for the positive class, i.e. a same-author prediction.

The words from Table 6 are marked green, the words from table 7 are marked red, and the punctuation from Table 6 is marked light blue. The green refers to the words or features that if the distance between those features increases, it is less likely that it will result in a same-author prediction, which is the effect we expect to see. The words marked red are from Table 7, where an increase in distance led to a higher likelihood for a same-author prediction.

Nothing says Texas like this .

HOPE YOU'RE SHUT DOWN

has YOUR shirt on - awesome.

BEAUTIFUL family! Blessed!

Back at ya dummy! This is what YOU do to us ;)

Probably not my place to give others advice but it's not a good idea to have this before a 10 mile run.

Trying to be "fair and balanced" but going down the toilet...

Watching from East Texas because

Yup! said you could find it this morning in his interview! !

Marquez Rodriguez will spend 30 months in federal prison for selling counterfeit immigration documents.

Ronna has tested positive too

Eye twitching ? What's up with that?

HOW FUN! (For your kiddo now - not for you at the time.)

Can't believe some one doesn't let know has died... ? ? ?

LOVE IT! says the came in from China

I was going to comment on this tweet - WHY did YOU take it down? People WRONG color?

Well, heck! Show us which one: - you must be a blast to hang out with!

Now we know why NEEDED a break every 30 minutes!

Book sales didn't pan out, huh...

Attaboy AGAIN Scott!

talking reminds me of Uncle Joe on Petticoat Junction - move on !

CAN NOT BELIEVE dot com doesn't have breaking news about Ruth Bader Ginsburg's passing...

I'm trying to type Biden drug test magically CHANGES my hashtag to something else... WHY?

and hubby will be felons forever

is SERIOUSLY ill!

is trending - He's crazy

or are you too busy watching your Red Sox? LOL

This is just cruel. Please don't retweet it five million times.

They must be better for Canadians... I just got ripped too many times, glad they're good to you!

think Uncle Joe - Petticoat Junction...

THANK YOU for fighting You're ! !

They're a rip off! Does no good to complain because they'll come back with "look at all the pennies we paid you before"

and Chris Wallace defends him as a stand up guy - so you know he's a joke...

Just because Romney says he won't block the vote for doesn't mean his vote will be a yes...

STUCK ON STUPID

RIGHT ON ERIC!

? tomorrow? ? He'll be LIVE on the EIB network!

Another crush left to soon... Rest in Peace Eddie, rock on!

Talk about the tomorrow Charlie - starring OUR FAVORITE PRESIDENT

Remember that blood spot in his eye ?

Figure 5: Predicted probability: 0.903 for a same-author prediction.

6 Discussion

The experiments show that the top-*k* words and descriptive features have discriminative power when distinguishing between same-author or different-author text samples when using transparent and shallow ML models. The effects of the coefficients of logistic regression, and their odds ratios, can be directly interpreted in a text sample, while using the model to provide a predicted probability of the same-author or different-author class. The example and results suggests that the most frequent unigrams and bigrams and author descriptive features, as in line with analysis and consensus from forensic linguistic experts, show predictive power to assess whether a text was written by the same author or not. Although the performance is not as good as some methods in other previously mentioned research, considering how rudimentary this approach is, performance is still quite decent.
It should be noted however, that although there is some predictive power in using the top-*k* unigrams, bigrams and descriptive features, the top most used words and even descriptive features are still quite a naive approach of language use for any individual. Ultimately, all that is really considered are counts – counts of words, use of punctuation, and so on – which are not all-encompassing of how human beings use language to communicate [12]. However, counts have the advantage of being quantifiable, which may be difficult or tedious to do manually for a forensic linguist when analyzing two specific pieces of text. Therefore, it may be beneficial to think of these experiments as a tool to quantify *some* parts of analysis that have been shown to be influential when assessing authorship, rather than it being a tool that can replace manual analysis as a whole. Furthermore, there are several limitations to be considered in this research.

6.1 Limitations

A few things stand out in the results. First, there are some content words in the top *k* words that show up in the top coefficients list, especially that the word ‘biden’, referring to president Joe Biden of the United States of America, and the word ‘trump’, referring to presidential candidate and former president Donald Trump, were relatively strong predictors in the entire list. This could be because the dataset was from 2021, and it may have been composed before that, in 2020, which was an election year in the United States (U.S.). Presidential candidates may have therefore been a hot topic on Twitter, as the U.S. has the highest number of Twitter users by country. The 2020 election also took place during the COVID-19 pandemic, and it was an especially controversial election compared to different years and was a frequently mentioned topic on social media well until later in 2021, as the U.S. Capitol attack also took place on January 6th 2021 [13]. That does make the dataset biased when it comes to topic, as a lot of tweets mention these presidential candidates, and the overall political situation in the U.S. was a frequently mentioned topic on social media overall during this time. As this dataset was also composed for the purpose of detecting hate speech rather than AV, it is even more likely that topics such as politics were included. This makes the dataset highly biased. Because no topic-masking was performed in these experiments, this could have affected prediction.

Second, microblogging data is difficult to translate to actual ground-truth data, as the author can only be identified by their online handle, and these handles are often anonymous. Twitter does not require verification of identity, or any information that can reliably link a handle to a specific person unless they choose to supply that information. Microblogging data is also not of the same structure as many forensic texts, as tweets can either just be posted on the author's feed, they can also be replies to others or serve as commentary on something else that cannot be included in raw text, such as images, news sources or other implied points of reference. Microblogging could still be a useful subgenre of forensic text, as there are plenty of limited content-moderated forums which may attract criminals or terrorists with anonymous authors such as ORC, Parler, Gab, 8chan or even private channels on Telegram.

Third, is it very difficult to detach topic as a whole when using word counts, which may introduce more bias when texts are truly very cross-topic or very short of length. Although the complete coefficient list contains many features that are topic-independent that are in line with research, i.e. low-semantic words, it is likely that the small dataset size and specific time and original purpose of the dataset is still topic-dependent to some degree. Granted, even real-world forensic cases might be topic-dependent – for example, when examining seized communication between several members within a specific crime circuit such as drug-, weapon- or human trafficking – it should be noted as a limitation, because this specific set-up may yield different results with a different dataset.

Fourth, although models such as Logistic Regression, SVM and GradientBoosting are considered transparent and/or interpretable models (possibly combined with post-hoc explainability methods) to machine learning experts, they still may not be considered as interpretable to judges, lawyers and other people of the court if analysis is submitted as evidence, as these methods are to domain experts. Court officials will still have to trust a forensic data scientist in their expertise to some extent, because even odds ratios might not be interpretable to laymen. However, this might not present a realistic problem: after all, people of the court may also not fully grasp the details of other forensic analysis either, such as DNA or blood spatter. If they accept that a forensic linguistic analysis is an acceptable piece of forensic evidence, it is up to both data scientists and forensic linguists to reach consenus of what constitutes "explainable (or transparent) enough".

7 Conclusion

The literature reviewed in this thesis showed that the current state of AV research is at the point where although promising AV methods have been developed thus far, currently, the majority is likely inadequate with regard to model interpretability for the high-stakes fields such as criminal forensics. Regarding forensic applicability, many of these methods are developed and tested with long and extensively pre-processed text to improve model performance, as well as developing complex architecture that are not transparent in their mechanism, so it is unlikely that many of these methods will be useful in a criminal forensic setting.

There are several challenges in AV regarding forensic applicability, but two substantial challenges are the nature of forensic text and the lack of transparency of most AV methods. Forensic text is often short, and full of non-standard language use and grammar mistakes, which can be difficult to analyze. A lack of openly available forensic texts is another challenge in developing methods that can be tested outside of litigation. The requirement that a forensic expert can explain the mechanism of their method in the courtroom and provide transparency and full understanding of the method to laymen will likely render most of the methods from recent research of not much use. Much of AV research focuses on improving model performance over interpretability and applying explainability methods within AV tasks, so the literature about this intersection of the task and interpretability is scarce.

The experiments in this thesis were conducted with minimally pre-processed short text, using word-based and descriptive features that have shown to have discriminative power in determining authorship following forensic linguistic literature, with transparent and two shallow ML models. Of the transparent model Logistic Regression, the coefficients can be interpreted directly, and widely used post-hoc explainability methods can be used for the SVM and XGBoost. Logistic regression using word-based and descriptive features achieved an F1 of 0.85 and AUC of 0.84 in this experimental set-up. The top coefficients and odds ratios show the strongest predictors, which can be showcased in an example comparing two texts and the predicted probability for that sample.

These results suggest that these methods could potentially form an effective yet interpretable method. Although more complex methods can achieve higher performance, the interpretable nature of these models and the features may be preferable in high-stakes applications such as criminal forensics. Transparent models, although quite naive approaches of AV, could serve as a tool to quantify *some* parts of text analysis when determining authorship, rather than functioning as a full replacement of a forensic linguist. A forensic linguist might use a transparent model such as the one presented in this thesis to conduct part of their analysis, if they wish so. Despite the many limitations in the experimental set-up in this thesis, a straight-forward approach using transparent models could lead to a more holistic approach in forensic linguistics, maximizing the advantages of both expertise of a forensic linguist and pattern recognition of machine learning so that some part of the analysis can be quantified if desired and applicable. To which degree these methods can be used in a forensic setting is hard to state exactly, but it will likely depend on the amount of text that is available for analysis, the type of text and task it concerns, and whether the desired methods of analysis can be tested appropriately to legal standards of evidence.

Future research should make it a priority to compose a comprehensive realistic forensic corpus, preferably with varying lengths of text and from various real cases where there is a groundtuth within the data, as well as available forensic linguistic expert analysis. This would enable researchers to conduct experiments with several types of forensic data such as threats, manifestos, and perhaps even several sources of communication between criminals with disputed authorship. An example of the latter could be where two phones were seized as evidence, and it must be determined based on the text messages on both phones whether they were written by the same person or not. Experiments using varying lengths of text and from different genres could provide insight on whether methods can and should be generalized across different types of authorship verification problems, or instead be developed for a specific type of analysis. Although it will likely be very difficult to compose such corpora due to confidentiality of sensitive data in criminal cases, it is essential to test further developed methods on real data in order to establish their applicability with certainty.

Explainability should also be a priority in methods developed for criminal forensics, preferably to be tested for both faithfulness on a technical level, and plausibility with use cases involving real-world experts, ranging from forensic linguists, judges and lawyers, and other machine learning researchers. Faithfulness might be difficult to achieve with deep architectures as long as there is no unanimity on how faithful and plausible explainability methods applied to hidden states are, but may be achieved when developers of AV methods are willing to partly compromise on optimal performance, especially if they make use of white-box models. While it may not be possible to developed a method that is fully plausible to all parties involved – after all, how plausible is a model such as logistic regression even to a legal expert with no technological background at all – the focus should be on methods where consensus can be achieved that it is a sufficiently reliable and transparent method as decided by the end users. Through engaging with the end user and experts from other fields and reaching consensus together rather than each expert staying in their own lane, it is far more likely that new methods will actually be used, approved, deployed and eventually be admissible in court as evidence.

References

[1] Unabomber — FBI.

[2] Janet Ainsworth and Patrick Juola. Who Wrote This: Modern Forensic Authorship Analysis as a Model for Valid Forensic Science. *Washington University Law Review*, 96:1159, 2018.

[3] Douglas Bagnall. Author Identification using Multi-headed Recurrent Neural Networks, August 2016. arXiv:1506.04891 [cs].

[4] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, June 2020.

[5] Nicole Mariah Sharon Belvisi, Naveed Muhammad, and Fernando Alonso-Fernandez. Forensic Authorship Analysis of Microblogging Texts Using N-Grams and Stylometric Features. *arXiv:2003.11545 [cs]*, March 2020. arXiv: 2003.11545.

[6] Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin. Is Attention Explanation? An Introduction to the Debate. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3889–3900, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[7] Benedikt Boenninghoff, Steffen Hessler, Dorothea Kolossa, and Robert M. Nickel. Explainable Authorship Verification in Social Media via Attention-based Similarity Learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 36–45, December 2019.

[8] Benedikt Boenninghoff, Robert M. Nickel, Steffen Zeiler, and Dorothea Kolossa. Similarity Learning for Authorship Verification in Social Media. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2457–2461, May 2019. ISSN: 2379-190X.

[9] Benedikt Boenninghoff, Julian Rupp, Robert M. Nickel, and Dorothea Kolossa. Deep Bayes Factor Scoring for Authorship Verification. *arXiv:2008.10105 [cs]*, August 2020. arXiv: 2008.10105.

[10] Dainis Boumber, Yifan Zhang, Marjan Hosseinia, Arjun Mukherjee, and Ricardo Vilalta. Robust Authorship Verification with Transfer Learning. March 2019. Publisher: EasyChair.

[11] Marcelo Luiz Brocardo, Issa Traore, Sherif Saad, and Isaac Woungang. Authorship verification for short messages using stylometry. In *2013 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–6, Athens, Greece, May 2013. IEEE.

[12] Carole E. Chaski. Best Practices and Admissibility of Forensic Author Identification. *Journal of Law and Policy*, 21:333, 2012.

[13] Emily Chen, Ashok Deb, and Emilio Ferrara. #Election2020: the first public Twitter dataset on the 2020 US Presidential election. *Journal of Computational Social Science*, 5(1):1–18, May 2022.

[14] Oren Halvani. *Practice-Oriented Authorship Verification*. Ph.D. Thesis, Technische Universität Darmstadt, Darmstadt, 2021.

[15] Oren Halvani and Lukas Graner. Rethinking the Evaluation Methodology of Authorship Verification Methods. In Patrice Bellot, Chiraz Trabelsi, Josiane Mothe, Fionn Murtagh, Jian Yun Nie, Laure Soulier, Eric SanJuan, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Lecture Notes in Computer Science, pages 40–51, Cham, 2018. Springer International Publishing.

[16] Oren Halvani, Lukas Graner, and Roey Regev. A Step Towards Interpretable Authorship Verification. *arXiv:2006.12418 [cs]*, July 2020. arXiv: 2006.12418.

[17] Oren Halvani, Christian Winter, and Lukas Graner. Authorship Verification based on Compression-Models, June 2017. arXiv:1706.00516 [cs].

[18] Oren Halvani, Christian Winter, and Lukas Graner. *On the Usefulness of Compression Models for Authorship Verification*. August 2017. Pages: 10.

[19] Oren Halvani, Christian Winter, and Lukas Graner. Assessing the Applicability of Authorship Verification Methods. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ARES '19, pages 1–10, New York, NY, USA, August 2019. Association for Computing Machinery.

[20] Ángel Hernández-Castañeda, Hiram Calvo, Ángel Hernández-Castañeda, and Hiram Calvo. Author Verification Using a Semantic Space Model. *Computación y Sistemas*, 21(2):167–179, June 2017. Publisher: Instituto Politécnico Nacional, Centro de Investigación en Computación.

[21] Baixiang Huang, Canyu Chen, and Kai Shu. Can Large Language Models Identify Authorship?, March 2024. arXiv:2403.08213 [cs].

[22] Chia-Yu Hung, Zhiqiang Hu, Yujia Hu, and Roy Ka-Wei Lee. Who Wrote it and Why? Prompting Large-Language Models for Authorship Verification, October 2023. arXiv:2310.08123 [cs].

[23] Manuela Hürlimann, Benno Weck, Esther Berg, Simon Šuster, and Malvina Nissim. GLAD: Groningen Lightweight Authorship Detection. January 2015.

[24] Sarthak Jain and Byron C. Wallace. Attention is not Explanation, May 2019. arXiv:1902.10186 [cs].

[25] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. *An Introduction to Statistical Learning: with Applications in Python*. Springer Nature, June 2023. Google-Books-ID: ygzJEAAAQBAJ.

[26] Moshe Koppel and Yaron Winter. Determining if two documents are written by the same author. *Journal of the Association for Information Science and Technology*, 65(1):178–187, 2014. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.22954.

[27] Johann Laux, Sandra Wachter, and Brent Mittelstadt. Trustworthy artificial intelligence and the European Union AI act: On the conflation of trustworthiness and acceptability of risk - Laux - 2024 - Regulation & Governance - Wiley Online Library, February 2023.

[28] Robert A. Leonard, Juliane E. R. Ford, and Tanya Karoli Christensen. Forensic Linguistics: Applying the Science of Linguistics to Issues of the Law. *Hofstra Law Review*, 45:881, 2016.

[29] Marina Litvak. *Deep Dive into Authorship Verification of email Messages with Convolutional Neural Network*. September 2018.

[30] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[31] Kim Luyckx and Walter Daelemans. Authorship Attribution and Verification with Many Authors and Limited Data. In Donia Scott and Hans Uszkoreit, editors, *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 513–520, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[32] Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. Towards Faithful Model Explanation in NLP: A Survey. *Computational Linguistics*, pages 1–67, March 2024.

[33] Gerald R McMenamin. *Forensic Linguistics: Advances in Forensic Stylistics*. CRC Press, 2002.

[34] NOS. De Toeslagenaffaire: van een miljoenen- naar een miljardenoperatie, January 2024.

[35] Nektaria Potha and Efstathios Stamatatos. A Profile-Based Method for Authorship Verification. In Aristidis Likas, Konstantinos Blekas, and Dimitris Kalles, editors, *Artificial Intelligence: Methods and Applications*, pages 313–326, Cham, 2014. Springer International Publishing.

[36] Nektaria Potha and Efstathios Stamatatos. Intrinsic Author Verification Using Topic Modeling. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, SETN '18, pages 1–7, New York, NY, USA, July 2018. Association for Computing Machinery.

[37] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier, August 2016. arXiv:1602.04938 [cs, stat].

[38] Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556, 2009. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.21001.

[39] Efstathios Stamatatos. Authorship Verification: A Review of Recent Advances. *Research in Computing Science*, 123(1):9–25, December 2016.

[40] Efstathios Stamatatos. Authorship Attribution Using Text Distortion. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1138–1149, Valencia, Spain, April 2017. Association for Computational Linguistics.

[41] Cor J Veenman and Zhenshi Li. Authorship Verification with Compression Features. page 6.

Appendices

A Function Words

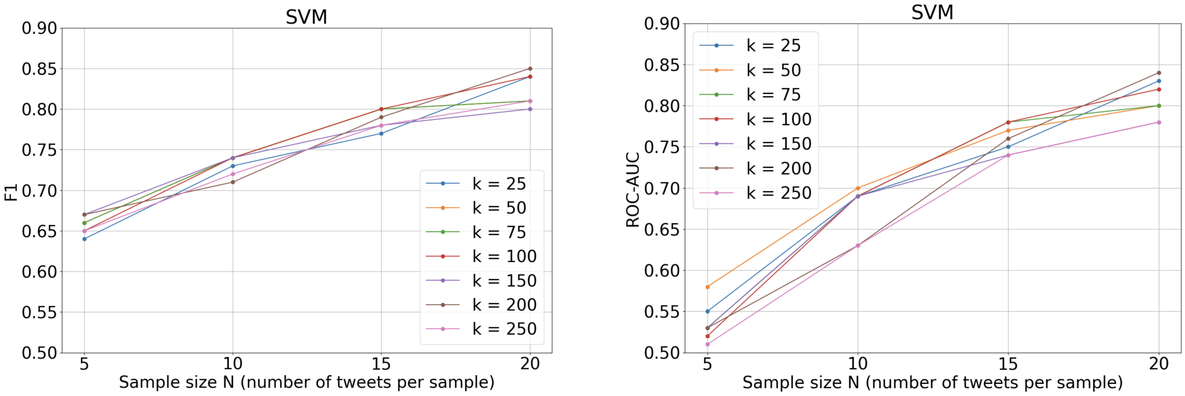
List of topic-agnostic function words used in experiments, as defined by Halvani [14]:

and
as
because
but
either
for
hence
however
if
neither
nor
once
a
an
both
each
either
every
no
other
our
some
above
across
after
among
below
beside
between
beyond
inside
outside
all
another
any
anyone
anything
everything
few
he
her
hers
herself
any
certain
each
either
few
less
lots
many
more
most
much
neither
can
could
might
must
ought
shall
will
get
go
take
make
do
have
give
set
do
did
does
got
getting
have
had
gives
giving
gave
give
gets
am
is
are
was
were
be
been
being
will
should
would
could
i'm
i'd
i'll
i've
he's
it's
we'd
she's
it'll
we're
how's
you're
almost
enough
hardly
just
nearly
quite
simply
so
too
again
always
never
normally
rarely
seldom
sometimes
usually
above
below
everywhere
here
in
inside
into
nowhere
out
outside
there
already
during
immediately
just
late
recently
still
then
sometimes
yet
hereafter
hereby
thereafter
thereby
therefore
therein
whereas
wherever
especially
mainly
particularly
generally
only
simply
exactly
merely
likewise
meanwhile
moreover
namely

nonetheless
otherwise
perhaps
rather
besides
furthermore
generally
hence
however
incidentally
subsequently
of course
as a result
in addition
because of
in contrast
on the other hand
as opposed to
in regard to
in relation to
inspite of
out of
with regard to

B Graphs

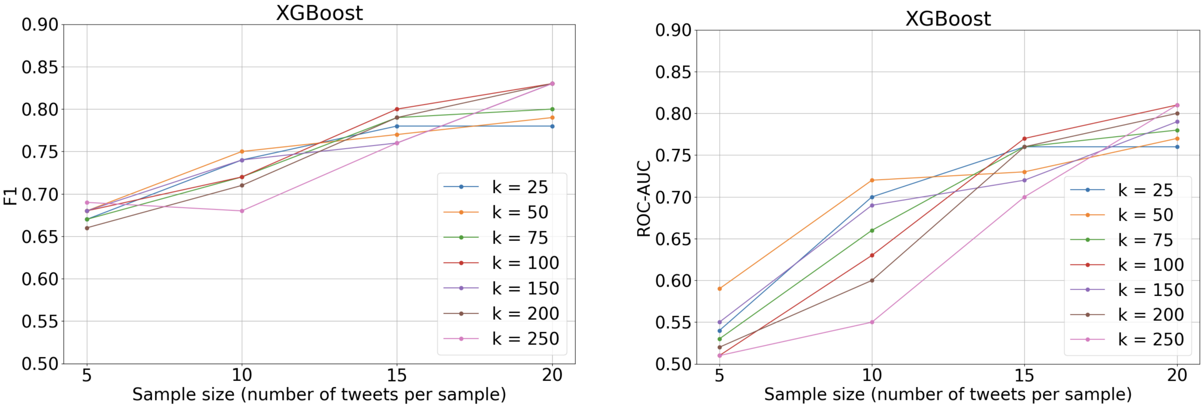
The graphs of SVM and XGBoost with varying lengths of N and k.



(a) F1

(b) ROC-AUC

Figure 6: Performance of the SVM classifier with varying values of N and k.



(a) F1

(b) ROC-AUC

Figure 7: Performance of the XGBoost classifier with varying values of N and k.