Evaluation Techniques for Authorship Attribution and Obfuscation

Malik Hashem Altakrori

School of Computer Science
McGill University
Montreal, Canada

November 2022

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

Evaluation Techniques for Authorship Attribution and Obfuscation

Malik H. Altakrori

Doctor of Philosophy

School of Computer Science

McGill University

2022

In this day and age, the vast majority of people have some form of online presence for work, social interaction, or both. Internet anonymity, where people can use pseudonyms or generic photos, allows people to share their opinions freely while protecting their privacy. Yet, this anonymity has been misused by maleficent people who hide behind it to commit their crimes. The topic of this thesis is evaluation techniques for authorship attribution and authorship obfuscation, two contrasting tasks that are closely related to the concept of anonymity. Authorship attribution, or identification, is the task of using the unique writing style of authors to identify the most plausible author of an anonymous text from a set of candidate authors of that text. Authorship obfuscation, on the other hand, is the task of manipulating a text to hide the author's **writing style** to prevent authorship attribution techniques from revealing the identity of that author. The importance of these two tasks is that while authorship attribution can be used to identify criminals, it can also be misused to identify whistle-blowers or persecute journalists who speak against authoritarian regimes. To fight this misuse, authorship obfuscation techniques are proposed. Aiming for better performance, most of the work on authorship attribution either proposes new writing style features or new attribution techniques. As a result, less attention is given to evaluation tasks that are important to better understand the authorship attribution problem. Similarly, existing research on anonymization either ignores the potential change in the document's

contents due to obfuscation or uses abstract and unintuitive measures to quantify such change. Indeed, achieving better performance is the ultimate goal. However, for tasks with critical real-life applications, we must know how the proposed techniques for such tasks work and to make sure that they work as intended. The contribution of this thesis is a set of evaluation techniques for authorship attribution and text obfuscation tasks. We believe the importance of evaluation techniques for these two closely related problems is two-fold: to help explain such techniques and their outcomes on downstream tasks and to direct researchers' efforts toward achieving higher performance using the findings of the proposed evaluation techniques. Firstly, I evaluate a large spectrum of existing English authorship attribution techniques on Arabic tweets. Given that such techniques were using hand-engineered features, I adapt existing features to the new language and domain. Moreover, I adapt a visualization tool to Arabic tweets to make the output of the attribution process more interpretable. Following that, I propose a new benchmark to investigate the negative effect that the topic of a document has on the attribution process. With this benchmark and two newly proposed error measures, I compare various writing style representations and measured their susceptibility to topic variations. Furthermore, I demonstrated the significance of using stylometric features as part of the writing style representation. Finally, given that authorship attribution techniques use topic cues to identify the authors, I investigate whether obfuscation techniques modify these topical cues to hide the author's identity in the obfuscated document. I re-evaluate state-of-the-art obfuscation tools on evasion, and content preservation, and propose a new dimension namely, misattribution. The results show that the performance of existing obfuscation techniques is highly overstated. In particular, these techniques are inferior to a simple back-translation baseline that achieves higher obfuscation performance, better content preservation, and lower misattribution.

Sommaire

Actuellement, la majorité des gens ont une certaine présence en ligne pour le travail, l'interaction sociale, ou les deux. L'anonymat sur Internet, qui permet d'utiliser des pseudonymes ou des photos génériques, permet aux gens de partager leurs opinions librement tout en protégeant leur vie privée. Pourtant, cet anonymat a été détourné par des personnes malveillantes qui s'en cachent pour commettre leurs crimes. Le sujet de cette thèse est l'évaluation des techniques d'attribution d'auteur et l'obscurcissement d'auteur, deux tâches opposées qui sont bien liées au concept d'anonymat. L'attribution d'auteur est la tâche qui consiste à utiliser les styles d'écriture uniques des auteurs pour identifier l'auteur le plus probable d'un texte anonyme parmi un ensemble de candidats d'auteurs de ce texte. L'obscurcissement de l'identité de l'auteur, quant à lui, consiste à manipuler un texte pour masquer le style d'écriture de l'auteur afin d'empêcher les techniques d'attribution d'identité de révéler l'identité de cet auteur. L'importance de ces deux tâches tient au fait que si l'attribution d'auteur peut être utilisée pour identifier des criminels, elle peut être également mal utilisée pour identifier des dénonciateurs ou persécuter des journalistes qui s'élèvent contre des régimes autoritaires. Pour lutter contre cette utilisation abusive, des techniques d'obscurcissement de la paternité sont proposées. Dans le but d'améliorer les performances, la plupart des travaux sur l'attribution d'auteur proposent soit de nouvelles caractéristiques de style d'écriture, soit de nouvelles techniques d'attribution. Par conséquent, moins d'attention est accordée aux tâches d'évaluation qui sont importantes pour mieux comprendre les problèmes d'attribution d'auteur et d'obfuscation. La contribution de cette thèse est un ensemble de techniques d'évaluation pour les tâches d'attribution d'auteur et d'obfuscation de texte. Nous pensons que l'importance des techniques d'évaluation pour ces deux problèmes bien liés est double: aider à expliquer ces techniques et leurs résultats sur des tâches en aval et orienter les efforts des chercheurs vers l'obtention de meilleures performances en utilisant les résultats des techniques d'évaluation pro-

posées. Premièrement, j'évalue un large éventail de techniques d'attribution d'auteur conçues pour l'anglais sur des tweets en arabe. Étant donné que ces techniques utilisent des caractéristiques définies manuellement, j'adapte les caractéristiques existantes à la nouvelle langue et au nouveau domaine ainsi qu'un outil de visualisation pour les tweets en arabe afin de rendre le résultat du processus d'attribution plus interprétable. Ensuite, je propose un nouveau repère pour étudier l'effet négatif que le sujet d'un document a sur le processus d'attribution. À l'aide de ce repère et de deux nouvelles mesures d'erreur nouvellement que je propose, j'ai comparé diverses représentations de styles d'écriture et mesuré leur sensibilité aux variations du sujet. En outre, j'ai démontré l'importance de l'utilisation de caractéristiques stylométriques dans la représentation du style d'écriture. Enfin, étant donné que les techniques d'attribution d'auteur d'un texte utilisent des indices thématiques pour identifier les auteurs, j'étudie si les techniques d'obscurcissement modifient ces indices thématiques pour cacher l'identité de l'auteur dans le document obscurci. Je réévalue les outils d'obfuscation les plus récents sur l'évasion et la préservation du contenu, et je propose une nouvelle dimension, à savoir la mésattribution. Les résultats montrent que les performances des techniques d'obfuscation existantes sont largement surestimées. En particulier, ces techniques sont inférieures à une simple base de rétro-traduction qui permet d'obtenir une meilleure performance d'obscurcissement, une meilleure préservation du contenu et une plus faible mésattribution.

"الحَـمْـدُ للّه رَبِ العَـالَمِـيـن"

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my two amazing supervisors, Prof. Benjamin Fung and Prof. Jackie Cheung for having me as their student. I am extremely lucky to have them as my supervisors where their mentorship, knowledge, and skills as well as their patience, kindness, and continuous support have transformed me into the person I am today. They have taught me how to be a better person and not only a better researcher.

To my committee members, Prof. David Meger and Prof. AJung Moon, and my examiners, Prof. Nicolas Papernot, Prof. Reihaneh Rabbany, and Prof. Fatiha Sadat, I am thankful for the valuable insights, discussions, and comments that they have provided throughout this journey.

I was extremely lucky to learn from some of the brilliant minds in the fields of Information Studies, Artificial Intelligence, Machine Learning, and NLP. I would like to thank all my professors: Prof. France Bouthillier, Prof. Catherine Guastavino, Prof. Jamshid Beheshti, Prof. Timothy O'Donnell, Prof. Joelle Pineau, Prof. Doina Precup, and Prof. Aaron Courville.

I would like to thank all the members of the Data Mining and Security (DMaS) lab, and the Computational Linguistics (CompLing) group, especially Jad Kabbara from CompLing, and Miles Q. Li from DMaS for all the help that they provided in addition to the rich discussions that we shared.

It gives me a huge sense of pride to be part of the RLLab which was and still is a home for so many brilliant minds. I had the chance to meet people that are so smart yet extremely humble and willing to help and provide their support unconditionally.

I have been very lucky to be surrounded by amazing people everywhere I looked

during my time at Mila, and I would like to express my gratitude to all those with whom I have crossed a path, exchanged a friendly nod, or discussed an issue on Slack during my studies. To all of you, thank you very much.

To my beloved wife, my closest friend, and the dearest of all. This Ph.D. would not have been possible without her next to me. Her endless support allowed me to focus on my Ph.D., and in my times of doubt, she kept me going. To her, I say: WE did it!

To my precious kids who have been with me since the beginning: Lyana, Hashem, and Omar who were the best excuse for my procrastination. Thank you for sacrificing your playtime for me to work. That's it! no more of that!

To my bigger family, my safety net. To my amazing father who is perhaps among the few fathers who literally had to climb mountains to go to (graduate) school. To my loving mother who got a university degree at times when getting a high-school certificate was considered a big achievement. You both instilled in me the dedication and curiosity to learn more. To my brothers: Mohannad, Husam, and Amir, I know I can always count on you.

Finally, the work in this thesis was generously supported by the Doctoral Scholarship from Fonds de Recherche du Quebec Nature et Technologies (FRQNT-275545), the Canada Research Chairs (CRC) Program (950-230623 and 950-232791), the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grants (RGPIN-2018-03872) and Collaborative Research and Training Experience Grants (CREATE -554764-2021), Canada CIFAR AI Chair program, and the Zayed University Research Incentive Fund (RIF) (R14025 and R13059).

To my parents, my wife, and my kids.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this day and age, the vast majority of people have some form of online presence for work, social interaction, or both. Internet anonymity, where people can use pseudonyms or generic photos, allows people to share their opinions freely while protecting their privacy. Yet, this anonymity has been misused by maleficent people who hide behind it to commit their crimes (Chaski, 2005).

The topic of this thesis is the evaluation of **authorship attribution** and **authorship obfuscation**, two contrasting tasks that are closely related to the concept of anonymity. Authorship attribution, or identification, is the task of using the unique **writing styles** of authors to identify the most plausible author of an anonymous text from a set of candidate authors of that text. Authorship obfuscation, on the other hand, is the task of manipulating a text to hide the author's **writing style** to prevent authorship identification techniques from revealing the identity of that author.

Initially, authorship attribution was used in the literacy domain to give credit and solve disputes over the true authors of famous literature work (Mosteller and Wallace, 1963). Currently, authorship attribution is being investigated for various applications in the computer science domain, especially in forensics where investigators aim to identify authors of electronic texts that are deemed harmful or threatening (Iqbal

et al., 2010a,b). However, these same techniques can be misused to identify whistle-blowers or journalists who speak against oppressing governments (Shetty et al., 2018). To counter this misuse, researchers proposed the authorship obfuscation task (Kacmarcik and Gamon, 2006). This task aims to protect the identity of an author by masking their writing style in a document to prevent authorship identification techniques from revealing the identity of that author (Bevendorff et al., 2019).

Early approaches to authorship attribution depended on manual inspection of the textual documents to identify the authors' writing patterns. The earliest (Holmes, 1998) attempt to quantify the writing style of authors is attributed to Augustus De Morgan (1882) who investigated word length, measured by the ratio of characters to words, as a unique writing style representation of authors. Mendenhall (1887) and Yule (1939) took inspiration from De Morgan and De Morgan, and showed that word lengths, word frequencies, as well as sentence lengths in words, are distinct among authors. However, extracting such features by hand was a tedious task due to having large amounts of text to analyze.

The first work that used a computational approach is (Mosteller and Wallace, 1963), which used the Naïve Bayes algorithm with the frequency of function words to identify the authors of the Federalist papers (Stamatatos, 2009). These papers are newspaper articles published between the years 1787 and 1788 by an anonymous group of authors using the same alias "Publius"(Juola, 2007). Since then, researchers have investigated various applications in the computer science domain, especially in forensics where investigators aim to identify authors of electronic texts that are deemed harmful or threatening.

Examples on these texts are extremists' websites (Abbasi and Chen, 2005b), source codes (Frantzeskou et al., 2007; Caliskan-Islam et al., 2015), emails (Iqbal et al., 2013; Ding et al., 2015), blog posts (Cavalcante et al., 2014), and social media posts such as tweets and Facebook posts (Koppel et al., 2009; Layton et al., 2010; Bhargava et al.,

2013). Lately, authorship attribution techniques have been investigated in new domains such as detecting machine-generated fake news (Schuster et al., 2020; Uchendu et al., 2020).

Alternatively, motivated by the real-life applications of authorship attribution, different elements of and constraints on the attribution process have been investigated, such as the number of candidate authors for an investigated document, the number of writing samples per author, and the type and length of each writing sample (Luyckx and Daelemans, 2011).

Authorship obfuscation is the inverse task to authorship attribution. The objective of this task is to protect the identity of an author of a document who wishes to be anonymous. Obfuscation techniques try to hide this identity by masking writing style features that an authorship attribution technique could potentially use to reveal the author's identity. Note that this task is different from anonymizing customers' or patients' data. In this problem, the assumption is that the authors did their best to hide their identity by not leaving identifying information in the documents and would use obfuscation techniques to hide identifying features of which they are not aware.

Working on obfuscation in parallel to working on authorship attribution is crucial for the following reasons. Firstly, the effectiveness of an obfuscation technique is measured by the decrease in the attribution accuracy on a dataset before and after the documents in the dataset are anonymized. Secondly, we can analyze the outcome of the obfuscation process to see which writing style features are more frequently masked to achieve better obfuscation performance. More importantly, since authorship attribution techniques can be misused, better obfuscation techniques are needed to help protect the identity of people.

1.1 An Argument for Evaluation Techniques

Among the first questions that one would ask while learning about authorship identification and obfuscation are *what is a writing style?* and *how do we quantify it?*

The earliest (Holmes, 1998) attempt to quantify the writing style of authors is attributed to Augustus De Morgan (1882) who investigated word length, measured by the ratio of characters to words, as a unique writing style representation of authors. A definition, however, can only be found later, particularly in (Holmes, 1998). According to Holmes, the writing style of an author is "*the unconscious aspect of their writing, of which the author is unaware. This aspect cannot be consciously manipulated, and has quantifiable features that maybe be unique to that author*".

One unique characteristic of the authorship identification and obfuscation problems is that a simple computationally-assisted approach would outperform humans by a large margin. The reason is that existing writing style features are a combination of local and global features in a document (Abbasi and Chen, 2005a). While a human maybe able to identify local features, such as a greeting line, a frequent type or a unique signature, it is hard for them to capture frequencies of characters, or estimate ratios of spaces to words. In addition, it is not possible for an author to identify their writing style since, by definition, they are not aware of what their writing style is. If a writing style feature was identified beforehand, e.g., average word length, and a human was asked to extract such a feature from all the documents by all of the candidate authors, it will be a long and tedious task (Mendenhall, 1887). This property imposes several challenges on these tasks. For example, we cannot ask humans to evaluate the performance of existing or novel techniques, we cannot ask them to attribute documents to authors to have a gold-standard dataset, and as a result, it is not possible to ask humans to identify what a unique writing style is, similar to what is done in most NLP tasks.

Numerous studies have emerged proposing new writing style representations, each

supported with some empirical proof to show that the new writing style representation leads to better authorship identification performance. The problem is, given the critical applications of both authorship identification and obfuscation, relying solely on the accuracy to evaluate these techniques does not give a complete picture of their performance. For example, if two techniques have the same accuracy they could be making errors on the same examples, or different ones. In that case, one has to investigate whether the error are made because the examples are challenging, or because of some other issue, e.g., bias in the training data.

When authorship attribution is used in criminal investigations, a domain expert would use the identification techniques to help law enforcement identify the most plausible author of an anonymous, threatening text (Ding et al., 2015; Rocha et al., 2016). Specifically, the role of this domain expert is to show the outcome of the attribution process, which includes presenting the most plausible author and the confidence in that outcome. For the jury members, on the other hand, they need to see high confidence in the output, e.g., high accuracy, in addition to understanding the authorship attribution technique enough to be able to trust its outcome. Understanding both authorship attribution techniques and their results is crucial because the outcome of the attribution process could be used as evidence in the courts of law and has to be explained to the jury members.

Traditionally, the evaluation of new authorship methods or writing style features for authorship attribution has been based on the difference in the accuracy either in the attribution process or in ablation studies. While this methodology enhanced the performance on the downstream task and helped answer *which* features perform well, there is a need for methods that can help us understand *why* certain features are performing better than others. Mainly, are these newly proposed features/techniques capturing the variations in the authors' writing styles, or simply better at picking the sub-topic cues in the collected samples for each author? This issue is critical for two

reasons. From the one hand, when a setup was proposed to prevent authorship attribution techniques from using these topic cues, the performances of these techniques dropped drastically (Stamatatos, 2013). This result shows that the performances of authorship attribution techniques is overstated. On the other hand, using such cues will lead to choosing a candidate author because of the topic similarity between the investigated text and that author's writing samples.

Furthermore, authorship attribution and obfuscation have critical real-life applications such as in the forensics domain (de Vel et al., 2001; Rocha et al., 2016). For such a critical application, little research is done on explaining authorship attribution techniques and their outcomes compared to achieving better performance. Because the result of the authorship attribution technique could decide the destiny of a suspect and be the difference between life or death, e.g., the case of the Unabomber (Leonard et al., 2016) and the Iranian programmer (Caliskan-Islam et al., 2015), more efforts should be put towards interpretability and explainability in authorship attribution.

An analogous issue in obfuscation is the potential change in the facts in the resulting anonymized document compared to the original one. Existing research on obfuscation either ignores these potential changes, or uses abstract measures to quantify them (Altakrori et al., 2022). The problem with using abstract measures, such as the change in a sentence embedding before and after obfuscation, is that they do not provide an explanation for "which" content is not preserved. This detail is essential for both researchers on obfuscation and users of such systems.

From a researchers' perspective, particularly those who developed an obfuscation technique, they need to know which facts were not preserved to investigate these instances in their models further or to inspect the evaluation measure to make sure it is working as intended. On the other hand, a system user needs to know if the message still delivers their key points, either fully or partially, and make the trade-off between anonymity and preserving the facts independently.

1.2 Thesis Statement

Authorship attribution and obfuscation are being used in critical applications, and their outcomes could have long-lasting effects on people's lives. Existing evaluation techniques show errors that different techniques make, but they do not explain how these errors are being made. In this thesis, I propose a set of evaluation techniques for the authorship attribution and obfuscation tasks. Specifically, I show how the topic affects authorship attribution models, and for obfuscation, I show what facts are being preserved and which techniques can have a smaller misattribution effect. In addition, I propose a new fairness-related measure, namely misattribution to characterize the framing side-effect of a successful obfuscation. Ultimately, this thesis aims to aid researchers to understand why existing models make such mistakes and to guide researchers toward both better performance and evaluation techniques.

1.3 Dissertation Objectives

The goal of this thesis is to provide tools for researchers on authorship attribution and authorship obfuscation to help them gain a better understanding of how existing techniques perform, then use these insights to achieve better performance on both tasks. These tools, therefore, comprise new datasets and new evaluation techniques. For this thesis, I will propose two new datasets for authorship attribution: one dataset is the first of its kind for Arabic short text, and the other is a curated and extended version of an existing dataset for the task of cross-topic authorship attribution.

In addition, I will propose two evaluation techniques, one for each task. Specifically, I will propose a new experimental setting to investigate the topic-writing style entanglement problem. For the authorship obfuscation task, I will propose the use of question-answering approaches to evaluating content preservation after obfuscating a document. I will additionally propose a new information-theoretic measure to charac-

terize *misattribution* in authorship obfuscation.

The objectives and contributions of this thesis are enumerated below.

1. **The first extensive authorship attribution study on Arabic short text, with visualized outcome**. I propose a new dataset of Arabic tweets and conduct an extensive, systematic study on existing authorship attribution techniques that were developed for English. Using the proposed dataset, I evaluate different authorship attribution approaches and show that existing techniques do not work out of the box for a new language or a new media. Instead, we need to adapt the writing style features from English to Arabic and from longer forms of text to fairly short Twitter posts.

2. **A novel evaluation task for authorship attribution**. This is the first work that aims to characterize the effect of the topic on the attribution task. Particularly, this setup allows us to probe the errors made by the authorship attribution techniques and use these errors to compare and rank different writing style features based on their susceptibility to topic variations. To demonstrate the effectiveness of this task, I draw on its conclusions to achieve new state-of-the-art results on authorship identification.

3. **Investigating the use of distributional word embeddings for writing style representation**. Using the task above, I show that word-level n-grams outperform pretrained embeddings from Transformer-based language models such as BERT and RoBERTa when used as features for cross-topic authorship attribution. This result is justified by how well such models capture the topic in a document, which results in a relatively high topic-related error.

4. **A multi-faceted evaluation framework for authorship obfuscation.** I evaluate existing obfuscation techniques on evasion and content preservation and show

that a neural text generation baseline can out-perform state-of-the-art obfuscation tools in both evaluation dimensions. Furthermore, I show that the generic writing style that is produced by text generation models makes it difficult for authorship attribution techniques to attribute the obfuscated text to any author. This behavior reduces the confidence in the outcome of authorship attribution techniques and, as a result, reduces the potential harm of misattribution.

1.4 Thesis Structure

The remaining six chapters of this thesis are structured as follows.

Chapter 2 provides a background on both authorship attribution and obfuscation. In this chapter, I describe all the elements of the authorship identification process starting with the writing style representation. In addition, I describe the approaches, evaluation scenarios, and the different constraints that aim to keep the evaluation settings similar to real-life applications of authorship attribution and obfuscation. For authorship obfuscation, I discuss the two groups under which such techniques fall and discuss existing content-preservation techniques. Finally, I provide a brief technical background on machine learning techniques that are used throughout this thesis.

In Chapter 3, I propose a new dataset for Arabic short text, specifically, Twitter posts, and evaluate existing authorship attribution techniques that were developed for English on Arabic short text. More concretely, I show the steps taken to adapt existing authorship techniques, which comprise writing style features and authorship attribution approaches, from English to Arabic short text. The evaluation includes both instance-based and profile-based approaches, with static and dynamic writing style feature representations. In addition, I demonstrate the use of a modified visualization tool that is based on the profile-based approach. This tool is used for Arabic tweets and presents the outcome of the attribution process clearly and intuitively.

In Chapter 4, I propose the topic confusion task. Using this task, I investigate the effect of the topic in an investigated document on the attribution process. I start by describing the process of collecting and curating the new Guarding dataset. Next, I explain the proposed setup for the topic confusion task and propose two errors that I use to probe the errors made by authorship attribution techniques.

In Chapter 5, I turn to the problem of authorship obfuscation and propose a multi-faceted evaluation of existing techniques. Specifically, I re-evaluate existing obfuscation baseline approaches with more recent, neurally-inspired techniques to set a more realistic baseline of what such generic obfuscation tools can do. In addition, I propose the use of question-answering-based approaches to evaluate content preservation after obfuscation as opposed to using outdated content preservation measures. Finally, I propose a new evaluation measure to characterize misattribution in obfuscation, which occurs when a successful obfuscation of a document results in attributing that document to another author.

Finally, Chapter 6 provides a general discussion on the results from all the previous chapters. Chapter 7 concludes this thesis by providing a discussion on the limitation and potential future directions.

1.5 Published Material

The main chapters of this dissertation are based on two published papers and one that is currently under review. All the venues are peer-reviewed.

- **Chapter 3**: Arabic authorship attribution for Twitter posts, cited as (Altakrori et al., 2018):

  Malik H. Altakrori, Farkhund Iqbal, Benjamin C. M. Fung, Steven H. H. Ding, and Abdallah Tubaishat. 2018. Arabic authorship attribution: An extensive study on Twitter posts. *ACM Transactions on Asian and Low-Resource Language Informa-*

tion Processing (TALLIP), 18(1):1–51.

Contributions of authors: Malik led the project, created and pre-processed the Twitter authorship dataset of Arabic tweets, modified the code and the authorship attribution algorithm to be used for the Arabic language, ran all the experiments and generated the results, and wrote the paper. Farkhund (co-PI from Zayed University) participated in forming the idea and the early discussions contributed to editing the paper and provided the funding. Benjamin (co-PI from McGill University) participated in formulating the idea, contributed to the discussions, gave continuous feedback on the paper and the discussion of results, and edited the paper. Steven provided his code of the authorship attribution algorithm that was developed for English emails. His work had been published at the time of use in this work. Abdallah (co-PI from Zayed University) participated in proposing the idea of the paper and provided the funding.

- Chapter 4: Evaluating the topic effect on the authorship attribution techniques, cited as (Altakrori et al., 2021):
  Malik H. Altakrori, Jackie Chi Kit Cheung, and Benjamin C. M. Fung. 2021. The Topic Confusion Task: A Novel Evaluation Scenario for Authorship Attribution. In *Findings of the 2021 Conf. on Empirical Methods in Natural Language Processing: EMNLP 2021*, pages 4242–4256, Punta Cana, Dominican Republic. Association for Computational Linguistics.

  Contributions of authors: Malik proposed the idea, led the project, collected the New Guardian Dataset, designed and executed the experiments, analyzed the results, and wrote the paper. Benjamin and Jackie participated in the discussion in all the different stages and edited the paper and provided the funding.

- Chapter 5: Proposing an multifaceted evaluation of obfuscation techniques, currently under review (Altakrori et al., 2022): Malik H. Altakrori, Thomas Scialom,

Benjamin C. M. Fung, and Jackie Chi Kit Cheung. 2022. A Multifaceted Framework to Evaluate Evasion, Content Preservation, and Misattribution in Authorship Obfuscation Techniques. "Under Review", In *Proc. of the 2022 Conf. on Empirical Methods in Natural Language Processing: EMNLP 2022*, 8 pages.

Contributions of authors: Malik proposed the idea, led the project, collected the New Guardian Dataset, designed and executed the experiments and collected the results, and wrote the paper. Benjamin and Jackie participated in the discussion in all the different stages and edited the paper. Thomas provided his code for QuestEval which was not published when the project started. He participated in the early discussions on content-preservation techniques. Both Benjamin and Jackie participated in shaping the idea, the discussions in all the different stages, and edited the paper. Jackie provided feedback on the human-evaluation study design and the analysis of its results. Benjamin provided the funding.

Chapter 2

Background

In this chapter, I provide an extensive background about writing style, authorship identification, and authorship obfuscation. Specifically, I start with a general overview of how the writing style started in the literature domain and evolved to its current state. Then, I move on to discuss authorship attribution, and its scenarios, approaches, and constraints. Next, I turn to authorship obfuscation and the two types of authorship obfuscation tools that can be used. I conclude this chapter with a technical background on machine learning and deep learning concepts that were used in this thesis and related work in the two aforementioned domains. Note that each subsequent chapter has its own related work section.

2.1 Writing Style Representation

The earliest attempt (Holmes, 1998) to describe the writing style is attributed to Augustus De Morgan (1882). A definition, however, can only be found later, particularly in (Holmes, 1998) where Holmes defines the writing style of an author as "*the unconscious aspect of their writing, of which the author is unaware. This aspect cannot be consciously manipulated, and has quantifiable features that maybe be unique to that author*". A similar

definition was proposed by Li et al. (2006) who was the first to coin the term *writeprint* which refers to "*the set of linguistic features that are most likely to reappear in an author's documents*".

In the literature, the term *writing style* has been commonly used to describe the variation between Early Modern English (Shakespearean) vs. Modern English (Xu et al., 2012; Jhamtani et al., 2017), formal vs. layman English (Cao et al., 2020), and different text attributes such as the sentiment (Lample et al., 2018). Note that, one may potentially see a hierarchical structure for these styles of writings where the highest level is the language of the generation which everyone uses and the lowest one is the author's writing style that is different from one author to another.

One of the key aspects of authorship attribution is quantifying the writing style to compare it for different authors. Historically, this has been done by looking at the existing, or absence of certain features in the writing samples of each author. Augustus De Morgan (1882) was the first to remark that differences in the word length can be observed between two different religious scriptures as an indication of variation in the writing style. The first *computationally-assisted* approach to authorship attribution is that of Mosteller and Wallace (1963) who used the frequency of function words with a Naive Bayes algorithm to investigate the authorship dispute over 12 essays from the total of 85 Federalist papers.

Since then, many writing style features and techniques were proposed to capture the writing style of authors, and this representation was either used to identify the author of the document, or modified to protect their identity. Below, I start by discussing the characteristics of the investigated text. Next, I explain the two main categories under which existing features fall, followed by a description of the characteristics of the investigated text. I reserve a special section on neurally-inspired approaches (See Section 2.2.3).

2.1.1 Characteristics of the Text

The literature shows that researchers are interested in the following three characteristics of the text: the domain, the language in which the text is written, and the text length. This is because these characteristics contribute to the selection of the feature set that will represent the authors' writing styles.

The Domain

As mentioned earlier, authorship attribution has been investigated in both literacy and various types of online discourse. In literacy, the analyzed books, plays, or poems are usually written in a formal, structured, and grammatical language. These properties allow the use of specific features such as rare words or the ratio of sentences to paragraphs. In contrast to literacy, social media content is short, informal, and has no restrictions on being grammatical. This deems using the features above less effective (Iqbal et al., 2013).

Because of that, some of the significant contributions in the authorship domains are creating a domain-specific set of features, such as the work of Silva et al. (2011) on Twitter where they focused on smileys and *Laughing Out Loud (LOL)* patterns and that of Layton et al. (2010) who analyzed user mentions "@" and hashtags "#".

The Language

The second property of text that has been in the attention circle of researchers is the language in which the text is written. That is because one cannot take a set of features that are known to work for English, for example, and directly apply it to another language regardless of how similar the two languages are (Abbasi and Chen, 2005b). For example, one commonly used set of features is the ratio of each alphabet to the whole text. If we use this set of features for French authorship attribution without accounting

for accents, for example, then we will be ignoring whether they use accents in their online communications, or not. This could potentially be an essential aspect of the author's writing style.

The Textual Length

Having *relatively* short text in authorship attribution makes the problem harder compared to having lengthy text. This is based on the hypothesis that longer texts will have better representations of an author's writing style, and a short anonymous text, such as "Have a nice day" is unlikely to have a unique writing style that can help identify its real author.

There is no specific length at which a text is considered short. This is because what is currently considered a lengthy text used to be considered short during the early stages of authorship attribution research. For example, when researchers started working on the attribution of blog posts, they were deemed short compared to books or plays. Later on, an email or a forum post with a couple of paragraphs was considered short. Currently, the term short is used to describe Short Messaging Service (SMS) texts and social media posts such as tweets and Facebook statuses.

To overcome such a challenge, Bhargava et al. (2013) proposed merging groups of tweets by the same author into artificial ones. Although this results in fewer writing samples, these artificial tweets will have richer content and increase the attribution performance.

2.1.2 Static Features

Static features are a set of predefined, hand-engineered features that were proposed by researchers to capture the writing style in documents. These features are considered static, in contrast to dynamic features (Sec. 2.1.3), because they remain fixed for any dataset on which authorship attribution is investigated. These static features are

grouped into three different categories (Li et al., 2006; Abbasi and Chen, 2006, 2008; Iqbal et al., 2008; Afroz et al., 2014): lexical, syntactic, and structural features. Throughout this work, we use the stylometric features that are shown in Table 2.1.

Lexical Features - Character-Level	**Lexical Features - Word-Level**
1. Characters count (N) 2. Ratio of digits to N 3. Ratio of letters to N 4. Ratio of uppercase letters to N 5. Ratio of tabs to N 6. Frequency of each alphabet (A-Z), ignoring case (26 features) 7. Frequency of special characters: <>%|{} []/\@#~ +-*=$^ &_()' (24 features).	1. Tokens count (T) 2. Average sentence length (in characters) 3. Average word length (in characters) 4. Ratio of alphabets to N 5. Ratio of short words to T (a short word has a length of 3 characters or less) 6. Ratio of word length to T. Example: 20% of the words are 7 characters long. (20 features) 7. Ratio of word types (the vocabulary set) to T
Syntactic Features	
1. Frequency of Punctuation: , . ? ! : ; ' " (8 features) 2. Frequency of each function word (O'Shea, 2013) (277 features)	

Table 2.1 List of stylometric features.

2.1.3 Dynamic Features

Dynamic features are more commonly known as n-gram features. They are considered dynamic because the actual features will be identified only after tokenizing the documents in the dataset and selecting all the n-grams that are more frequent than a specific threshold value. Using n-grams is a common approach to representing documents in authorship attribution (Kešelj et al., 2003; Stamatatos, 2013; Sapkota et al., 2014, 2015). For most text classification tasks, tokenization is done on either the word or the character level. For authorship attribution, however, Part-Of-Speech (POS)-level n-grams are also used for authorship attribution, and they have been proven to be an important indication of style (Sundararajan and Woodard, 2018).

2.2 Authorship Attribution

Authorship attribution (AA) is one of the three research problems under authorship analysis (Iqbal et al., 2020). These problems are authorship attribution, authorship verification (Iqbal et al., 2010b), and authorship characterization (Koppel et al., 2002, 2009; Iqbal et al., 2013).

One way to differentiate these problems is based on the size of the set of candidate authors (Stamatatos, 2009). In authorship attribution (or identification) there is a fairly small, closed set of candidate authors and the task is to identify the most plausible author of the investigated text from this candidate set. Authorship verification is a special case of authorship attribution where the set of candidate authors has only one author. In this case, the task becomes to verify whether the investigated text is written by that author or not. Finally, authorship profiling (or characterization) is used when the set of candidate authors is too big and cannot be identified, in which case researchers try to identify characteristics of the real author such as their age, gender, or education level.

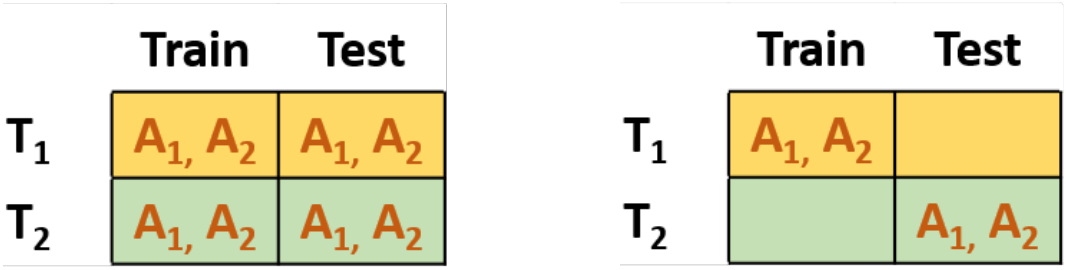
In this section, we start by discussing the process of quantifying an author's writing style, followed by the approaches to identifying the author based on the extracted writing styles. Next, we discuss the evaluation scenarios of authorship attribution. We conclude by discussing neural authorship attribution methods.

2.2.1 Authorship Attribution Scenarios

Previous authorship attribution evaluations can be classified into same-topic or cross-topic scenarios, depending on whether new, unseen topics are introduced at test time.

Blei (2012) defined a topic as "*a distribution over a fixed set words, that is specified before the data is generated.*". In authorship attribution, the topic is used informally to describe what a document is generally about, and it is common to only use the topic labels that are provided with the articles when collecting data for an authorship attribution

dataset. For example, the topics in the Guardian dataset (Stamatatos, 2013) are "UK", "Politics", "Society", and "World".



(a) Same-topic (b) Cross-topic

Figure 2.1 Authorship attribution scenarios. (T: Topic, A: Author)

Same-Topic Authorship Attribution

Figure 2.1a shows the same-topic scenario where all pairs of authors and topics in the test set are attested at training time. Researchers, however, consider this setup to be artificial (Koppel et al., 2009) as it is unlikely that writing samples of candidate authors across multiple topics would be available in real-life applications. As an example, the topic of a threatening message is quite different from people's daily life writings. Indeed, such setups are created by researchers who would choose a set of authors and collect a few writing samples for each one, shuffle these samples randomly, then split them into training, validation, and testing sets.

Same-topic authorship attribution has been investigated for different languages (Kešelj et al., 2003; Altheneyan and Menai, 2014; Ouamour and Sayoud, 2013; Ding et al., 2019; Sanchez-Perez et al., 2017; Markov et al., 2017a; Altakrori et al., 2018) and numerous applications in electronic communication media (Abbasi and Chen, 2005b, 2006; Frantzeskou et al., 2007; Iqbal et al., 2008, 2013; Koppel et al., 2009; Ding et al., 2015; Layton et al., 2010; Silva et al., 2011; Layton et al., 2012a; Bhargava et al., 2013; Schwartz et al., 2013; Schmid et al., 2015).

Cross-Topic Authorship Attribution

There have been more recent attempts to investigate authorship attribution in more realistic scenarios, and many studies emerged where the constraints differ from the training to the testing samples. Examples of these studies are (Bogdanova and Lazaridou, 2014) on cross-language, (Goldstein-Stewart et al., 2009; Custódio and Paraboni, 2019) on cross-domain/genre, and finally, (Mikros and Argiri, 2007; Overdorf and Greenstadt, 2016; Sundararajan and Woodard, 2018; Stamatatos, 2013, 2017, 2018; Barlas and Stamatatos, 2020) on cross-topic.

The cross-topic scenario (Stamatatos, 2013) is more realistic than the same-topic one where new, unseen topics are introduced in the test phase, as shown in Figure 2.1b. This task requires attribution techniques to focus on cues that capture the authors' writing style across topics. Compared to the same-topic scenario, the performance of well-known authorship methods using word- and character-level n-gram features is much lower, as they relied on topic-specific cues to infer authorship.

2.2.2 Approaches to Authorship Attribution

Stamatatos (2009) described three approaches to extract the writing style from writing samples. The first approach is instance-based, in which a writing style is extracted from every sample separately. By using this approach the candidate author will have s-styles, where s = number of writing samples per author. The second approach is profile-based in which all the writing samples for a particular author are used to generate one writing style for that author. Note that the profile-based approach is different from authorship profiling, where the task is to infer the characteristics of the author such as age, gender, education level, etc. The third approach is a hybrid one that starts as an instance-based one, then the features are aggregated over all the instances to create one profile per author.

The Instance-based Approach

The instance-based approach compares the investigated text to writing styles in other documents, so it assumes that each document is an example on its author's writing style with a small variation.

Supervised Techniques. This setup allows us to cast the attribution problem as a classification one where the writing samples are the training instances, the authors' names are the class labels, and the anonymous text is the test instance. A classification model, such as Support Vector Machines (SVM) or decision trees, is then trained on the writing samples using the extracted features to predict a class, i.e., an author for the anonymous text. If needed, the model's validation accuracy can be interpreted as a confidence indication. Examples of instance-based approaches are (Silva et al., 2011; Schwartz et al., 2013; Bhargava et al., 2013; Iqbal et al., 2013; Cavalcante et al., 2014; Ruder et al., 2016).

Unsupervised Techniques. Historically, unsupervised techniques such Principle Component Analysis (PCA) (Burrows, 1992) and topic modeling (Rosen-Zvi et al., 2004; Seroussi et al., 2014) were used for authorship attribution. For example, Burrows applied PCA to the frequency of words then plotted the first two components. The resulting graph was visually examined to identify clusters of words for each document and highlight the similarity between the documents. However, supervised methods were favored over such techniques because supervised techniques can already handle multidimensional representation better and do not require further inspection (Juola, 2008). Instead a prediction of the author is made directly.

The Profile-based Approach

In contrast to the instance-based approach, the profile-based approach combines the features' values from all the writing samples that belong to the same author to create one author's writing style. As a result, the anonymous text is compared to one writing style per author instead of one writing style per writing sample (Kešelj et al., 2003; Frantzeskou et al., 2007; Layton et al., 2012b). To compare the writing styles, one can use generative models such as Naïve Bayes, profiles overlap (Frantzeskou et al., 2007), or other distance metrics such as the Cosine similarity, the Euclidean distance, or the Kullback-Leibler (KL) divergence.

Instance-based vs. Profile-based

Stamatatos (2009) compared and contrasted the two approaches for attribution. He explained that both approaches can use similarity-based distance metrics and both have a low attribution (or classification) time. The main advantage of using the instance-based approach, according to Stamatatos (2009), is being able to use powerful machine learning algorithms such as Support Vector Machine (SVM) and decision trees for authorship classification. This, however, makes this approach expensive to train. In contrast, the profile-based approach does not need training and its outcome is easier to visualize since there is one writing style per author. Nevertheless, it is difficult for the profile-based approach to handle the combination of different stylometric features, such as the greeting line or signature which can be different from one instance to another.

Typically, one would choose the attribution approach or the classification technique based on the application. For example, if the goal is to use the outcome of the attribution process as evidence in the courts of law, then it is easier to show one writing style per author, and the similarity between the author's profiles and the writing sample, as opposed to showing multiple styles per author.

2.2.3 Recent, Neurally Inspired, Trends in Authorship Attribution

In this section, we focus on recent, neurally inspired work on authorship attribution. Neural networks are proven to be very successful in various domains where they achieve human-like performance. From the literature, we see that researchers took one of these two directions: the first is to use neural network techniques for classification in authorship attribution. Alternatively, the other direction is to use representation learning techniques to learn a new feature representation and then use it to represent the authors' writing styles in the attribution process.

Neural Networks as Classifiers

One way to interpret how a Neural Network works is that it learns an implicit representation for the data in the hidden layers, and then performs the classification at the output layer based on the learned abstract representation (Goodfellow et al., 2016). Researchers on authorship attribution experimented with the suitability of neural network for this task. For example, Ruder et al. (2016), Ge et al. (2016), Shrestha et al. (2017), Sari et al. (2017) Hitschler et al. (2017), and Ding et al. (2019) have all shown that their neurally inspired approaches achieve very high accuracy. Still, they require a large amount of data to train from scratch (Zhang et al., 2015) which makes them inapplicable to real-life scenarios with limited data (Luyckx and Daelemans, 2011).

Neural Representations for Authorship Attribution

Ding et al. (2019) proposed an algorithm to learn an explicit representation of the writing style for authorship attribution. In their work, they train three shallow neural networks to generate topical, lexical, character, and syntactic representations at the documents' level. The learned representation was used with different classification algorithms such as SVM to predict the author of a document. Posadas-Durán et al. (2017)

and Gómez-Adorno et al. (2018) managed to train the well-known document-to-vector (doc2vec) architecture proposed by Le and Mikolov (2014) on the Guardian dataset which contains very few samples.

Alternatively, Barlas and Stamatatos (2020, 2021) explored the widely used and massively pretrained transformer-based (Vaswani et al., 2017) language models for authorship attribution. Barlas and Stamatatos (2020) used the cross-topic dataset in (Goldstein-Stewart et al., 2009) with the classification model from (Bagnall, 2015) and a pretrained embeddings from ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019) and ULMFit (Howard and Ruder, 2018). The embeddings were not finetuned, but rather replaced the 1-hot representation for words in the attribution process.

2.2.4 Constraints on the Attribution Process

In this section, we discuss some constraints that are imposed on the attribution process by the applications in which attribution is used. In a lab setting, researchers may propose different assumptions that might not be very accurate, or realistic, to simplify a research problem that is too hard to solve in real-life (Luyckx and Daelemans, 2011). One example of these assumptions is the independence of the features that the Naïve Bayes algorithm assumes. Here, we discuss three assumptions that are commonly assumed about the number of authors and the writing samples.

The Size of the Candidate Set

Luyckx and Daelemans (2011) criticized research that assumes that the set of candidate authors contains a small number of authors (from two to ten), or that each candidate author has an abundant amount of writing samples that can be used to extract the authors' writing styles. The reason behind this criticism is that the attribution process performs well and achieves high accuracy in identifying the real authors in such ex-

perimental settings where this is not guaranteed in real-life scenarios. In fact, Luyckx and Daelemans (2011) show that the performance drops drastically as the number of authors increases beyond ten with a reasonable number of writing samples per author. There exist some studies on tens and hundreds of authors. However, they all assume the availability of a massive amount of writing samples for each author.

The Availability of the Writing Samples

Regarding the writing samples, one should account for issues related to their quantity and quality for each author. For example, having a varying number of samples per author while using an instance-based approach will introduce a class-imbalance problem, which in return may cause a bias in the classification algorithm towards the author with more samples. One potential domain where this behavior is observed is on social media where different people adapt different behaviors. For example, while one candidate author may be very active on Twitter and publish new content every day, others may be active once or twice a week, or a month.

Additionally, it should not be assumed that the writing samples all have similar lengths. Whether it is in web forums, emails, or tweets, writing samples can be as long as the specified limit (if any), or as short as "Thanks" or"Good morning!". The text length has effects on both the instance-based and the profile-based approaches. Some instances such as "Good Morning!" or "Have a nice weekend" are too generic and do not have enough content to convey a unique, stand-alone writing style. For the profile-based approach as well, having too many short writing samples makes it harder to create a single writing style that can be distinguished from those of other authors.

2.2.5 Visualization for Authorship Attribution

State-of-the-art attribution techniques that are known to have good accuracy are complex and/or hard to interpret or visualize. For example, consider an SVM model that maps the input into a new dimension when the input is nonlinearly separable. Such models cannot be visualized beyond 3-dimensions, where each dimension represents a feature. Since a typical attribution problem has much more than three attributes and hence requires more dimensions, it is impossible to visualize the writing samples of all the authors on a plane and the decision boundary that divides them. In contrast, a decision tree is easy to present, either as a tree or by converting it into a set of rules. However, decision trees do not perform as well as an SVM or a random forest.

We identified the work of (Kjell et al., 1994; Abbasi and Chen, 2006; Benjamin et al., 2014) and (Ding et al., 2015) to be the only work on visualization with authorship attribution. The problem with (Kjell et al., 1994) is that it produces 3D images, one image per author, to be used for author identification; however, these images are difficult to compare holistically. In contrast, (Benjamin et al., 2014) visualizes each feature separately and does not aggregate them to find the most plausible author. Instead, it leaves the decision for the user to find the most plausible author. The problem with this approach is that it causes the decision to be dependent on the user's understanding of the visualized figures.

Finally, Abbasi and Chen (2006) produces one graphical representation per feature, but this representation cannot scale up to a large number of features. Additionally, the authors highlighted a limitation of their approach by saying that its performance is constrained when used for text less than 30–40 words long. This limitation prevents its application to Twitter posts as tweets are naturally much shorter. Ding et al. (2015) provides a visualization technique that overcomes all these previous points. The algorithm provides the most plausible author and its confidence for this outcome, then motivates its findings to help the user understand the outcome.

2.3 Authorship Obfuscation

Authorship obfuscation (Kacmarcik and Gamon, 2006) techniques aim to hide an author's writing style which can be used by authorship identification tools to reveal the true identity of that author. Here, the assumption is that the author has already taken the precautions to hide their identity by removing any identifying information such as their name or address from the text. By using an obfuscation technique, a user aims to hide their writing habits which may or may not be known to them.

2.3.1 Tools for Authorship Obfuscation

Users typically have the following choices (Potthast et al., 2016) to obfuscate their writing style. The first is to manually rewrite the text with or without a reference writing style that they try to imitate (Brennan et al., 2012; Afroz et al., 2012, 2014). Alternatively, if a user is not sure of what is considered part of their writing style, then a computer-assisted tool that has a set of predefined writing-style features, such as Anonymouth (McDonald et al., 2012, 2013), can be used to show the users potential parts in their text that are used to predict their identity. Then, a user would modify that part and observe a drop in the identification performance using that tool. Finally, obfuscation can be automated using two groups of tools: generic off-the-shelf tools, and application-specific obfuscation tools. Below, I discuss each one of these groups with examples of each one.

Generic Tools

Examples of off-the-shelf tools include machine translation, paraphrasing, and data augmentation approaches (Khosmood and Levinson, 2010; Khosmood, 2012; Mansoorizadeh et al., 2016). These tools have been adapted for the purpose of generating a slightly modified version of a document. Commonly, these tools are used as baselines to

be compared against obfuscation-specific techniques (Brennan et al., 2012; Keswani et al., 2016) because they are easy to use, require no further training or extra data from the user, and need minimal knowledge about the obfuscation process. One example is (Mansoorizadeh et al., 2016) which removes random words from a sentence then uses the BERT model to fill-in the blank. This approach mitigates the issue of inserting words that do not fit the context in the sentence. Table 2.2 is another example of these tools where translating a sentence into different languages and then back to the original one creates a modified version of the original sentence (Rao and Rohatgi, 2000).

Text	Language	
How is it going bro	-	En
Wie geht es dir, Bruder?	En	De
Comment vas-tu mon frére?	De	Fr
How are you my brother?	Fr	En

Table 2.2 Back translation is a technique used to paraphrase a sentence by translating it to different languages and then back to the original language.

When machine translation approaches were initially used, only statistical machine translation (SMT) methods such as Moses (Koehn et al., 2007) and Google's previous Translate API (Wu et al., 2016) were available. They were shown to suffer from low obfuscation effectiveness and generated text with poor linguistic fluency compared to obfuscation techniques. In contrast, more recent neural machine translation approaches are able to generate higher-quality translations compared to SMT approaches according to some evaluation metrics (Wu et al., 2016). This development warrants re-evaluating their performance, especially as both Brennan et al. (2012) and Keswani et al. (2016) used SMT approaches.

Obfuscation-specific Tools

In contrast, obfuscation tools are built specifically to hide the author's identity and are tested against state-of-the-art authorship attribution techniques. While these tools

require further training and/or additional data, they have been shown to be more effective than generic tools. Examples on such approaches are (Mahmood et al., 2019) and (Bevendorff et al., 2019, 2020).

(Mahmood et al., 2019) replaces words based on their GloVE word embeddings given that the candidate replacement has the same sentiment. This technique, however, requires knowledge of the authorship attribution classifier and uses that knowledge as a heuristic to decide when to stop replacing words. Alternatively, (Bevendorff et al., 2019) requires a target author profile that is based on tri-gram frequencies. This rule-based approach changes the text while incurring costs, and the goal is to generate a document with high similarity to a target profile with minimum cost.

Other than the generic and the obfuscation-sepcific tools, there exists another category of approaches where the obfuscation is done on the representation level, so no change is done on the original document. Since the original text remains intact, there is no point in evaluating the content preservation in these techniques. An example of this work is Weggenmann and Kerschbaum (2018). This category is beyond the scope of this thesis.

Finally, neural-based, obfuscation-specific approaches, e.g., (Emmery et al., 2018; Bo et al., 2021), are still deemed impractical for the authorship obfuscation domain where researchers would attribute this impracticality to the lack of large training datasets which these neural approaches require (Bevendorff et al., 2020).

2.3.2 Evaluating Obfuscation Techniques

The evaluation of an obfuscation technique is based on its ability to evade detection by an authorship attribution technique. With that in mind, it is still important that the obfuscated text contains the same conveyed message after obfuscation. Below, I discuss the evaluation techniques for both aspects: evasion and content preservation.

Evaluating Evasion

Authorship identification techniques are used to evaluate the performance of authorship obfuscation techniques. If the identification technique was able to identify the original author before obfuscation but failed to identify that author after obfuscation then the obfuscated document has evaded detection. Examples of authorship identification techniques were described earlier in Section 2.2.2.

The evaluation process is as follows. We start by training and tuning an authorship identification tool on the training and validation documents, respectively. Then, we record the identification accuracy on the original test documents. Next, we use an obfuscation technique to modify the test documents to hide the authors' writing styles in these documents. Finally, without further training/fine-tuning the identification tool, we measure the authorship identification performance on the obfuscated test documents. The effectiveness of an obfuscation technique is quantified by the difference in identification performance before and after obfuscation over all the test documents in the investigated dataset.

Evaluating Content Preservation

Evaluating content preservation in text is important even if we value safety (Potthast et al., 2016). This is because people want to maintain their privacy while sharing their opinions freely. Besides obfuscation, content evaluation techniques are applicable to other NLG tasks such as machine translation and summarization, and these techniques fall within one of three groups: token-based, model-based, and question-answering-based models.

Token-based evaluation metrics depend on the token overlap between a source and a target document. Examples of these metrics are BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and ROUGE-L (Lin, 2004).

Both BLEU and METEOR are used in the evaluation of machine translation. While

both measures use tokens overlap, METEOR apply an alignment between the compared texts and uses stemming as part of the alignment process. ROUGE, on the other hand, is used in the summarization task and has multiple variants based on the size of the token in n-grams. ROUGE-1 and ROUGE-2, for example, compare two texts using uni-grams and bi-grams, respectively.

While these metrics are among the early ones to be used, Maynez et al. (2020); Honovich et al. (2021) showed that these measures have a lower correlation with human scores for fact preserving in text summarization.

With recent advances in representation learning, particularly in word and sentence embeddings, new model-based metrics were adapted where a smaller change in the sentence embedding indicates higher content preservation. Examples of such metrics are the Universal Sentence Encoder (USE) (Cer et al., 2018) and BERTScore (Zhang et al., 2020) which uses the cosine similarity to calculate a score after generating the sentence embeddings for the text to be compared. Here, a higher similarity score is considered an indication of better content preservation.

More recently, the summarization community proposed a new, question-answering-based approach to evaluate content preservation in summarization. The argument for this work is that the content is considered preserved if we can give the same answer to a particular question both before and after summarization. An example of this work is (Scialom et al., 2021) that calculates a content preservation score without requiring a reference summary.

2.4 Technical Background

In this section, I briefly discuss the basic concepts behind the classification techniques and algorithms that were used in this thesis in the authorship attribution problems, and to evaluate the evasion performance of obfuscation techniques. I use "training

instances" to refer to the authors' writing samples, and "test instances" to refer to the anonymous text.

2.4.1 Instance-Based Learning

The name "Instance-Based" (IB) could be a bit confusing since all classifiers use instances to build classification models. This classification technique stores all the instances and uses them during the classification, or the testing phase, and hence, the name Lazy Learners (Han and Kamber, 2001). Examples of IB learners are Instance-Based K-nearest neighbors (KNN) and Voting Feature Intervals (VFI) (Demiröz and Güvenir, 1997).

In KNN, a test instance is compared to the other training instances using a distance metric such as the Euclidean, or the Manhattan Distance. If K=1, the test instance will be labeled with the same class as its nearest neighbor. If K $>$ 1, a majority voting is used to pick the label.

In contrast to KNN where each one of the K nearest instances votes for a class, VFI asks every feature, or attribute, to vote for a label. To do that, local voting is performed within the intervals of each feature. The interval within which the test instance falls nominates the class that has a majority in that interval. The class with the most votes among the features is selected and associated with the test instance.

2.4.2 Probabilistic Approaches

One way to look at the classification problem is by calculating the conditional probability of class y given a test sample x, i.e., $P(y|x)$ for all the possible class values then picking the class with the highest probability.

There exist two approaches to calculating this conditional probability. The first one is a discriminative approach where the probability $P(y|x)$ is directly estimated. An example of this approach is **Logistic Regression**. The second approach is a generative

one which uses Bayes rule, given by Eq. 2.1, to model $P(x|y)$ and $P(y)$ then estimate $P(y|x)$. An example of such an approach is **Naïve Bayes**.

$$P(y = c|x) = \frac{P(x|y = c)P(y = c)}{p(x)}, \tag{2.1}$$

where c is one of the classes.

Logistic Regression

Logistic regression (LR) is a discriminative technique for estimating the conditional probability $P(y|x)$. This algorithm estimates a linear decision boundary by setting the log odds to zero, then applies a soft threshold function, specifically, the Sigmoid (σ) to squash the output between zero and one. The parameters for this decision boundary are estimated using an optimization technique such as Gradient Descent. The formula for the logistic function for a binary classification task is given by Eq. 2.2.

$$P(y = 1|x_i) = \sigma(w^T x_i), \tag{2.2}$$

where

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

Naïve Bayes

A Nïve Bayes classifier is a generative approach that applies the Bayes rule of conditional probability (Witten and Frank, 2005) to predict the class of a given instance by estimating $P(x|y)$ and $P(y)$. In terms of computational complexity, a data set with many attributes that are dependent on each other would be too exhaustive for the computer resources (Han and Kamber, 2001). Therefore, to simplify this process, Naïve Bayes assumes that all the attributes are independent and are of the same weight. Because this is not true in most real-world scenarios, the term "Naïve" was associated with this clas-

sifier. Given n attributes, Naïve Bayes assumes that all the attributes are independent as shown in Eq. 2.3.

$$P(x|y) = P(x^1, x^2, ...x^n|y),$$

$$P(x|y) = P(x^1|y)P(x^2|y, x^1)P(x^n|y, x^1, x^2, ...x^{n-1}).$$

Using the Naïve assumption:

$$P(x|y) = P(x^1|y)P(x^2|y)P(x^n|y), \tag{2.3}$$

where x^i is attribute i in instance x.

2.4.3 Support Vector Machines (SVM)

An SVM is a supervised learning (Boser et al., 1992) algorithm that extends a linear classification model to solve a multi-class, linear, or nonlinear classification problem where the n-attributes are mapped to an n-dimensional plane with n-axes (Witten and Frank, 2005). Such models use various optimization techniques (Witten and Frank, 2005) to find the *Maximum Marginal Hyperplane* (MMH) (Han and Kamber, 2001) that can separate these instances. This makes SVM models slow and computationally expensive (Boser et al., 1992), but very accurate (Witten and Frank, 2005) as they always provide the global solution (Han and Kamber, 2001).

2.4.4 Decision Trees

Decision Trees are rule-based techniques that use if-then-else rules to separate the output space into regions where (James et al., 2013) instances are classified based on the majority of the classes in the region in which they appear. Initially, Quinlan (1993) introduced the ID3 algorithm that uses **Information Gain** as an attribute splitter (Witten and Frank, 2005). C4.5 is the result of a number of improvements applied to ID3,

among them is using the gain ratio instead of the Information Gain and using **pruning** to remove the "unreliable" branches caused by noise, or due to over-fitting (Han and Kamber, 2001; Witten and Frank, 2005), as well as dealing with instances with missing values or numerical attributes (Witten and Frank, 2005).

2.4.5 Random Forests (RF)

RF (Breiman, 2001), an example of **Ensemble Learners**, is a technique that utilizes bagging and randomization to produce a classification model that outperforms these individual classifiers (Witten and Frank, 2005). While bagging is performed using decision trees a number of times (instead of using different classifiers), random splitting is utilized when an attribute is to be chosen in each iteration of the tree induction process. This random attribute is selected from the N best attributes instead of the single "best" attribute (Breiman, 2001).

2.4.6 Neural Networks

Bishop (2006) defined a Neural network as a multi-layer logistic regression algorithm. The most basic neural network, also known as a **feedforward** neural network, has three layers: Input, Hidden, and output. The input layer has D units, or nodes, where D equals the number of attributes for the training instances, and the output layer has K nodes where K equals the number of classes. The number of the hidden units M is a design choice, but a common suggestion is to use at least 2 * the input (D).

Training a neural network is done in two steps: a forward step, where the instances are fed to the network and a prediction is made, and a **backpropagation** step where the error in prediction is estimated using a cost function and propagated back to update the weights accordingly. For brevity, I provide the formula to all three layers in one function as shown in Eq. 2.4

$$y_k(x, w) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \tag{2.4}$$

In Eq. 4, w is the weight (or parameter), the superscript indicates the layer number, where layer 1 is the input layer, and the subscript indicates the direction of the connection (to–from). For example, $w^(1)_j i$ is the weight in layer 1 which connects between node i in the input layer and node j in the hidden layer. w_{j0}^1 is the bias in the input layer, h(·) is an activation function. y_k is the output at node k of the output layer.

If a feedforward neural net is to be used for images, then an image with 32 * 32 pixels will require 1024 input nodes, and the classification will be depending on all the 1024 pixels. However, the main element in the image probably occupies a partial region of the image, not all of it. Convolution Neural Nets (ConvNets) (LeCun et al., 1989) use filters (as weights) and convolution to detect these shapes in an image, then use them for classification. This can be viewed as dividing the image into sub-images or regions and using each region as a feature in the next hidden layer (where the neural net has more than one hidden layer, as opposed to the earlier example).

For text, an additional step is necessary to convert a document to a 2-dimensional matrix. A number of approaches were used for this step, specifically by deciding on a vocabulary set, then treating each word as a row represented with a fixed size representation. Different representations were used such as one-hot encoding, pretrained embeddings, or random values as discussed previously in Sec. 2.2.3.

Chapter 3

Arabic Authorship Attribution: An Extensive Study on Twitter Posts

I start this thesis with evaluating existing techniques on new languages and domains. As discussed later in Section 3.2, most of the work on authorship attribution is on English language media. Arabic, on the other hand, is one of the mainstream languages, yet little work has been done on Arabic authorship attribution. Given that Arabic has different characteristics compared to English, I start by evaluating existing authorship attribution techniques that were developed for English on a new dataset of Arabic tweets that I collected for this purpose.

3.1 Introduction

Criminals are exploiting the anonymous nature of the Internet to covertly commit crimes online. These perpetrators create fake internet IDs while conducting illegitimate, malicious social media communications or spreading hate-speech. Authorship analysis techniques have been successful (Iqbal et al., 2008; de Vel et al., 2001) in defending users against such attacks by showing that authors use similar writing styles

across different accounts. By doing this, they are addressing the misuse of anonymity without sacrificing the privacy of other Internet users.

Authorship attribution helps identify the original author of a given anonymous text by extracting and analyzing author-specific writing style features (Stamatatos, 2009). Initially, authorship attribution was used in the field of literature to identify the original authors of novels, plays, or poems (Juola, 2006; Koppel et al., 2009; Stamatatos, 2009). Later, its applications have been extended to forensics by investigating the true authors of malicious texts and/or using the analysis results as evidence in courts of law (Chaski, 2005). Various computational techniques have been proposed for different languages and types of text such as Twitter posts, Facebook status, Short Message Service (SMS) messages, or chat conversations.

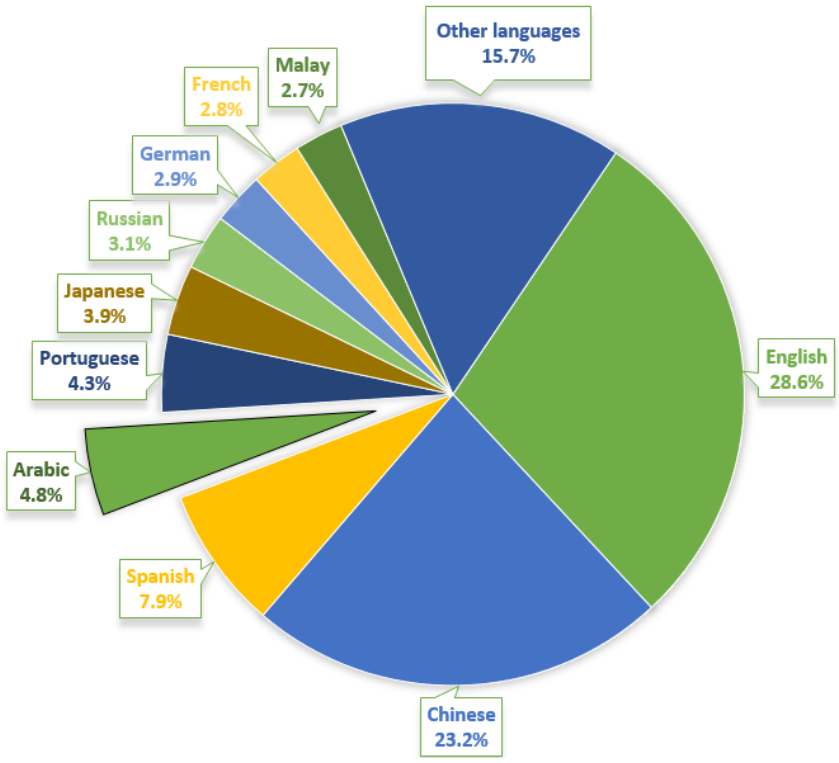


Figure 3.1 Top ten languages used over the Internet in terms of percentage of users according to Miniwatts Marketing Group (2013).

Compared to the large body of authorship attribution research for popular languages such as English (Luyckx and Daelemans, 2008, 2011; Frantzeskou et al., 2007) and Chinese (Keselj et al., 2003; Zheng et al., 2006), only around 10 studies are dedicated to Arabic authorship analysis (Abbasi and Chen, 2005b; Altheneyan and Menai, 2014; Kumar and Chaurasia, 2012; Ouamour and Sayoud, 2013; Shaker and Corne, 2010; Alwajeeh et al., 2014). However, Arabic is the 4^{th} most popular language used over the Internet after English, Chinese, and Spanish (see Figure 3.1). It accounts for 4.8% of the total number of Internet users [1]. The research literature shows that researchers mostly direct their efforts toward English, with little work being done on other languages and Arabic is one example. This lack of research is attributed to the various challenges that face researchers when they analyze an Arabic document.

For example, techniques that are developed for English may not be directly applicable to other languages that do not share many strong similarity to English. As noted in Abbasi and Chen (2005b), Arabic has three characteristics that are different from English, namely word length and elongation, inflection, and diacritics. These characteristics prevent the direct application of English authorship analysis techniques because of their effect on the feature extraction process. For instance, a feature representation that was designed for English language will not include the ratio of elongation to alphabets because elongation does not appear in English. On the other hand, because elongation is used for Arabic, one needs to empirically investigate whether these features make a significant contribution to the authors' writing style or not.

Moreover, among these few relevant studies on Arabic, none focus on Arabic short text such as social media posts, chat logs, and emails. Cybercrime investigators must frequently deal with these kinds of short texts. Existing Arabic authorship studies assume that there are plenty of text data for authorship analysis. Specialized methods have been proposed for novels (Kumar and Chaurasia, 2012), books (Ouamour and

[1] In 2020, Arabic is still in the forth place, but with 5.2% of the total number of Internet users.

Sayoud, 2013; Altheneyan and Menai, 2014; Shaker and Corne, 2010), articles (Alwajeeh et al., 2014), and the combination of forum messages (Abbasi and Chen, 2005b). However, this may not be the case in real-life applications. Forensic investigators may have only a limited set of samples to be analyzed, especially for cybercrime investigations. Authorship attribution over short text is more challenging due to the limited information carried by the text. Even for English, the length of the text sample has a significant impact on the attribution result. Specialized techniques are needed for the short text scenario (Iqbal et al., 2008).

Furthermore, little focus has been given to visualizing the results beyond providing accuracy and a confidence value. For English, only three techniques (Kjell et al., 1994; Abbasi and Chen, 2006; Benjamin et al., 2014) have been proposed for this purpose. For Arabic, no techniques have been adapted or proposed. Classification techniques that are known to produce the best results are mostly black-box methods that are hard to interpret. As an example, Support Vector Machines (SVM) and Random Forests are known to produce good accuracy, yet the results are difficult to interpret or explain. In contrast, while it is easier to visualize the rules of a decision tree, such methods are commonly outperformed by a SVM classifier. Law enforcement agents use the skills of an expert witness, such as an authorship attribution expert or a linguistics expert. The role of these experts is to help law enforcement officers narrow down the number of potential suspects, or provide evidence to justify a conclusion in a court of law. In order for experts to perform their role properly, they have to find the most plausible author from the candidate authors and show how they reached their conclusion in a clear and presentable way. To do so, they must use a classification model that provides high accuracy as well as being easy to present and explain, as opposed to using a model that is vague or complex, even if they have to sacrifice the performance.

We address the aforementioned research gap by focusing on Arabic short texts from

the social media platform Twitter[2]. We customize and apply existing Arabic authorship attribution techniques on Twitter data. We also adapt an English visualizable attribution technique based on n-gram to Arabic. We compare their performance on Twitter posts and report our findings. To our best knowledge, this is the first work to report the performance of authorship attribution on short Arabic texts in general. Moreover, this is the first work to adapt a visualizable approach for Arabic authorship attribution.

3.1.1 Problem Statement

We first provide an informal description of the authorship attribution problem in short text, followed by a formal problem definition. Given a set of candidate authors of an anonymous, relatively short, text and a set of sample writings for each one of the candidate authors, an authorship attribution expert analyzes the anonymous text and the sample writings of each author to capture the writing styles of the anonymous text as well as the sample writings of each candidate author. Based on that, the authorship attribution expert identifies the most plausible author of the anonymous text as the author whose writing style has the highest similarity to the writing style captured from the anonymous text. As mentioned earlier, this work addresses the authorship attribution problem in Arabic text, specifically, Arabic tweets. However, for brevity, we drop the word "Arabic" from "Arabic writing samples" and "Arabic tweets". Furthermore, we use the terms "a writing sample" and "tweet" interchangeably throughout the paper to refer to the same unit of text.

Formally, let $\mathcal{C} = \{c_1, \ldots, c_n\}$ be the set of candidate authors of an Arabic anonymous text a and $\mathcal{W}_i = \{w_1, \ldots, w_m\}$ be a relatively large collection of sample writings that belong to candidate author $c_i \in \mathcal{C}$. Finally, let $f(a, \mathcal{W}_i)$ be a function that computes the similarity between the writing style of the anonymous text a and the set of writing samples $\mathcal{W}_i$. The *problem of authorship attribution* is to identify the most plausible author

[2] www.twitter.com

c_i from the set of candidate authors $\mathcal{C}$, where $\forall\ c_i, c_j \in \mathcal{C}$ and $c_i \neq c_j$, $f(a, \mathcal{W}_i) > f(a, \mathcal{W}_j)$.

3.1.2 Research Questions

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

1. *How does the n-gram approach perform compared to state-of-the-art instance-based classification techniques under varying attribution scenarios?* A typical authorship attribution problem has three factors that affect the performance, namely, the number of candidate authors, the number of writing samples per candidate author, the length of the writing samples, and the anonymous text under investigation. We benchmark the state-of-the-art instance-based classification techniques, such as Naïve Bayes (NB), Support Vector Machines (SVM), Decision Trees, and Random Forests (RF), with stylometric features. Then we compare their performance with the n-gram approach under varying attribution factors.

2. *Which n-gram level (character, word, or syntactic) is the most helpful in distinguishing the authors' writing styles?* The use of the n-gram approach in authorship analysis has not been studied for Arabic short text. In this paper, we will investigate the performance of the n-gram approach on the characters, word, and syntactic levels. For the syntactic level, we use part-of-speech (POS) n-grams.

3. *How important are diacritics to the attribution process when the n-gram approach is used?* Diacritics in Arabic appear on the character level and their presences, while being optional, may affect the meaning of a word. For example, the word "كتب" can be read as "كَتَبَ = he wrote" or "كُتُب = books". The fact that their presence could change a word's meaning is one of the morphological properties that make Arabic text different from English text. We will compare the performance of the

attribution process using n-grams before and after removing the diacritics from the text.

4. *When using instance-based classification techniques, how important is it to use all three categories of stylometric features?* There are three categories of stylometric features: lexical, structural, and syntactic. They have been intensively studied for authorship analysis, especially in English. For the sake of completeness, we investigate the importance of each category for the attribution process in Arabic.

3.1.3 Contribution

The contribution of this work is summarized as follows:

- *Providing the first extensive authorship study on Arabic tweets.* To the best of our knowledge, this is the first work that investigates authorship attribution on Arabic short text in general and Arabic tweets in specific. We conduct a systematic evaluation for various attribution techniques on Arabic tweets and make the dataset publicly available [3] Not only does this open the door for further investigation of other authorship attribution techniques to be used for Arabic, but can also serve as a benchmark of experimental results for future work.

- *Providing an interactive system to visualize the attribution process and results.* Our work on Arabic authorship analysis is a significant extension of (Ding et al., 2015), which supports only English. One main application of authorship attribution techniques is to use the results and analysis as evidence in the context of criminal investigation; therefore, the interpretability of the results is as important as high accuracy. We have adapted the original models that were developed specifically

[3]Due to Twitter regulations for developers (Twitter, 2017), we cannot explicitly share the actual text for each tweet. Instead, a list of Ids and a Python script to crawl the tweets are provided via http://dmas.lab.mcgill.ca/data/Arabic_Twitter_dataset.zip. While this is the common practice in the research community (Taylor, 2017), This process may result in a different dataset if some accounts or tweets were deleted by their original authors.

for English and proposed using a language detection tool to automate selecting between Arabic and English; we added Stanford's part-of-speech tagger for Arabic and we changed the forms' orientation to show English or Arabic. With these modifications, the tool is able to visualize the authorship attribution evidence for the Arabic language in an intuitive and convincing style.

The rest of the paper is organized as follows: Section 3.2 presents a comprehensive review of authorship attribution work on Arabic in general and short English text. In Section 3.4, we describe our experimental design, followed by the results and discussion in Section 3.5. Finally, Section 3.7 concludes the paper.

3.2 Related Work

This research targets authorship attribution for short Arabic messages, specifically, Arabic tweets. Although Arabic is widely used over the Internet (Shaker and Corne, 2010), literature shows that authorship attribution research focuses on English, while research on Arabic is scarce. To the best of our knowledge, none of the available work on Arabic targets short text. The term "short" was formerly used to describe one page or a blog post; currently, the term is used to describe much shorter forms of text, e.g., a Facebook status, a Twitter post, an SMS message, or an Instant chat Message (IM) (Layton et al., 2010).

We start by reviewing the techniques used for authorship attribution on non-Arabic short text and then review the work done on Arabic text, regardless of the text length. While doing this, we keep in mind the difference between Arabic and English as detailed in (Abbasi and Chen, 2005b). According to Abbasi and Chen (2005b), authorship attribution techniques developed for English cannot be directly applied to Arabic text without modifications since Arabic and English languages have different language properties.

3.2.1 Authorship Attribution on Non-Arabic Short Text

Authorship attribution on non-Arabic short text has addressed different social media, e.g., SMS messages, chat logs, and Twitter posts. In reviewing the work on these topics, we look at the features and the classification techniques. For a recent, and more comprehensive review of existing authorship attribution literature on Arabic text see (Alqahtani and Dohler, 2022).

Chat Logs

One area in which authorship attribution on short text is investigated is chat conversations, such as Internet Relay Chat (IRC) and online chat rooms. Examples of the work done on these domains are (Inches et al., 2013) and (Layton et al., 2012a).

Inches et al. (2013) used word-level n-grams as features and statistical modeling, namely X^2 and Kullback-Leibler divergence, to compute the similarity between a candidate author and a query text. They started by generating an author profile for each author, which they did in two steps. The first step is concatenating all the text generated by a single user into one document. The next step divides the text into vectors by using "*non-letter* characters" as tokens, or stop marks. When a query text is introduced, a profile is created for it, and its profile is compared to all the authors' profiles to find the most similar one.

Layton et al. (2012a) used three similar techniques, namely Common n-grams (CNG) (Kešelj et al., 2003), Source Code Author Profiles (SCAP) (Frantzeskou et al., 2007), and Re-centered Local Profiles (RLP) (Layton et al., 2012b) to collect character-level n-grams and create the authors' profiles. In addition, they applied the Inverse Author Frequency (IAF) to weight the n-grams, which, as they explained in their paper, is merely a trajectory of the Inverse Document Frequency (IDF) weighting approach on profiles, as opposed to documents. To compare the distance between different profiles, they used the relative distance (Kešelj et al., 2003) for the CNG-generated profiles, the

size of the intersection between two profiles for the SCAP-generated profiles, and a modified version of the cosine similarity presented in (Layton et al., 2012b).

SMS

Ragel et al. (2013) and Ishihara (2011) addressed the authorship attribution problem in SMS messages. The number of authors in Ragel et al. (2013) was 20 authors, while Ishihara (2011) reached 228 authors. In their paper, Ishihara (2011) reported using 38,193 messages for all the authors. This makes the average number of messages per author around 167 messages. On the other hand, Ragel et al. (2013) used 500 messages for each author; hence, the total number of messages in their dataset was 10,000 SMS messages.

Both Ishihara (2011) and Ragel et al. (2013) used word n-grams as features and combined the SMS messages together, in one document, to increase the text size. This is important because SMS messages, by nature, are limited in size, containing very few words. In terms of classification and validation, Ragel et al. (2013) divided their dataset into training and validation sets, and then they created a profile for each author in both sets. In the following step, they used the Euclidean distance and the cosine similarity to measure the distance between the profiles in the validation and the training sets. They show that grouping 300–400 SMS messages per author increases the uniqueness of an author's profile, leading to a higher accuracy when an author's profile is being classified. However, they do not mention the number of words a profile contained when they combined all these messages, nor the average number of words per message.

In contrast, Ishihara (2011) grouped the SMS messages until a certain number of words was reached. For example, if the current total number of words is 197 words, the maximum is 200 and the next message to be added has four words, they would stop at 197. If it had less than four words, then they would add this message and check the next one. In terms of validation, they created two sets of messages: one set contained messages from the same author, and the other contained messages from

different authors. Then, a word n-grams model was built for each set of messages. To measure the similarity between the different models, they used the Log Likelihood Ratio function. The results show that accuracy reaches 80% when 2,200 or more words per set were used. Since the application of our work is mainly for digital forensics, we believe that it would not be possible to collect enough messages from the candidate authors to get this high number of words.

Twitter Posts

Twitter is the most targeted source for short text in the authorship attribution literature. The number of candidate authors in these studies was on a small scale ranging from 10 to 120 authors in (Bhargava et al., 2013) and (Silva et al., 2011), respectively, and on a large scale up to 1,000 authors in (Schwartz et al., 2013). The most common feature representation approach that was used is the character-level n-grams.

Both Cavalcante et al. (2014) and Schwartz et al. (2013) used character- and word-level n-grams as features while using a SVM classification model. In both papers, the number of candidate authors started at 50, then increased gradually to 500 in (Cavalcante et al., 2014), and to 1,000 authors in (Schwartz et al., 2013). A major difference between these two papers is that Schwartz et al. (2013) proposed the concept of *K-signature*, which they define as "the author's unique writing style features that appear in at least K% of the author's Twitter posts".

Layton et al. (2010) used the SCAP methodology to address the authorship attribution problem for Twitter posts. In their paper, they divided the tweets into training and validation tweets. Then, they combined all the training tweets that belonged to the same author in one document and extracted only character level n-grams from this document as features. To create a profile for that author, they picked the top L most frequent features in his document and ignored the rest. The tweets in the validation set were handled in the same way and a profile was created for each author as well. Fi-

nally, the validation profile was compared to every author's profile and the similarity was simply measured by counting the number of common n-grams between the test profile and the candidate author's profile, i.e., the intersection between the test profile and the candidate author's profile. This similarity measure is known as the Simplified Profile Intersection (SPI).

Bhargava et al. (2013) and Silva et al. (2011) chose a different approach to extract features than the common n-grams method. Bhargava et al. (2013) proposed four categories of features: lexical, syntactic, Twitter-specific, and "other". Examples of lexical features are the total number of words per tweet and the total number of words per sentence. Examples of syntactic features are the number of punctuation marks per sentence and the number of uppercase letters. Examples of Twitter-specific features are the ratio of hashtags to words and whether the tweet is a retweet. Finally, examples of features that belonged to the "other" category are the frequency of emoticons and the number of emoticons per word. As for the classification model, a radial, nonlinear kernel for SVM was used. Similarly, Silva et al. (2011) applied a combination of features that they categorized into four groups: quantitative markers, marks of emotion, punctuation, and abbreviations. Two differences were observed in these two papers: the first is that Silva et al. (2011) used a linear SVM instead of a nonlinear one. The other difference is that Bhargava et al. (2013) experimented with combining the set of Twitter posts into a number of groups in order to enlarge the text body before the feature extraction step. Bhargava et al. (2013) showed that grouping a set of 10 tweets together achieved better accuracy.

3.2.2 Arabic Authorship Attribution

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

Kumar and Chaurasia (2012) performed authorship attribution on Arabic novels where the corpus consisted of a training and a test set for four authors. The average number of words per author in the training set was 67,635 words and the test set was 164,673.25 words. In terms of features, they used the initial and final bi-grams and tri-grams (Rubin, 1978) that, in the case of bi-grams, are formed from the first two letters and the last two letters of every word. Similarly, tri-grams are formed of three letters. The classification process is based on the dissimilarity measure algorithm (Kešelj et al., 2003). Different *profiles* for each user were tested, a profile was the most frequent 200, 500, and 700 bi- or tri-grams. For each profile setting, a dissimilarity threshold value was calculated by comparing the author's profile from the training set with the profile from the test set. For a test document, a new profile was built and the dissimilarity value between the author's profile and the unknown document was compared to the author's dissimilarity threshold value. Their results suggest a 100% accuracy when the initial tri-gram is used with any of the profile sizes.

Ouamour and Sayoud (2013) built their dataset from ancient Arabic books, where the average number of words per book was around 500 words. They collected three books for each one of the ten authors in their dataset. Their features set consisted of (1 to 4)-grams word-level features in addition to "rare words". The datasets were tested using different machine learning classifiers. Each feature was tested alone using Manhattan distance, cosine distance, Stamatatos distance, Canberra distance, Multi-Layer Perceptron (MLP), SVM, and Linear regression. The best results were reported when rare words and 1-word gram features were paired with an SVM classifier. As the value of n (in n-grams) increased, the reported results showed significant deterioration. They correlated this deterioration in performance to the small-sized text they used.

Shaker and Corne (2010) created a dataset comprised of 14 books written by six different authors, with the average number of words per book being 23,942 words. Their approach utilized the most frequent function words to discriminate between two text-

books. Motivated by the work in (Mosteller and Wallace, 1963), they generated a list of 105 function words and ignored the 40 words that are comparatively less frequent than the rest, to end up with a list of 65 **A**rabic **f**unction **w**ords, denoted by *AFW65*. Further filtering was applied to the *AFW65* set where the 11 words with the least frequency variance were removed. The resulting set of 54 **A**rabic **f**unction **w**ords was denoted by *AFW54*. This created a new set of words that consisted of 54 function words. Each book was divided into chunks of words and two sizes were experimented on, 1,000 and 2,000 words. For each chunk, a feature vector was created using the ratio of each function word, and the author name was assigned to the chunk as a class. This created four experimental settings. They used a hybrid approach of Evolutionary Algorithm and Linear Discriminant Analysis that they developed for English in (Shaker et al., 2007) to classify a set of test documents. The best reported result was achieved when a chunk of 2,000 words was used along with the *AFW54* set of function words.

Alwajeeh et al. (2014) built their text corpus from online articles. They manually selected five authors with 100 articles each. Then they manually annotated the articles and extracted the features from them. The average number of words per article was 470.34 words and features, such as the average number of general articles, characters per word, punctuation per article, and unique roots per article, were extracted. They also investigated the effect of using a Khoja stemmer, which returns the original root of the specified word. They used Naïve Bayes and SVM as their methods of classification and the reported accuracy almost reached 100%. In their discussion of the results, they highlighted the negative effect that the Khoja stemmer introduced. They explained that the reason behind this effect is that root stemming causes two different words with different meanings to become the same word, which in return leads to information loss and, therefore, bad performance. These results are inline with the recent findings of Omar and Hamouda (2020) who also demonstrated that using stemming hinders the performance of authorship attribution techniques on Arabic.

Abbasi and Chen (2005b) and Altheneyan and Menai (2014) used exactly the same set of features. In fact, Abbasi and Chen (2005b) proposed these features and then Altheneyan and Menai (2014) adapted their suggestions in their paper. There are 418 features divided as follows: 79 lexical features, 262 syntactic features, 62 structural, and 15 content-specific. Altheneyan and Menai (2014) tested these features on a dataset comprised of ten authors and 30 books, with an average number of words per book ranging between 1,980 to 2,020 words. On the other hand, Abbasi and Chen (2005b) collected 20 web forum messages for each of their 20 authors and the average number of words per message was 580.69 words. In both these studies, variations of Naïve Bayes, SVM, and C4.5, a famous decision tree classifier, were used, and the reported accuracy ranged from 71.93% to 97.43%.

Rabab'ah et al. (2016) collected a dataset of 37,445 tweets for 12 users from the top Arab users on Twitter. On average, they collected around 3120 tweets per author. Three sets of features were used in this study: stylometric features provided by (Al-Ayyoub et al., 2017a), uni-grams, and morphological features extracted using MADAMIRA tool (Pasha et al., 2014), which is a tool made by combining the functionality of MADA (Habash and Rambow, 2005) and AMIRA (Diab et al., 2007) tools for Arabic feature extraction. They used Naïve Bayes, Decision Trees, and SVM for classification and experimented with the sets of features separately and combined. The best result was achieved using SVM with all three sets of features. This study was followed by (Al-Ayyoub et al., 2017b) on the same dataset, where Al-Ayyoub et al. (2017b) investigated the effect of using feature selection tools such as Principle Component Analysis (PCA) and Information Gain on reducing the running time of the classification process.

Al-Ayyoub et al. (2017a) investigated the authorship problem for news articles. They collected around 6,000 articles for 22 authors, where each author has 220 articles on average, and the articles' lengths ranged between 202 and 565 words. For their work, they considered two sets of features. In the first set, they compiled a list of stylo-

metric features from (Abbasi and Chen, 2005a,b; Shaker and Corne, 2010; Cheng et al., 2011; Otoom et al., 2014), while in the other set they considered all the uni-grams that have more than 1000 occurrences in the dataset, then applied feature reduction using a correlation-based feature selection technique (Hall, 1998). The value for a feature is the TF-IDF score. Finally, they used Naïve Bayes, Bayes Networks, and SVM to compare the performance using each feature set separately. The outcome of this study is that using stylometric features yielded a higher accuracy compared to using uni-grams with TF-IDF scores.

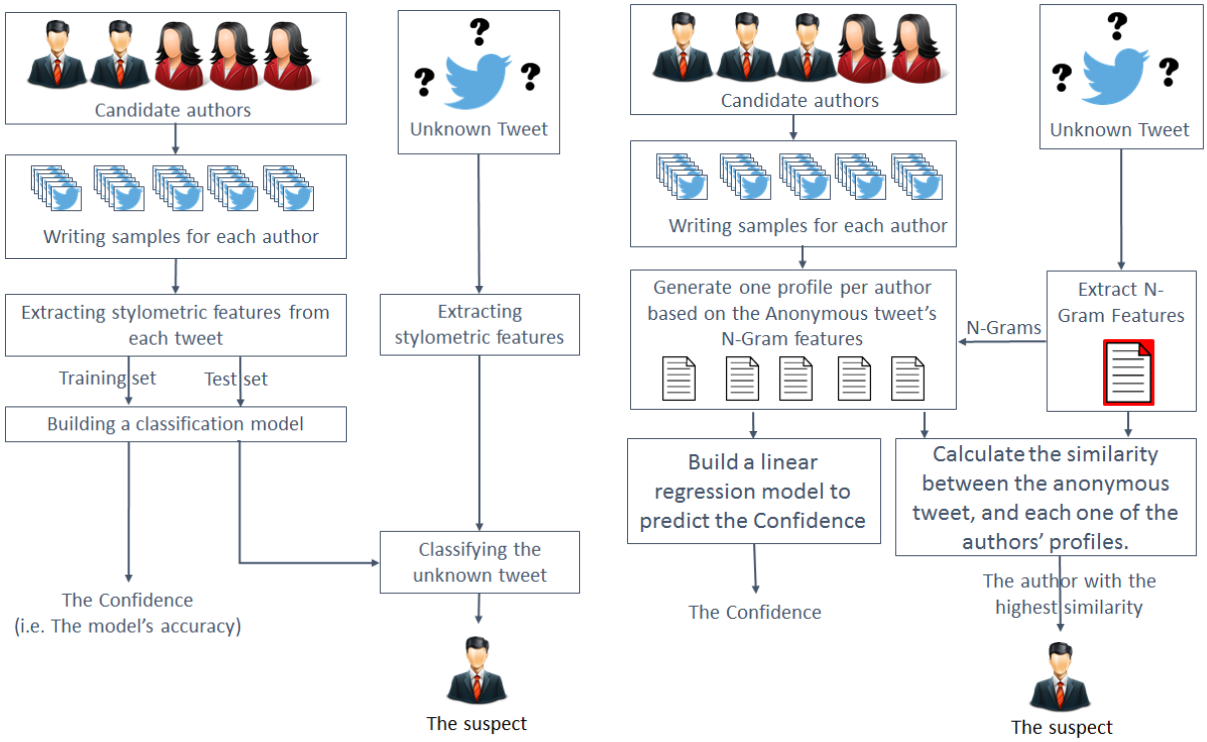
As discussed above, all the relevant works either focus on long Arabic text data or use a very large number of writing samples per author. Given the continuously increasing size of the text generated online, there is a pressing need to benchmark the performance of existing techniques on Arabic short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

3.3 Authorship Attribution

As described in Chapter 2, Stamatatos (2009) described three approaches to extract the writing style from writing samples. The first approach is instance-based, in which a writing style is extracted from every sample separately. By using this approach the candidate author will have s-styles, where s = number of writing samples per author. The second approach is profile-based in which all the writing samples for a particular author are used to generate one writing style for that author. Note that the profile-based approach is different from authorship profiling, where the task is to infer the characteristics of the author such as age, gender, education level, etc. The third approach is a

hybrid one that starts as an instance-based one, then the features are aggregated over all the instances to create one profile per author.

Figure 3.2a shows the steps for the attribution process using the instance-based versus the profile-based approach (Figure 3.2b). In Section 3.3.1, we explain the steps for the instance-based approach and in Section 3.3.2 we explain the steps of the profile-based approach.



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

Figure 3.2 Instance-based vs. profile-based authorship attribution.

3.3.1 Instance-Based Authorship Attribution

Refer to Figure 3.2a. The first step in the attribution process is to collect a set of writing samples for each one of the candidate authors. This is explained in detail in Section 3.4.1. Assuming that the set of candidate authors is identified and a collection

of writing samples for each one of them is collected, the next step is to analyze these writing samples to extract the writing-style features. We discuss the various types of features in Section 3.3.1 below.

Extracting Stylometric Features

In this section, we provide a detailed description of the features that we extracted for each one of the writing samples. As mentioned earlier, Abbasi and Chen (2005b, 2008) highlighted that techniques developed for English cannot be directly applied to Arabic due to the different morphological nature of each language, which directly affects the feature extraction process. In fact, the features list is the only language-dependent element in the authorship attribution process, while the choice of the classification model, such as Naïve Bayes, or SVM, is not. This is because the feature extraction process uses the features list to convert text to the feature vectors, which the classifiers use to build a classification model.

In general, an English-based features list can be used as a starting point for creating a new features list for non-English authorship analysis. First, some basic language-specific features have to be modified in the feature list. For example, the frequencies of alphabets in English have to be replaced with the frequencies of Arabic letters and the ratio of capital letters in English has to be removed because Arabic letters have only one case. On the other hand, the use of elongation "ـ" can be found in Arabic, but not in English. Therefore, it has to be included as a feature for Arabic. Second, the list of features has to be modified when the source of the investigated text changes from news articles, for instance, to tweets. This is because some features are meaningful in one source but not in another. Consider the feature "The greeting line". This feature is only meaningful in e-mail analysis. Looking for a greeting line in a tweet will not yield any results. Finally, there are some features that are common and can work for different languages and in different domains, such as "the ratio of space to characters",

but the number of such features is low. If the number of features is low, i.e., the features are not representative, the accuracy of the classification model will be very low. After that, choosing a classification model is not an issue because the classifier will use the feature vectors the same way whether they were generated for an English text or a non-English text.

We adapted a similar structure of Arabic features presented in (Abbasi and Chen, 2005b), with two main differences: (1) we removed all the features that are inapplicable to Twitter posts (e.g., font size and greetings), and (2) we used a larger set of function words. The following steps show how the features were extracted. We have three categories of features: lexical, structural, and syntactic.

Lexical Features We started by counting the number of characters that included diacritics or "Arabic Tashkil", special characters, and punctuation and excluded white space characters such as spaces, tabs, and newline. Let M be the number of characters in a tweet whether this character occupies a location or not. Examples are alphabets and diacritics. Next, we calculated the ratios of digits (0–9) to M, spaces to M, tabs to M, and spaces to all characters. Lastly, we generated the ratio of every single alphabet to M. Despite the fact that the collected tweets are in Arabic, we also calculated the ratio of the English alphabet. This was important since we observed some tweets that included both English and Arabic words. Finally, it is important to mention that we considered the Alif " ا " letter and the Alif with Hamza " أ " letter to be two different letters, as opposed to Altheneyan and Menai (2014), who combined them under the Alif " ا " letter.

The previous set of features was observed on the characters' level and that is why they are called character-based features. This next set of features, however, was observed on the word level and is therefore called word-based features. The first word-feature is intuitive, which is the word count W. Before the words were counted, we

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

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

- Word-based features:
  1. Word count (W).
  2. Average word length.
  3. Ratio of short words [1–3] character to W.

Structural Features The average sentence length, in terms of characters, was calculated. To do so, newline "\n", period ".", question mark "?" and exclamation mark "!"

characters were used to divide a tweet into a set of sentences. This feature is also used to find the average sentence length the same way the average word's length was calculated. The last structural feature obtained is the ratio of blank lines to all lines. This can be calculated by looking for two newline characters together, i.e., \n\n. A summary of the previous features is provided below:

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

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

- Diacritics. Arabic scripts have many diacritics and Tashkil "تشكيل", where the latter includes the Harakat "حركات" (vowel marks). Ideally, Tashkil in Modern Standard Arabic is used to represent missing vowels and consonant length and it helps identify the words' grammatical tags in a sentence. For example, the question من ضرب الرجلُ؟ means: whom did the man hit? ("من = who/whom", "الرجل = the man" and "ضرب = hit"). However, if the Damma (ُ) on the word الرجلُ is replaced with a Fatha (َ), the question will become من ضرب الرجلَ؟, meaning: who hit the man? So, the change on Tashkil on the word الرجل from Damma to Fatha changed it from being the subject to the object. If Tashkil was not provided, then

the context can help to understand the statement. However, if no context was provided, the reader will not be able to tell which question is being asked. The following diacritics have been observed:

  - Hamza: "أ، ؤ" and stand-alone "ء" were converted to "أ" and "ئ" and "إ" were converted to "إ".
  - Tanwin symbols: "باً، بٌ، بٍ". Tanwin always accompanies a letter. It never appears alone.
  - Shadda: "بّ" the doubling of consonants. It also does not appear alone.
  - Madda: "آ" the fusion of two Hamzas into one. "ءأ".
  - Harakat: includes Fatha "بَ", Kasra "بِ", Damma "بُ" and Sukoon "بْ". They always accompany a letter.

- Punctuation. Arabic punctuation is similar to English; the difference is mainly in the direction that the punctuation faces. For example, while the English question mark "?" faces the left (faces the question), the Arabic question mark faces the other way "؟". Below is the set of punctuation marks that were observed:

  - Arabic Comma ،
  - Arabic colon :
  - Arabic Semi-colon ؛
  - Arabic Question mark ؟
  - Arabic Exclamation mark !
  - Arabic Single quote ‘
  - Arabic End single quote ’
  - Arabic Qasheeda ـ only used for "decoration"

  In addition to Arabic punctuation, English punctuation marks , . ; " ’ and ? were also added to the punctuation set of features.

- Function words. Function words are a set of words that can be grouped together

due to a common property, for example, names of months or pronouns. Below we list examples of these function words, keeping in mind that each word is considered a stand-alone feature:

- Interrogative nouns "أسماء الاستفهام", e.g., "ماذا، متى، هل، كيف، كم".
- Demonstrative nouns "أسماء الإشارة", e.g., "هذا، هذه، ذلك".
- Conditional nouns "أسماء الشرط", e.g., "إن، حيثما، كيفما، أيّ".
- Exceptional nouns "أدوات الاستثناء", e.g., "إلا، غير، سوى".
- Relative pronouns "الاسم الموصول", e.g., "الذي، التي، اللذان، اللتان، اللاتي".
- Conjunction pronouns "حروف العطف", e.g., "ثم، أو، بل".
- Prepositions "أحرف الجر", e.g., "من، إلى، في، عن، على".
- Indefinite pronouns "الضمائر", e.g., "أنا، أنت، هو، هما، هم، هي، هما".
- Eljazm pronouns "حروف الجزم", e.g., "إنْ، لم، لا".
- Incomplete verbs "الأفعال الناقصة". This family of verbs contains a sub-family of verbs: Kada Wa Akhwatoha "كاد وأخواتها", Inna Wa Akhawatoha "إنّ وأخواتها", Kana Wa Akhawatoha "كان وأخواتها".
- Common and famous Arabic names, such as:
- Names of Arabic world countries, e.g., "الاردن، السعودية، الامارات" and their capitals' names, e.g., "عمّان، الرياض، أبوظبي".
- Some popular Arabic names[4], e.g., "محمد، عمر، عثمان، عبيدة" (ArabiNames.com, 2015).
- Names of Hijri and Gregorian Arabic months, e.g., "يونيو، محرم، شعبان، جولية".
- Numeral names and ordinal numbers, e.g., "عاشر، عشرة، الأول".
- Currency names, e.g., "دينار، درهم، دولار".

In total, there are 541 different function words, which means 541 additional distinct features. To summarize how syntactic Arabic-specific features were collected using the

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

feature extraction tool, the following list is provided:

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

Preprocessing for the Instance-Based Approach

Instance-based authorship attribution is considered a classification problem, in which a model is used to classify an anonymous text after training this model on the writing samples of the candidate authors. To do that, various data mining tools can be used, such as WEKA (Hall et al., 2009) or RapidMiner (Hofmann and Klinkenberg, 2013). We used the Java implementation of WEKA to train and validate the classification models.

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

the number of useless features in a specific experiment varies from one experiment to another.

The reason why these useless features appear in the first place is the large number of features that we use. For example, we collect 541 function word features. It is very unlikely that a small set of tweets will contain all these function words. (See Table A.1 in Section 3.4.1 for a list of the top 100 highest frequency function words). However, this should not affect the accuracy of the model since such features will not be used to build the classification model. Even if such a feature appears later in the validation instance (i.e., the anonymous tweet), the already built model will not be able to use it. On the other hand, removing these useless features should reduce the time for training and validation for a classification model.

Figure 3.3 shows the maximum, average, and minimum ratio of usage for each feature grouped by the category. A ratio is calculated by counting the number of times a feature was used in an experiment, divided by 200 experiments[5]. For example, a ratio of 90% means a feature was used in 180 experiments and removed by the RemoveUseless filter in 20 other experiments.

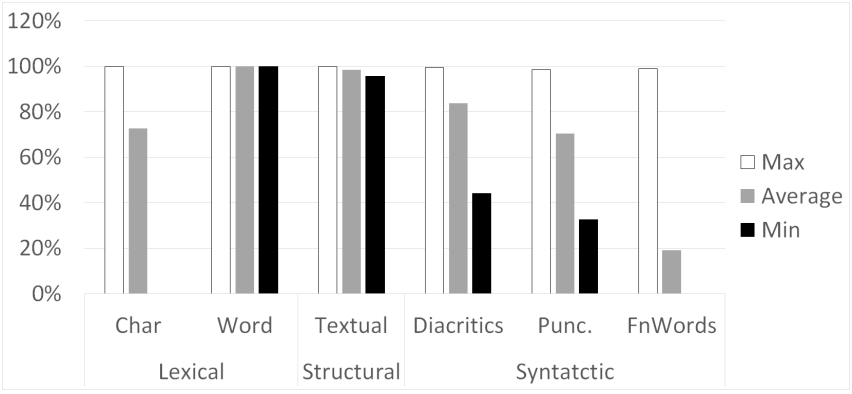


Figure 3.3 The maximum, average, and minimum ratio of usage for each feature grouped by the category.

[5]All the possible settings as per the experimental setup in Section 3.4.2

As Figure 3.3 shows, only function-word features are used less than 60% on average. Word-level lexical features and textual features were used in all experiments with an average ratio of usage being 100% and 98.5%, respectively. The minus sign '-' was the only character-level feature that was not used in any experiment and had a ratio of 0%. For the function-word features, around 26% of them were not used in any experiment. We listed these features in Appendix A.2.

The Classification Process

Figure 3.2a illustrates the process of attributing an anonymous tweet to one of the candidate authors using the instance-based approach. As the figure shows, the next step after extracting the features is to build the classification model. To do that, we used the 10-fold cross-validation technique and divided the writing samples into train and validation sets. The result of this process is a classification model and its accuracy, which is used as confidence. If the model's accuracy is low, this means that the model is not able to differentiate between the authors based on their writing samples. Regardless of how low the accuracy is, the model will always output a candidate author for the anonymous text. In this case, it is up to the authorship attribution domain expert to evaluate to decide whether to accept the results or not.

We used four different classification techniques to evaluate the performance of the instance-based authorship attribution approach in order to compare it with the performance of the profile-based approach. These techniques are Naïve Bayes, Support Vector Machines (SVM), Decision Trees (DT), and Random Forests (RF).

The reason for choosing these classification models is this: one main application of authorship attribution techniques is to use their results and analysis as evidence in courts of law. Because of that, the interpretability of the results is as important as high accuracy. It is crucial that the findings are not only accurate but also intuitive and convincing. For example, SVM and Random Forests are known for high accuracy.

However, the resulting models are complex and can only be seen as a black box, as opposed to Naïve Bayes and decision trees, whose results are easier to represent. An authorship attribution expert needs to explain a conclusion rather than merely present it and using such complex models will not enable an expert to do so. I discussed these techniques in Section 2.4.

3.3.2 Profile-Based Authorship Attribution

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

Extracting the N-Grams From the Anonymous Tweets and the Sample Tweets

This section describes the process of extracting the n-gram features from the authors' writing samples. A *gram* is a unit of text (i.e., token) based on the modality level and n is the integer number of consecutive tokens that are considered as one feature. The n-gram approach can be applied on three modality levels: characters, words, and parts-of-speech (POS). For example, 2-grams on the character level means that we tokenize a text into a series of characters, then we take every two consecutive characters as one feature. Table 3.1 shows a sample sentence and the corresponding n-gram features for each modality level.

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

Table 3.1 Sample n-grams extracted from the text: "it is noticed and appreciated" and the corresponding part-of-speech tag sequence is "PRP VBZ VBN CC VBN".

We used Stanford's (Manning et al., 2014a) NLP library[6] for text segmentation, tokenization, and POS tagging. Note that the term n-gram is used in the literature to indicate that the number of tokens in a feature is exactly n. For example, the term 3-grams, or tri-grams, means each feature has exactly three tokens. In this work, however, we extract all the grams of lengths 1 to n as shown in Table 3.1 and for brevity, we use the term n-gram instead of $1-n$-grams.

Generating the Authors' Profiles

After extracting the features, we use the TF-IDF technique to score each feature and create the authors' profiles. This scoring technique, which is famous for its simplic-

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

ity, gives a high score for words that appear many times in one piece of text (Term Frequency) while penalizing terms that appear many times in other documents. To understand the motivation behind this technique, consider the word "the". Due to its usage, the word "the" is likely to appear many times in a document, compared to other terms in that same document. On the other hand, it will also appear frequently in other authors' documents. Therefore, it will get a low score for being very common, i.e., not unique for a certain author. In contrast, consider the word "kindly" that might be common in one author's text while being replaced by the word "please" by another author. If this was the case, then these two terms will have high scores, given that they appeared in one author's text more frequently than for other authors.

Eq. 3.1 and Eq. 3.2 show the formulas to calculate the Term Frequency (TF) and the Inverse Document Frequency (IDF), respectively, where W_i is all the writing samples, i.e., tweets, for candidate author c_i, $|C|$ is the size of the set of candidate authors, i.e., the number of candidate authors and b is a constant that equals 0.1.

$$TF\left(gram, W_i\right) = \frac{frequency\left(gram, W_i\right)}{maxGramFrequency\left(W_i\right)}, \tag{3.1}$$

$$IDF\left(gram\right) = log\left(\frac{|C|}{b + |AuthorsEverUsed\left(gram\right)|}\right) \tag{3.2}$$

To illustrate how these equations are used to generate the profiles of the anonymous tweet and the authors, we provide the following example: let a be an anonymous tweet and a set of two candidate authors C, where W_1 and W_2 contain all the tweets written by authors c_1 and c_2, respectively. Let the n-gram features for a, W_1, and W_2 be extracted as per Section 3.3.2. For the sake of this example, assume that we are performing the attribution process on the word level only and that the feature-vector

based on the anonymous tweet a is [$gram_1$, $gram_2$, $gram_3$].

To generate a profile for the anonymous tweet, we only use Eq. 3.1 and we do not use the writing samples of the candidate authors. Assume that the frequencies for $gram_1$, $gram_2$, and $gram_3$ in the anonymous tweet are 1, 2, and 5, then the profile for this anonymous tweet on the word-level will be [0.2, 0.4, 1].

To generate a profile for author c_1, we compute the frequencies for the same grams: $gram_1$, $gram_2$, and $gram_3$ in the author's tweets. For example, if we examine all the author's tweets we find that $gram_2$ appears 5 times, then $TF(gram_2, W_1) = 5$. Assume that $TF(gram_1, W_1) = 3$ and $TF(gram_3, W_1) = 0$, i.e., $gram_3$ does not appear in any of author c_1's tweets. So far, using only Eq. 3.1, the feature-vector for c_1 is [0.6, 1, 0]. Similarly for author c_2, we calculate the frequencies for $gram_1$, $gram_2$, and $gram_3$ using Eq. 3.1. Assume that the resulting frequencies for $gram_1$, $gram_2$, and $gram_3$ are 4, 0, and 0. Therefore, the feature-vector for c_2 is [1, 0, 0].

Next, we need to calculate the IDF value for each gram, given by Eq. 3.2. Notice that the number of authors is fixed and b is a constant, so we only need to calculate $|AuthorsEverUsed(gram)|$, i.e., find the number of authors who used that gram in any of their tweets. $Gram_1$ was used by both authors, so $IDF(gram_1) = log(2/(0.1+2)) \approx$ -0.02. $Gram_2$ was used by only one author, so $IDF(gram_2) = log(2/(0.1+1)) \approx 0.26$. Finally, $gram_3$ was not used by any author, so $IDF(gram_3) = log(2/(0.1+0)) \approx 1.3$. Notice that had we not used the constant b, a division by zero would have occurred. Therefore, we added the constant b and set its magnitude to be smaller than 1. Finally, we penalize each TF value with the corresponding IDF value. The resulting feature-vectors for author c_1 and c_2 on the word-level are [-0.012, 0.26, 0] and [-0.02, 0, 0], respectively.

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

Computing the Similarity Scores per Modality Level

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

To measure the similarity between the anonymous tweet and an author's profile on a certain modality level, we need to calculate the distance between their vector profiles. One simple technique to calculate this distance is the Cosine Similarity that is shown in Eq. 3.3. From a geometric perspective, the smaller the angle between two vectors, the more similar they are. By simplifying the formula of the cosine similarity, the distance ends up being the dot product of the two vectors.

$$\begin{aligned} similarity\left(\overrightarrow{S_i}, \overrightarrow{S_\alpha}\right) &= proj_{\overrightarrow{S_\alpha}}\overrightarrow{S_i} \times \parallel \overrightarrow{S_\alpha} \parallel \\ &= \parallel \overrightarrow{S_i} \parallel \times \cos(\theta_i) \times \parallel \overrightarrow{S_\alpha} \parallel \\ &= \parallel \overrightarrow{S_i} \parallel \times \frac{\overrightarrow{S_i}.\overrightarrow{S_w}}{\parallel \overrightarrow{S_i} \parallel \times \parallel \overrightarrow{S_\alpha} \parallel} \times \parallel \overrightarrow{S_\alpha} \parallel \\ &= \overrightarrow{S_i}.\overrightarrow{S_w} \end{aligned} \tag{3.3}$$

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

Using this similarity measure, we compute three similarity scores for each author,

one on each modality level.

Calculating the Confidence per Modality Level

In the previous step, we calculated the similarity scores for each author. In this step, we calculate a confidence value that describes the model's ability to discriminate between the authors' profiles. To understand the motivation behind this step, consider these two cases with the following similarity scores on the same modality level for three candidate authors. Case1: $Sim(C_1, a)$ = 5, $Sim(C_2, a)$ = 2, $Sim(C_3, a)$ = 1 and Case2: $Sim(C_1, a)$ = 5, $Sim(C_2, a)$ = 4.7, $Sim(C_3, a)$ = 4.2. In both cases C_1 has the highest similarity score; however, in Case2 the similarity scores are too close to each other, which means that the authors' writing styles are very similar. We quantify the model's ability to discriminate the authors' profiles by measuring the model's ability to correctly predict the author of each one of the writing samples if they were used as anonymous tweets. To do that, we divide the authors' writing samples into 10 folds: 9 folds to be used as writing samples and one fold to be used as anonymous tweets. Note that this is done for each modality level separately.

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

author with the highest similarity score is the most plausible author.

Before we move on to the next validation tweet, we extract the following six features from the current tweet: (1) the highest similarity score, (2) the lowest similarity score, (3) the average similarity score, (4) the difference between the highest and the second to the highest similarity scores (i.e., the runner-up), (5) the length of the validation tweet (in tokens, on the same modality level), (6) the number of tokens that appear in both the anonymous (i.e., validation) tweet and the tweets of the author with the highest similarity score. We repeat the same steps for the rest of the tweets in the validation set.

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

We perform this process 10 times, each time considering a new fold for validation. At the end of this 10-fold process, we will have a feature vector of length 6 and a confidence value for each tweet in the set of writing samples of every author, for a specific modality level. To calculate the model's confidence score for the original anonymous tweet, we use a linear regression model. This model is trained on the writing samples and the same 6 features are extracted from the anonymous tweet based on the similarity scores of the authors that were measured in Section 3.3.2. The model is then used to predict a confidence score for the anonymous tweet on a specific modality level.

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

Combining the Similarity Scores and Confidence Values to Predict the Actual Author and Compute the Overall Confidence

The final step in predicting the candidate authors is to use the similarity scores that were calculated in Section 3.3.2 and the confidence values that were calculated in Section 3.3.2 to generate a cumulative similarity score per candidate author and an overall confidence value for the model. To compute the cumulative score, we normalize each author's similarity score in each modality by multiplying it by the corresponding confidence value, then we sum all three scores together. The most plausible author is the one with the maximum combined, normalized score. For example, assuming that we have 2 authors: c_1 and c_2 and that the similarity scores for c_1 on the lexical, character, and POS level are [3, 1, 5], respectively, while the similarity scores for c_2 on the lexical, character, and POS level are [2, 4, 2], respectively and the model's confidence scores on the lexical, character, and POS level are [0.6, 0.84, 0.5], respectively, then the cumulative score for c_1 equals (1.8 + 0.84 + 2.5) = 5.14, and the cumulative score for c_2 equals (1.2 + 3.36 + 1) = 5.56. Based on the cumulative similarity scores, c_2 is the most plausible author.

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

To choose the overall confidence of the model, we take the maximum confidence among the set of confidence scores for the modalities whose prediction matches the predicted author. For example, consider the same example from above where the lexical, character, and POS normalized scores for c_1 are [1.8, 0.84, 2.5] and for c_2 are [1.2, 3.36, 1]. If we consider each modality separately, then the most plausible authors on the lexical, character and POS levels are c_1, c_2, and c_1, respectively. Because we only look at the modalities whose prediction matches the predicted author based on the cu-

mulative score, we only consider the confidence of the character level modality. In this case, the overall confidence is $max(0.84)$ = 0.84 or 84%.

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

3.4 Experimental Design

3.4.1 Dataset

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

Data Collection

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

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

on Twitter. However, we aimed for a more challenging and realistic scenario where the number of tweets in a user's account is small.

We started by retrieving a set of random tweets that are written in Arabic. These tweets were a mixture of Arabic and non-Arabic tweets such as *'Farsi'* or Urdu because these languages use very similar character sets. We extracted only the 'user' information from each tweet and ignored the tweets' meta-data. Note that one line of work on authorship attribution, known as multi-modal authorship attribution (Boutwell, 2011; Saevanee et al., 2015), uses different types of meta-data with the text itself to identify the author of an anonymous text. For example, Boutwell (2011), uses the cellphone tower information of a suspected cellphone to identify the author of an investigated SMS.

To create a dataset with many users, we repeated the previous step multiple times, each time keeping only the user information until we collected a list of around 160 different usernames. Next, for each username in the created list, we retrieved a set of tweets from the user's timeline and filtered out tweets that contained only a hyperlink or an emoji. We also replaced all the usernames and hashtags in the tweets' bodies with the '@' symbol and the '#', respectively. This was necessary because (1) these elements are not part of the writer's style but the style of the original creator of these usernames or hashtags (Layton et al., 2010) and (2) usernames can reveal an author's social network, which can give a strong indication of who the real author is. Our goal was to reach 2000 tweets per author, but most of the retrieved authors had much fewer than that. We manually inspected every author's tweets looking for non-Arabic ones. An author whose tweets were not in Arabic was removed from the dataset. We stored the authors' names and their tweets, along with the tweets' features. This was important for reducing the running time of the experiments because it is likely for a tweet to be used in multiple experiments.

After preprocessing the retrieved tweets, 155 Twitter users remained on the list and

115,786 tweets were collected with an average of 747 tweets per user. Table 3.2 shows some descriptive statistics of the dataset and Table A.1 in Appendix A.3 shows the top 100 most frequent function words.

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

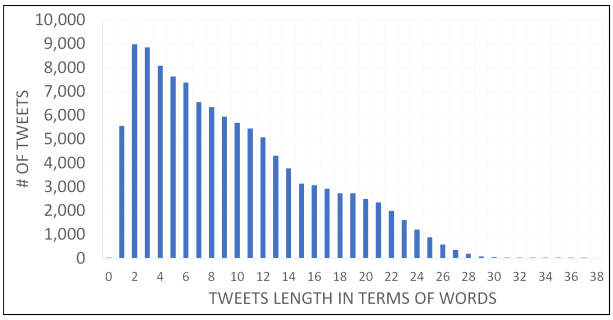
(a) The whole dataset.

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

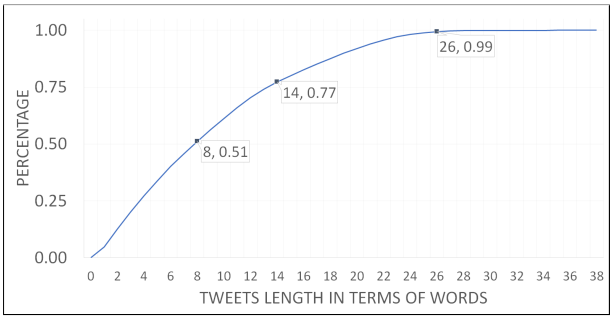
(b) Tweets.

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

(c) Diacritics.



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



(e) Empirical cumulative distribution function.

Table 3.2 Descriptive statistics for the dataset.

Modern Standard Arabic vs. Colloquial Arabic

Upon manual inspection of the nature of the collected tweets, we noticed that the tweets are written in a mixture of Modern Standard Arabic (MSA) and colloquial Ara-

bic, with the majority of the tweets being in Modern Standard Arabic. It is important to note, however, that we did not further investigate whether the collected tweets belong to a specific region or to different ones. This is important due to the variations in Arabic language that could be observed in different Arab regions. Such variations have been investigated in (Alamr, 2022). On the other hand, we have not included in this work tweets that are written in non-Arabic characters.

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

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

The effect on the profile-based approach As described earlier, the profile-based approach does not look for certain features. Instead, the features are generated from the writing samples. In our case, the n-gram features are extracted from the anonymous tweet, and then the same n-grams are extracted from the writing samples for each candidate author. This makes the profile-based approach indifferent to the nature of the language used. In fact, it has an advantage over the instance-based approach since an

instance-based approach is likely to miss typos that appear in the text unless they are hard-coded as features. In case all the words in the text are collected as features (i.e., collecting word 1-grams) then such typos will be caught if no selection of the top k features was applied or if the typo occurs very frequently in the text.

3.4.2 Experimental Setup

We ran all our experiments[7] on small datasets containing between 2 and 20 authors, each with 25 writing samples. Our decision of conducting this study on a small number of candidate authors is justified by the real-world application of authorship attribution. In most cases, a law enforcement officer or the plaintiff of a civil case has a small number of candidate authors for a piece of anonymous text (Luyckx and Daelemans, 2008). Although some of these candidate authors may be very prolific on Twitter and have many writing samples, we aimed for a more challenging scenario where the number of writing samples per author is small. Our experimental setting simulates real-life scenarios of conducting authorship attribution. This setting ensures that the reported results are realistic and that the accuracy is not overestimated.

This decision was also justified by the work of Luyckx and Daelemans (2011) and Ding et al. (2015). Luyckx and Daelemans (2011) have worked on students' essays authorship analysis and highlighted that the accuracy of the authorship attribution process begins to drop significantly as the number of authors increases beyond two. Ding et al. (2015) have conducted a set of experiments on emails authorship analysis to compare the performance of their proposed approach to the performance of SVM and DT classifiers. They conducted their experiments for 2, 5, 10, and 20 authors and the results of these experiments agree with the findings of Luyckx and Daelemans (2011). Given that an average tweet is much shorter than an email, we limited the number of

[7] All experiments were conducted on a workstation running Windows 7 (64-bit) on an Intel® Core(™) i7-4700HQ CPU @ 2.40 GHz (8 CPUs) with 16 GB RAM.

authors in our experiments to 20 authors.

Similar to (Ding et al., 2015), we conducted the experiments on groups of 2, 5, 10, and 20 candidate authors. For each group, we sampled 5 mini-datasets (a, b, c, d, and e) containing the same number of authors. Sampling for each mini-dataset was done without replacement, while sampling across datasets was done with replacement. For example, for a group of 10 authors in a certain experimental setting, we sampled five mini-datasets (a, b, c, d, and e) where each mini-dataset contains 10 authors. Author x may appear in any dataset only once (without replacement) but may appear in one or more datasets (with replacement). Each experimental setting was repeated 10 times. Each time, the 10-fold cross-validation approach was used to divide the dataset into training and validation sets. The goal of such configuration is that we test with various writing styles for the same number of authors and, hence, reduce the variance in the reported results. Table A.1 in Appendix A.3 shows that the difference in performance for the profile-based approach on the mini-datasets at the $\alpha = 0.05$ is significant for all the groups of candidate authors ($p < 0.01$).

The results are reported as the average of 50 runs (10-folds are considered 1 run), which is calculated as the average of the 10 runs for each one of the five mini-dataset (a, b, c, d, and e). The outcomes of an experimental setting are four accuracy values for four datasets containing 2, 5, 10, and 20 candidate authors. Each value resembles the percentage of correctly classified tweets using a certain classifier, i.e., the predicted author for that tweet is the actual author.

We used WEKA's implementation of the classification algorithms in Section 3.3.1. Table A.1 below shows the implementation of each algorithm and the parameters that were changed from WEKA's default settings. Among the four implementations, LibSVM is the only model that is not available directly in WEKA's package and had to be included manually.

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

* Available under WEKA.Classifiers

Table 3.3 Implementations of the classification algorithms and their parameters.

3.5 Results and Discussion

As described in Section 3.1.2, the aim of these experiments is to answer four research questions (RQ) focusing on the performance of the n-gram approach. However, due to the lack of research on Arabic authorship analysis and for the sake of completeness, we extended our analysis to include the behavior of the instance-based classification techniques.

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

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

Increasing the Number of Candidate Authors

We consider this to be the baseline scenario with 25 tweets per candidate author and without specifying a condition on the minimum number of words per tweet. We ran this experiment on datasets containing 2, 5, 10, and 20 candidate authors as per the

experimental setup described in Section 3.4.2. The results of this experiment are shown in Figure 3.4.

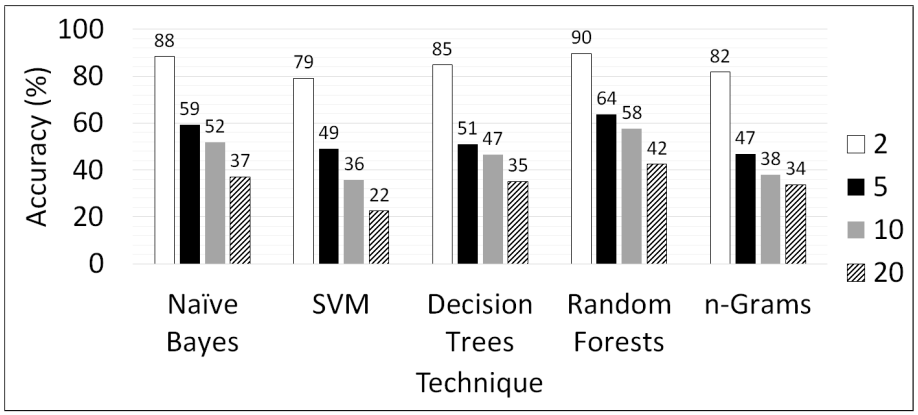


Figure 3.4 Baseline scenario: instance-based (NB, SVM, DT, and RF) vs. profile-based (n-gram) for 2, 5, 10 and 20 authors.

Figure 3.4 shows that the accuracy dropped for all the attribution techniques as the number of candidate authors increased. An Analysis of Variance (ANOVA) test was conducted at the $\alpha = 0.05$ level to determine the significance of this change, where the two factors are the classification technique and the number of candidate authors in a dataset. The results showed that both independent variables, increasing the number of authors and using different attribution techniques, had significant effects on the accuracy of the attribution process (DV).

To compare the different numbers of candidate authors we conducted an ANOVA: single factor[8] and the result showed that there is a significant effect of increasing the number of authors (IV) on the accuracy of the attribution process (DV) at the $\alpha = 0.05$ level. We conducted post hoc comparisons using the Tukey HSD[9] test at the $\alpha = 0.05$ level to compare the four conditions. The results showed that there was a significant difference in the accuracy when the number of candidate authors increased from 2 authors to 5, 10, and 20 authors, as well as from 5 to 20 authors. However, when

[8] Conducted online, using http://astatsa.com/OneWay_Anova_with_TukeyHSD/
[9] See footnote 8

the number of authors increased from 5 to 10 and from 10 to 20 the difference was insignificant. The Mean, the Standard Deviation (SD), and the results of the ANOVA test are summarized in Table A.2.

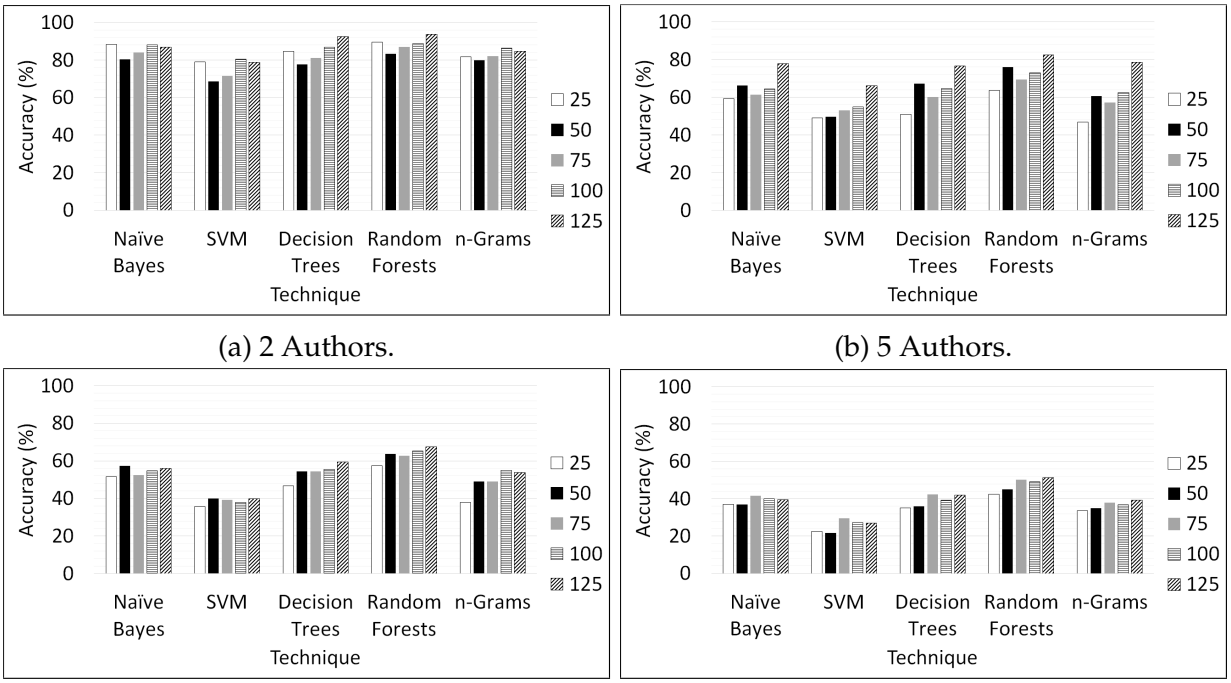
As we are interested in the performance of the n-gram approach, we conducted four paired two-sample t-Tests at the $\alpha = 0.05$ level. In each test, we compared the accuracy of the n-gram approach to one of the instance-based classifiers. The results showed that there is an insignificant difference in the accuracy when the n-gram approach is compared to either SVM or DT. In contrast, the n-gram approach performed significantly worse than NB and RF, respectively. The Mean, SD, and results of the ANOVA test are summarized in Table A.3.

Increasing the Number of Tweets per Author

In this experiment, we increased the number of tweets per author from 25 (the baseline) to 125 tweets, in steps of 25. We wanted to see if the increase in the number of tweets would help the attribution techniques perform better as the number of candidate authors increased. We did not increase the number of tweets per author beyond 125 to keep the scenario realistic, as suggested in (Luyckx and Daelemans, 2008). Figure 3.5 shows the results of this experiment where each set of authors was tested with 25, 50, 75, 100, and 125 tweets per author.

We conducted four one-way ANOVA tests, one for each number of candidate authors, at the $\alpha = 0.05$ level to test whether the increase in the number of tweets per candidate yielded a significant change in the performance of the attribution process (Mean and SD are provided in Table A.4). The outcome of these four ANOVA tests showed that the change in the performance (DV), which was caused by increasing the number of tweets per author (IV), was significant only for 5 candidate authors while being insignificant for 2, 10, and 20 candidate authors.

A post hoc Tukey test was conducted on the set of 5 candidate authors to see which



(a) 2 Authors.

(b) 5 Authors.

(c) 10 Authors.

(d) 20 Authors.

Figure 3.5 Increasing the number of tweets per candidate author (from 25 to 125)

increase in tweets per author had a significant effect. The results of this test showed that among the 10 possible pairwise comparisons only two were significant: increasing the number of tweets from 25 (the baseline) to 125 (the maximum number of tweets) $[p = 0.001]$ and from 75 to 125 $[p = 0.02]$.

Based on these results we noticed that a larger number of additional tweets was needed to have a statistically significant increase in the performance; however, this increase was limited to five authors. As the number of authors increased beyond five, having 125 tweets per author was not significant. As mentioned earlier, increasing the number of tweets to more than 125 tweets per author is unrealistic (Luyckx and Daelemans, 2008). Furthermore, increasing the number of tweets per author will lead to a longer time span that covers these tweets, and, so, they are more likely to cover a larger number of topics. As new topics emerge constantly in daily life, it is very

likely that the authors' styles have adapted to these topics; therefore, introducing more tweets could have a negative effect on the attribution. These findings agree with the work of Bhargava et al. (2013).

As we are interested in the performance of the n-gram based approach, we conducted four ANOVA: single Factor t-tests, one for each set of candidate authors, at the $\alpha = 0.05$ level (The mean and SD are shown in Table A.5). All four t-test results showed that the performance of the various attribution techniques is significantly different.

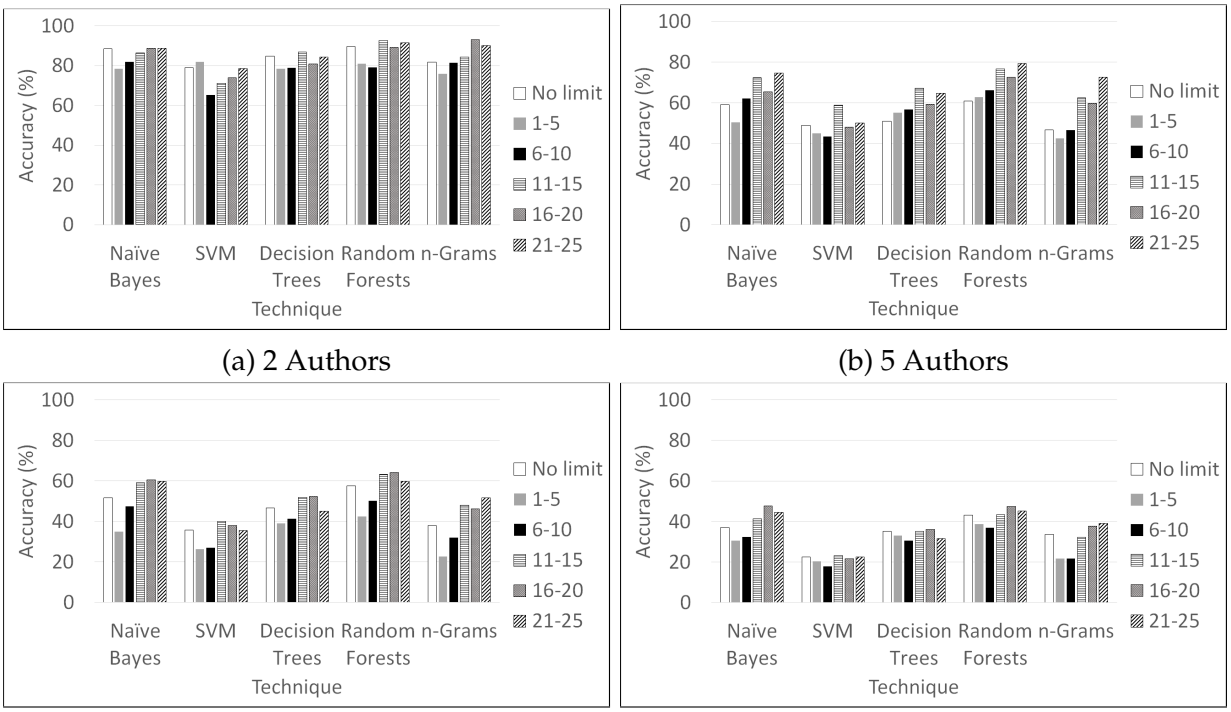
Using four post hoc Tukey tests, we investigated the significance of the difference in performance between the n-gram approach versus other instance-based algorithms. The results of these tests showed that for all four tests there is no significant difference between using the n-gram approach versus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

Specifying the Minimum Number of Words per Tweet

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

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

of this experiment are shown in Figure 3.6.



(a) 2 Authors

(b) 5 Authors

(c) 10 Authors

(d) 20 Authors

Figure 3.6 Specifying the minimum number of words per tweet

Figures 3.6 show that there is an increase in the performance when the size of tweets increases. To test the significance of this increase we run four ANOVA: single Factor t-tests, one for each set of candidate authors, at the $\alpha = 0.05$ level. The results of these four tests showed that the difference is insignificant for 2 and 20 authors while being significant for 5 and 10. Upon further inspection of the significance of the results for 5 authors using a post hoc Tukey test, we noticed that the differences for all the pairwise comparisons were insignificant. This agrees with Simon (2005), who explains that it is possible to have a significant F-score while having insignificant post hoc test p-values. In contrast, the post hoc Tukey test for 10 authors shows that the increase in the performance when the range of tweets' length increased from 1-5 to 6-10 or to 11-15 is significant. The detailed results are presented in Appendix A.3 in Table A.6.

To evaluate the performance of using n-grams compared to other classifiers we looked at the F-scores of four ANOVA: single Factor t-tests at the $\alpha = 0.05$ level. The scores showed that for all four number of authors' settings, the difference among the classification techniques is significant. To identify the significant pairwise comparisons we run four post hoc Tukey tests [detailed results are in Appendix A.3, in Table A.7]. The results of the Tukey tests showed that, except for Random Forests, using n-grams is either the same (i.e., the difference is insignificant) or better than using the other classification techniques. Only Random Forests performed significantly better than n-grams and that was when the number of authors was 5, 10, and 20. When the number of authors was 2, the difference between Random Forests and n-grams was insignificant.

Merging Tweets Into Artificial Tweets

In this last attempt to evaluate the performance of the attribution techniques, we try to address the problem of the small text size by merging groups of five tweets into single artificial tweets. Based on that, the 25 tweets per author that were used in Section 3.5.1 "Increasing the number of tweets per author" experiment are grouped into 5 artificial tweets, where each artificial tweet is created by concatenating the text of the 5 tweets.

The resulting experimental setting is the following: similar to Section 3.5.1, we use sets of 2, 5, 10, and 20 authors with 5, 10, 15, 20, and 25 artificial tweets in each experiment. These artificial tweets are generated from 25, 50, 75, 100, and 125 tweets, respectively, the exact same tweets that were used in Section 3.5.1. The reason why we used the same tweets is to have one variable in the new experiment, which is merging the tweets into groups. Had we used a new set of single tweets, then merged them for this experiment, that would have generated some bias based on the content of the new tweets, even if we controlled the text size over the selected tweets. Therefore, we kept the tweet ID for the tweets used in the aforementioned experiments, then used the IDs to group the tweets into artificial tweets.

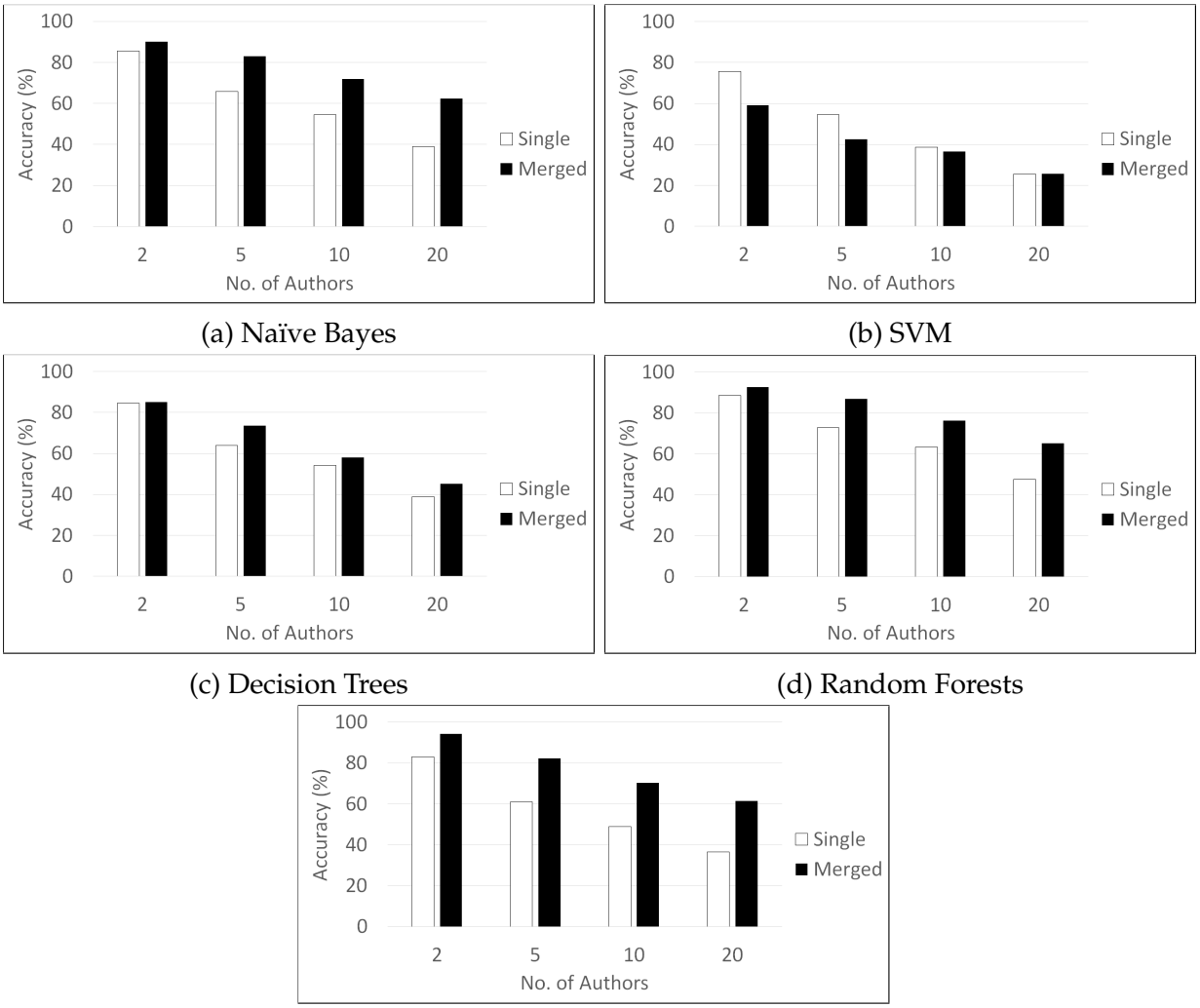
Similar to previous experiments, we start by looking at the performance of the various attribution techniques. We use 2, 5, 10, and 20 authors with 5, 10, 15, 20, and 25 artificial tweets per author. As every 5 tweets are grouped into one artificial tweet, this is equivalent to 25, 50, 75, 100, and 125 tweets per author. The results of these experiments are shown in Figure 3.7. We report the accuracy of a certain classification technique as the average of its performance for the varying number of tweets per author, i.e., the average performance for 5, 10, 15, 20, and 25 tweets per author.

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

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

Upon further analysis of the results using Paired Two Sample t-tests at the $\alpha = 0.05$ level, the difference was significant for Naïve Bayes, Random Forests, and n-grams. In contrast, the difference between SVM and DT was insignificant. The detailed results are presented in Table A.8 in Appendix A.3 .

We compare the performance of n-grams to other classification techniques using ANOVA t-test at the $\alpha = 0.05$ level. The results of this t-test show a significant difference between the 5 techniques. Upon using a post hoc Tukey test, the only significant comparison was between SVM and n-grams. The detailed results are presented in Table A.9 in Appendix A.3. In general, our experimental result is in line with the experiments shown in (Ding et al., 2015) and (Luyckx and Daelemans, 2011). As the number of candidate authors increases, the complexity of the classification also increases,



(a) Naïve Bayes (b) SVM

(c) Decision Trees (d) Random Forests

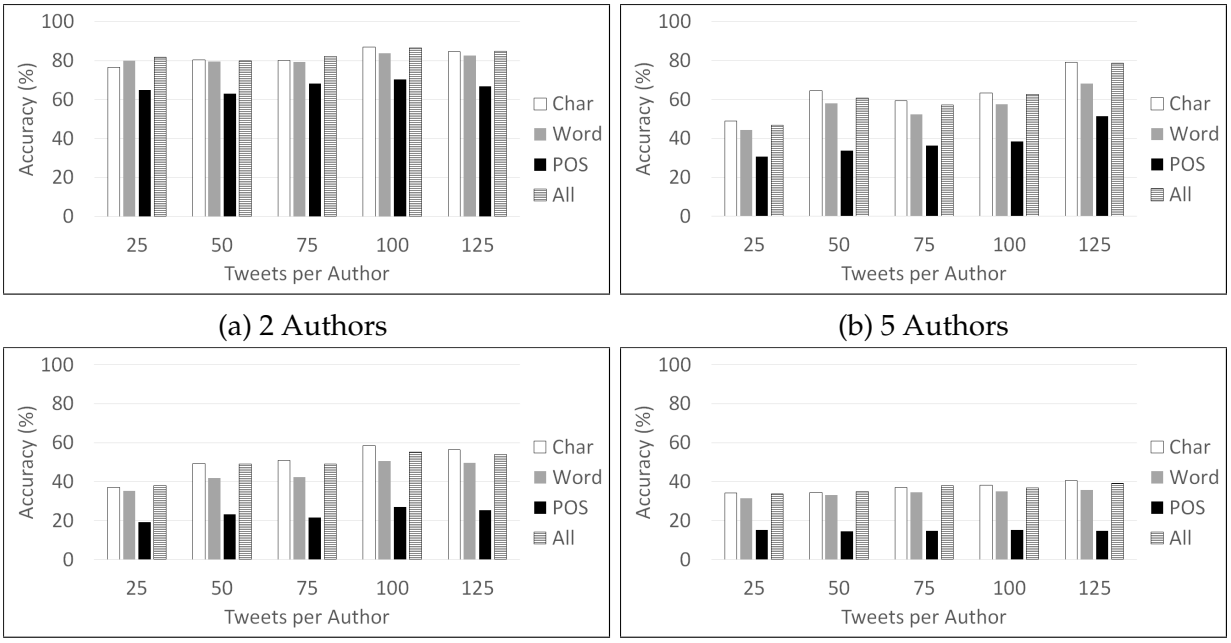
(e) n-gram

Figure 3.7 Merging tweets into groups of five tweets.

which in general leads to decreased accuracy. On the other hand, the performance of the profile-based approach with n-grams is on par with instance-based models. This indicates that there is no trade-off between accuracy and visualization.

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

In this section, we answer the second research question: which n-gram level has the highest effect on the attribution process? As mentioned earlier, n-grams are the N consecutive tokens from a tokenized text, where the tokenization is on the character-, word-, or POS-level. In all the previous experiments we used all three levels of modalities, and these modalities were evaluated using a linear regression model to calculate the confidence, as shown in Section 3.3.2. The modality level with the highest confidence was used in predicting the candidate author. In this set of experiments, however, we evaluate each modality level separately for 2, 5, 10, and 20 authors, with 25, 50, 75, 100, and 125 tweets per author. The results of these experiments are depicted in Figure 3.8



(a) 2 Authors (b) 5 Authors

(c) 10 Authors (d) 20 Authors

Figure 3.8 Evaluating each n-gram modality separately.

Statistical analysis using One-way ANOVA tests at the $\alpha = 0.05$ level shows a sig-

nificant difference among the four modality levels (character, word, POS, or all of the levels together). We further analyzed the results using post hoc Tukey tests, and the results showed that there is an insignificant difference between using either character-level, word-level, or all three levels of modalities combined. In contrast, the difference was significant when only POS n-grams were used. The results of the ANOVA test and the Tukey tests are provided in Appendix A.3, Tables A.10 and A.11, respectively.

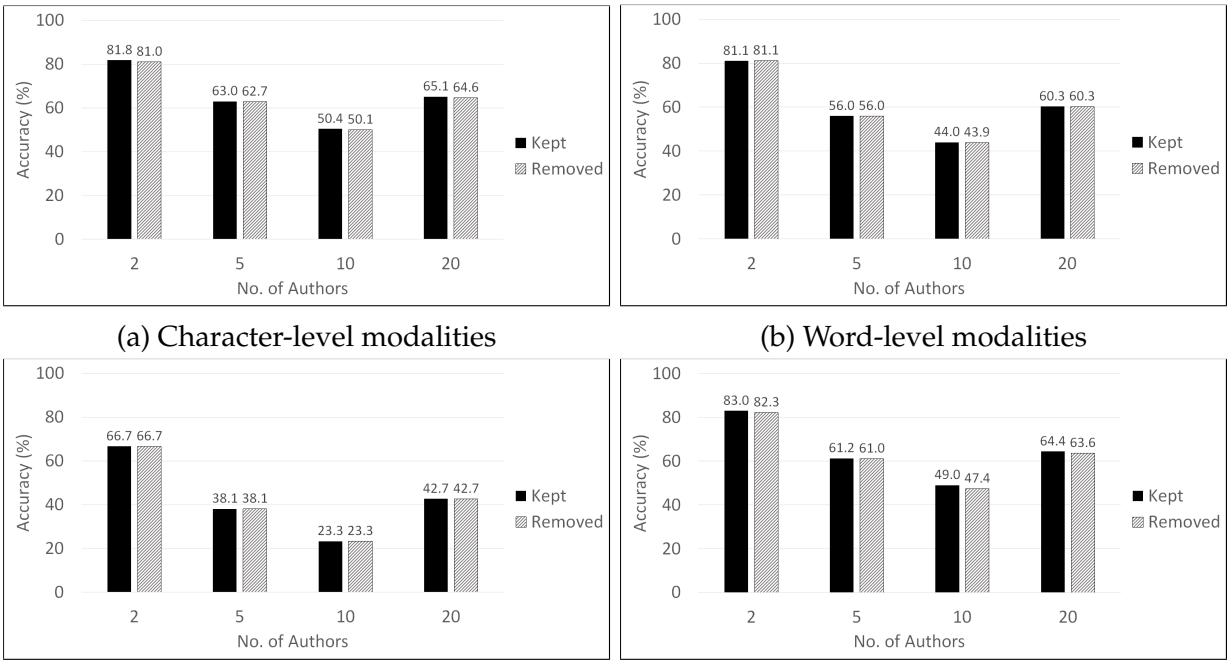
Generally, the experimental result is in line with the results reported in (Ding et al., 2015). They showed that lexical modality performs the best for English. However, for Arabic text, we find that the character modality performs better than the lexical modality with respect to the mean value. Statistically, their difference is insignificant. We suspect that for Arabic, matching n-grams is harder than English, as Arabic tends to merge pronouns with words instead of separating them (Liberman, 2008). For example, English uses "his book" and "her book", which translates to "كتابه" and "كتابها" in Arabic. If we look at 1-grams for these sentences, then we have three grams for English: "his", "her", and "book"; and two grams for Arabic: "كتابه" and "كتابها". As noticed in Arabic, the word "book" was distributed over two grams and could be distributed over more grams, depending on the pronouns attached to it. However, using character modalities, specifically, using 4-grams, the word "book", i.e., "كتاب" will be matched to the same gram in both cases.

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

The use of diacritics is one of the major morphological properties that make the Arabic language different from English and this prevents authorship attribution techniques that are developed for English from being used in Arabic. The use of diacritics in Arabic is optional both in Modern Standard Arabic and Colloquial Arabic. In this set of experiments, we aim to see whether removing diacritics before the attribution

process or keeping them would have an effect on the outcome. As a reminder, 40% of the tweets in our dataset contain diacritics (See Section 3.4).

We use the same experimental setup as the previous experiments: 2, 5, 10, and 20 authors with 25, 50, 75, 100, and 125 tweets per author. We evaluate the diacritics effect on char- and word-level modalities, but not POS. This is because diacritics should be removed before retrieving POS tags. The results of these experiments are shown in Figure 3.9.



(a) Character-level modalities

(b) Word-level modalities

(c) POS-level modalities

(d) All modality levels

Figure 3.9 Evaluating the effect of diacritics on the n-gram approach. The figures showed the average performance for a set of authors on 25, 50, 75, 100, and 125 tweets per author.

As shown in the figure, removing diacritics barely has any effect on the attribution process. We verify that using 4 one-way ANOVA tests at the $\alpha = 0.05$ level. The results of these tests confirm that the difference for the different modality levels (including POS) are insignificant for all 4 sets of authors. Mean and SD and the results of the ANOVA tests are provided in Appendix A.3, in Table A.12 and Table A.12, respectively.

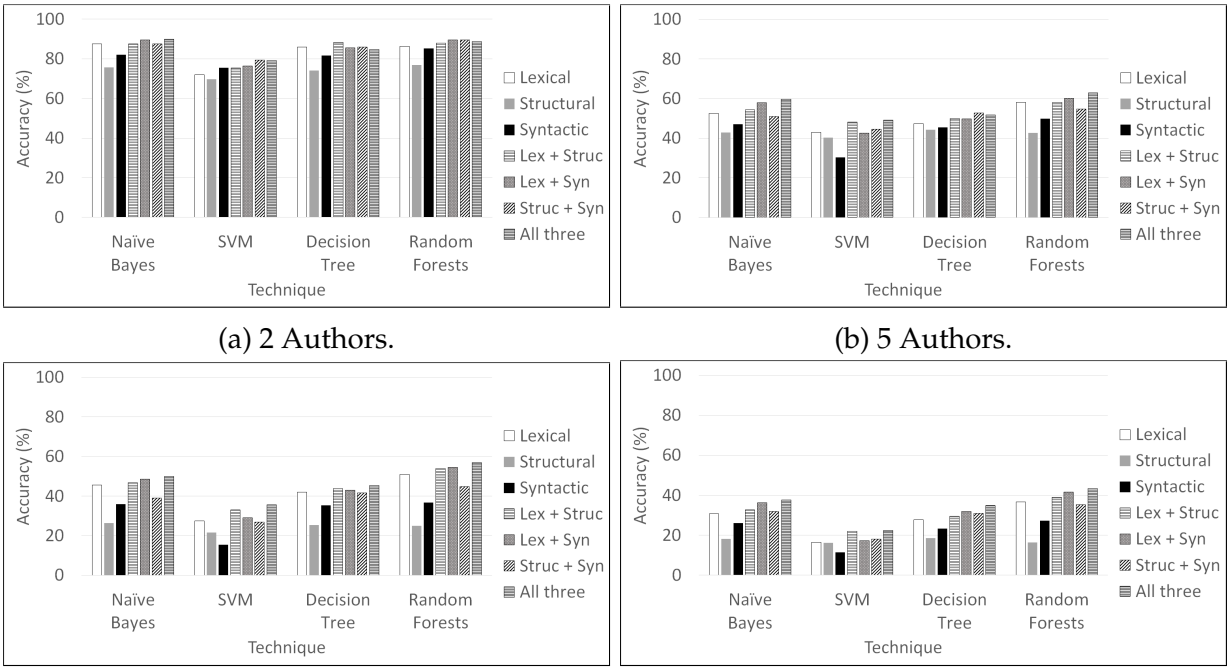
3.5.4 RQ4. When Using Classification Techniques, How Important Is It to Use All Three Categories of Stylometric Features (Lexical, Structure, Syntactic)?

In this section, we investigate the effect of adding more features to the attribution process. As described in Section 3.3.1, we have three categories of features, namely: lexical, structural, and syntactic features. The goal of this section is to evaluate which category (or combination of categories) gives the best performance in the attribution process. Additionally, is adding more features helpful for the attribution process or not?

To do that, we performed the attribution process using instance-based classifiers for 2, 5, 10, and 20 authors with 25 tweets per author. For each set of authors we used one of the following seven sets of features: lexical (Lex), structural (Struc), syntactic (Syn), lexical and structural (Lex + Struc), lexical and syntactic (Lex + Syn), structural and syntactic (Struc + Syn), and all three sets of features. Figure 3.10 shows the results of these experiments.

Figure 3.10 shows a variation in the results as the number of authors change (Mean and SD are provided in Appendix A.3-Table A.14). To help explain this variation, we used ANOVA tests at the $\alpha = 0.05$ level to evaluate the significance of the difference for each set of authors.

For 2 authors, the difference was statistically insignificant $[F(6, 21) = 2.48, p = 0.056]$. For 5 authors, the ANOVA test showed a statistically significant difference $[F(6, 21) = 2.60, p = 0.047]$, but a post hoc Tukey test showed that the difference is statistically insignificant for all the possible pairwise evaluations (Results are provided in Appendix A.3-Table A.15). As mentioned earlier, it is possible to have contradicting results between the ANOVA and the Tukey tests due to the difference in sensitivity for each test, as explained by Simon (2005). These results are in line with literature on English, which suggests that the problem of authorship attribution is relatively easy when the number of authors is small (Luyckx and Daelemans, 2011); therefore, a small number of features is enough for a classification algorithm to reach its best performance



(a) 2 Authors.

(b) 5 Authors.

(c) 10 Authors.

(d) 20 Authors.

Figure 3.10 Evaluating feature categories with instance-based classifiers.

on a small number of authors (in this case, 2 and 5 authors), given enough training samples for each author.

In contrast, an ANOVA test for 10 authors showed a statistically significant difference among the seven sets of features $[F(6, 21) = 3.39, p = 0.017]$. The results of a post hoc Tukey test (Provided in Appendix A.3-Table A.16) showed that the difference is statistically significant only for the pairwise comparison between Structural features and using all three categories of features $[p = 0.024]$. This suggests that as the number of authors increased from 5 to 10, the classification algorithms benefited from adding more features. As Figure 3.10.c shows, structural features came in last for 3 out of four classifiers. However, using other features was statistically insignificant for all the cases except for using all the feature categories together.

Finally and similar to the first case with 2 authors, the difference for 20 authors was statistically insignificant for all seven feature categories. $[F(6, 21) = 2.38, p = 0.064]$.

This indicates that with this large number of authors, classification algorithms are not able to perform well regardless of the number of features that are used. Based on the result of this experiment, we believe that applying authorship attribution on a large scale, i.e., with a much larger set of candidate authors will not be possible by using classification techniques. Instead, one should investigate developing new techniques for authorship attribution, or propose a new feature representation that can be used with traditional classification techniques.

3.6 Visualizing the Result of the Attribution Process

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

The visualization tool provided by Ding et al. (2015) was initially designed for English emails. We adapted it to work with both Arabic and English, while no modifications were required for it to work with tweets. Specifically, we used Microsoft Translator[10] to detect the language used in the anonymous text, and based on the results we changed the POS tagger and the text direction in text boxes. We do not translate POS tags to Arabic for readability issues. This automatic detection of language allows for easier incorporation of other languages by simply including the POS tagger in the source code and specifying the text direction for the added language.

The approach of using the Hue, Saturation, and Lightness to encode the scores and

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

calculate the value per parameter is explained in detail by Ding et al. (2015) and will be omitted from this paper for brevity.

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

1. *A tuple of C-scores for each feature.* For each feature that was used, on all three modality levels, a score is provided for each author based on their writing samples. These values are calculated as explained in Section 3.3.2, and a sample of the output is provided in Figure 3.11. The highlighted feature is the word "قبل". This word appears only for author 1, and therefore it has scores of 0 for the other two authors because its frequency in their writing samples is 0.

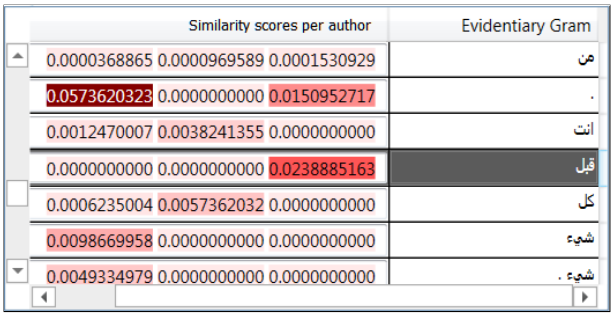


Figure 3.11 A sample of features' scores per author. The score for author 1 is the first column from the right.

2. *The authors' scores projected on the anonymous text.* As Figure 3.12 shows, the whole authors' profiles are projected on the anonymous text to allow for visual comparison between the candidate authors. A sentence in the anonymous text is represented with three lines: the line in the middle shows the sentence as it is and is used to show the word-level n-grams. The HSL for a word is modified to reflect its score for a particular author. The upper line looks empty; however, it is used to reflect the scores of character-level n-grams. Finally, the lower line contains the corresponding POS tag for each word[11] and is used to reflect the scores of POS-

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

level n-grams. Figure 3.12 shows the similarity between an anonymous text and three authors' profiles: Author 1, Author 2, and Author 3. The figure suggests that Author 3 is the most plausible author as it shows a high score (represented with a dark highlight) for using commas, the word "قبل" that appears only for Author 3 and does not appear for the other two authors and the use of full stops.

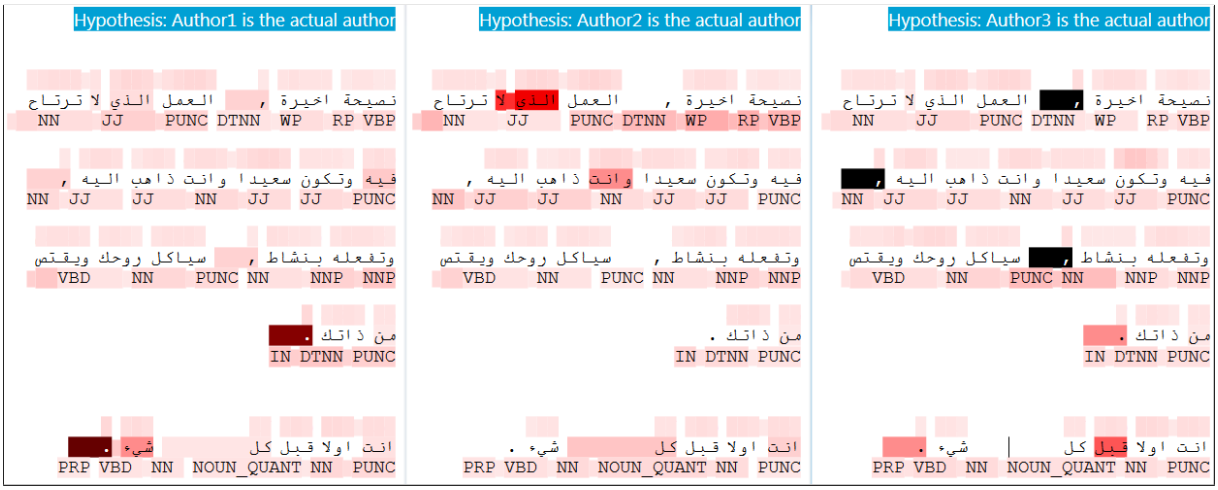


Figure 3.12 The results of the attribution problem for 3 authors and 25 tweets each.

3. *A cumulative features score*. As shown in Figure 3.12, both Author 1 and Author 2 have similarities to the anonymous texts, but from the number of similar features and the color intensity, the user is expected to identify the most plausible author. However, as the number of candidate authors and the features increase, it is expected that this identification task becomes harder.

   Figure 3.13 presents a cumulative feature score for each author that is based on adding the score of each individual feature, starting with the feature that appears first in the anonymous text. These features include character- word- and POS-level features. As shown in the figure, all the scores start and increase by the same level which can be verified using Figure 3.12. For the first few features

(until feature 20) the similarity scores are low for all the authors, then (at feature 55) and as more features are seen, the similarity scores are clearly different. Even before reaching feature 68, it is clear that Author 3 has accumulated the highest similarity score.

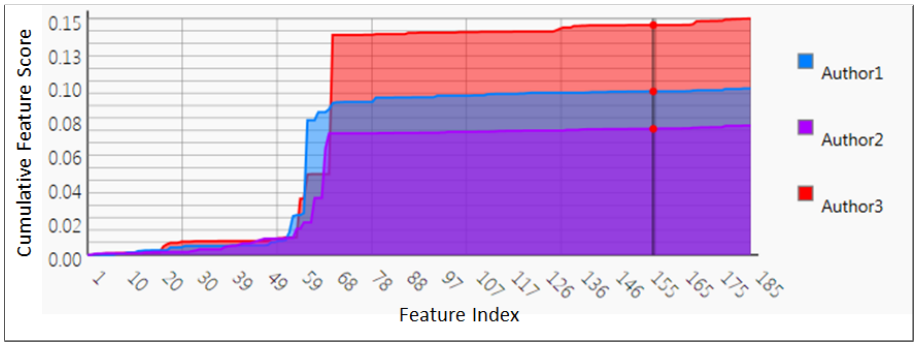


Figure 3.13 A cumulative feature score. The area under the curve represents the difference in authors' scores.

4. *The prediction and the confidence based on each modality level separately*. Finally, we provide the typical output of an attribution process. Figure 3.14 shows the prediction from each modality level and its confidence. The overall prediction is chosen based on the highest normalized score as explained in Section 3.3.2, and the overall confidence is the maximum confidence for the modalities whose prediction matches the overall prediction. In this example, all the modalities' predictions match the overall prediction, and the overall confidence is the maximum of all the confidence values.

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

Figure 3.14 The most plausible author and the confidence for each modality level.

3.7 Conclusion, Limitations, and Future Work

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

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

Among the limitations that still face the n-gram attribution approach is the scarcity of text, whether that is in the anonymous text or the writing samples of the authors. The effect can be seen when very few features appear in both the anonymous text and the writing samples; therefore, it will be very hard to compare the visualized writing styles of three or four authors using a tweet with 3–4 words. Additionally, recent studies showed that current authorship attribution techniques capture the topic in addition to an author's writing style. This means that if the anonymous text is about a topic that is not represented in the author's writing style, then the performance of the attribution techniques will decrease drastically. Using n-grams on various modality levels instead of only using word-level n-grams partially mitigates the issue of the topic. However, there is a need for a better representation of an author's style.

This work aims at laying the foundation for future work in Arabic authorship at-

tribution for short texts. We hope that this work will open the door for further work on Arabic in order to keep up with the work on other languages. In addition to investigating new techniques for style representation, such techniques should utilize the huge development in deep learning, specifically in the representation learning domain. Some examples on such development are the Language preprocessing tool, e.g. Stanford NLTK (Manning et al., 2014b) and pretrained language models of Arabic text, e.g., (Antoun et al., 2020). These tools are particularly important because applying deep learning directly for authorship attribution will be ineffective, due to the small size of the data available for training.

Chapter 4

The Topic Confusion Task: A Novel Evaluation Scenario for Authorship Attribution

In the previous chapter, I evaluated existing authorship attribution techniques that were used for the English language on Arabic tweets. The evaluation has shown that existing approaches, more specifically, the hand-engineered features, need to be adapted every time a new domain or a new language is investigated. For example, and similar to (Layton et al., 2010), stylometric features that were used in authorship attribution for English web forums, e.g., ratio of characters to new lines, font color and font size, had to be removed because such features are nonexistent in tweets. Alternatively, Arabic function words replaced the English one, and ratio of diacritics to alphabets was introduced.

In this chapter, I turn to investigate the writing style features for the same language and the same domain. In particular, literature on authorship attribution has shown that existing techniques are influenced by the topic of the document and may attribute a document to the wrong author merely because the author's writing samples are on

the same topic as the investigated document. To alleviate this problem, researchers proposed the cross-topic scenario that requires the training and the testing documents to have unique topics such that a topic that is seen in training is not seen in testing.

The issue that I have identified in this scenario is that, while it prevents authorship attribution techniques from using topic cues to identify the author, it does not allow us to study which features are affected by the topic cues more than others. In this chapter, I propose a new benchmark to evaluate different authorship attribution approaches with respect to their susceptibility to topic variations.

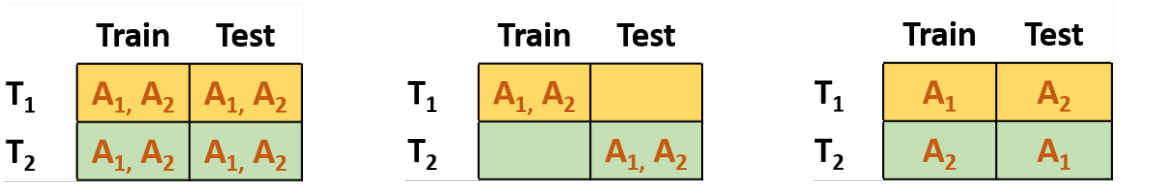
4.1 Introduction

Authorship attribution is the problem of identifying the most plausible author of an anonymous text from a closed set of candidate authors. The importance of this problem is that it can reveal characteristics of an author given a relatively small number of their writing samples. Early approaches to authorship attribution depended on manual inspection of the textual documents to identify the authors' writing patterns, and Mendenhall (1887) showed that word length and frequency statistics are distinct among authors.

Since the first computational approach to authorship attribution (Mosteller and Wallace, 1963), researchers have aimed at finding new sets of features for existing domains/languages, adapting existing features to new languages or communication domains, or using new classification techniques, e.g., (Abbasi and Chen, 2005b; Stamatatos, 2013; Silva et al., 2011; Layton et al., 2012a; Iqbal et al., 2013; Caliskan-Islam et al., 2015; Zhang et al., 2018; Altakrori et al., 2018; Barlas and Stamatatos, 2020). Alternatively, motivated by the real-life applications of authorship attribution different elements of and constraints on the attribution process have been investigated (Houvardas and Stamatatos, 2006; Luyckx and Daelemans, 2011; Goldstein-Stewart et al.,

2009; Stamatatos, 2013; Wang et al., 2021).

Currently, authorship attribution is used in criminal investigations where a domain expert would use authorship techniques to help law enforcement identify the most plausible author of an anonymous, threatening text (Ding et al., 2015; Rocha et al., 2016). Explaining both authorship attribution techniques and their results is crucial because the outcome of the attribution process could be used as evidence in the courts of law and has to be explained to the jury members.



(a) Same-topic (b) Cross-topic (c) Topic-confusion (Proposed)

Figure 4.1 Authorship attribution scenarios. (T: Topic, A: Author)

Researchers have investigated same-topic (Fig. 4.1a) and cross-topic (Fig. 4.1b) scenarios of authorship attribution, which differ according to whether unseen topics are used in the testing phase. The cross-topic setting is considered more realistic than the same-topic setting, but it causes the performance of well-known authorship attribution techniques to drop drastically. This drop is attributed to the topic-writing style entanglement problem where existing writing style features are capturing the topic variations in the collected documents rather than the authors' writing styles.

Traditionally, the evaluation of new authorship methods or writing style features for authorship attribution has been based on the difference in the accuracy either on the attribution task or in ablation studies. While this methodology enhanced the performance on the downstream task and helped answer *which* features perform well, there is a need for methods that can help us understand *why* certain features are performing better than others. Specifically, do these newly proposed features/techniques actually capture the stylistic variations of an author, or are they simply better at picking

out sub-topic cues that correlate with each author?

In this work[1], we propose a new evaluation setting, the topic confusion task. We propose to control the topic distribution by making it dependent on the author, switching the topic-author pairs between training and testing. This setup allows us to measure the degree to which certain features are influenced by the topic, as opposed to the author's identity. The intuition is as follows: the more a feature is influenced by the topic of a document to identify its author, the more confusing it will be to the classifier when the topic-author combination is switched, which will lead to worse authorship attribution performance. To better understand the writing style and the capacity of the used features, we use the accuracy and split the error on this task into one portion that is caused by the models' confusion about the topics, and another portion that is caused by the features' inability to capture the authors' writing styles.

The primary contributions of this work are the following:

- We propose topic confusion as a new evaluation setting in authorship attribution and use it to measure the effectiveness of features in the attribution process.
- Our evaluation shows that word-level n-grams can easily outperform pretrained embeddings from BERT and RoBERTa models when used as features for cross-topic authorship attribution. The results also show that a combination of n-grams on the part-of-speech (POS) tags and stylometric features, which were outperformed by word- and character-level n-grams in earlier work on authorship attribution can indeed enhance cross-topic authorship attribution. Finally, when these features are combined with the current state of the art, we achieve a new, higher accuracy.
- We present a cleaner, curated, and more balanced version of the Guardian dataset to be used for future work on both same-topic, and cross-topic authorship attri-

[1]Code will be made available on https://malikaltakrori.github.io/

bution. The main goal is to prevent any external factors, such as the dataset imbalance, from affecting the attribution results.

4.2 Related Work

The first work that used a computational approach is (Mosteller and Wallace, 1963), which used the Naïve Bayes algorithm with the frequency of function words to identify the authors of the Federalist papers (Juola, 2007). Research efforts have aimed at finding new sets of features for current domains/languages, adapting existing features to new languages or media, or using new classification techniques (Frantzeskou et al., 2007; Iqbal et al., 2013; Stamatatos, 2013; Sapkota et al., 2014, 2015; Ding et al., 2015; Altakrori et al., 2018).

Recent attempts have been made to investigate authorship attribution in realistic scenarios, and many studies have emerged where the constraints differ from the training to the testing samples such as (Bogdanova and Lazaridou, 2014) on cross-language, (Goldstein-Stewart et al., 2009; Custódio and Paraboni, 2019) on cross-domain /genre, and finally, (Sundararajan and Woodard, 2018; Stamatatos, 2017, 2018; Barlas and Stamatatos, 2020, 2021) on cross-topic.

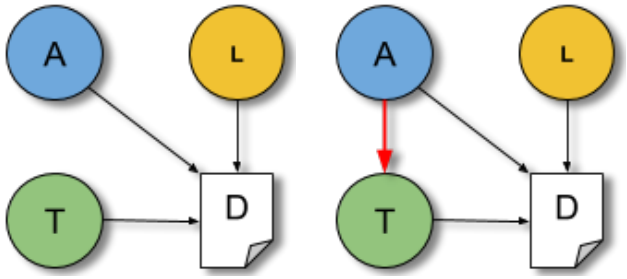
Stamatatos (2017, 2018); Barlas and Stamatatos (2020, 2021) achieved state-of-the-art results on cross-topic authorship attribution. (Stamatatos, 2017, 2018) proposed a character- and word- level n-grams approach motivated by text distortion (Granados et al., 2012) for topic classification. In contrast to (Granados et al., 2012), Stamatatos kept the most frequent words and masked the rest of the text. Barlas and Stamatatos (2020, 2021) explored the widely used and massively pretrained transformer-based (Vaswani et al., 2017) language models for authorship attribution. Specifically, they trained a separate language model for each candidate author with a pretrained embeddings layer from ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), GPT-

2 (Radford et al., 2019) and ULMFit (Howard and Ruder, 2018). Each model was presented with words from the investigated document, and the most plausible author for that document is the one whose model has the lowest average perplexity.

4.3 The Topic Confusion Task

4.3.1 Theoretical Motivation

Figure 4.2a shows the assumed relationship diagram between a document, its author, its topic, and the language rules[2] that govern the writing process (Ding et al., 2019). According to Ding et al. (2019), these are the factors that affect the process of writing a document. Given a topic's distribution over words, the author picks a subset of these words and connects them using the language rules which govern what words accompany these topical words and how sentences are structured.



(a) Assumed. (b) Proposed.

Figure 4.2 The relationship diagram between the topic (T), the author's style (A), the language (L), and the document (D).

Eq. 4.1 shows the joint probability while ignoring the language model, and assum-

[2]There could be other unknown factors that affect any random variable which the attribution process is not aware of.

ing the topic distribution is independent from that of the author.

$$P(A,T,D) = P(A)P(T)P(D|A,T) \tag{4.1}$$

$$P(A=a|D) \propto \sum_{t}^{T} [P(A=a)P(T=t) P(D|T=t,A=a)] \tag{4.2}$$

During the attribution process, the model is used to predict an author given an anonymous document using Eq. 4.2, which follows from Eq. 4.1 after applying the Bayes rule. The same argument about the topic also applies to the language model, but for simplicity, we only focus on the topic since POS tags have been shown to capture the stylistic variations in language grammar between authors.

Same-topic scenarios assume that the topic is independent from the author and that all the topics are available in both training and testing sets. As a result, assuming a balanced dataset of documents and topics per author, T in the joint distribution will be set to a fixed value, and $P(T=t)$ is constant $\frac{1}{|T|}$, where $|T|$ is the number of topics in the dataset. If the dataset has only one topic, e.g., T= sports then $P(T=sports)$=1 and $P(A=a|D,T)$ is $\propto P(A=a)P(D|A=a)$. This assumption is unrealistic and unintuitive because different authors write on different topics with varying interest in each topic.

In contrast, cross-topic scenarios assume that the topic is independent from the author. This is clear from the cross-topic setup where the topic values are fixed during training and testing. While this setup highlighted a critical flaw in same-topic scenarios and encouraged classification models to rely less on topic cues for authorship attribution, it does not help identify the causes of the errors resulting from changing the topic between training and testing.

Instead, we propose a setting in which the topic is dependent on the author, as shown in Figure 4.2b, but this dependence varies between training and testing. Our

intuition about the effect of the author's writing style on the topic is the following. Consider a topic that has a unique word distribution. When an author writes on this topic, they are bound to generate a slightly different word distribution of that topic in their document. The reason is the limited document length which forces the author to choose a subset of words to describe that specific topic. Now, the topic is dependent on the author's writing choices, and this dependency will vary from one author to another since the same idea can be worded in multiple ways using different word synonyms.

Because we allow the topic to depend on the author, the joint distribution changes from Eq. 4.1 to Eq. 4.3 and the conditional probability of an author given the anonymous document changes to Eq. 4.4.

$$P(A,T,D) = P(A)P(T|A)P(D|A,T) \tag{4.3}$$

$$P(A=a|D) \propto \sum_{t}^{T} [P(A=a)P(T=t|A=a) P(D|T=t,A=a)] \tag{4.4}$$

Now, we can create a scenario where a learning algorithm only sees samples on one topic for a specific author in the training set but a different topic in the test set, then we measure the error caused by this switch. Note that this proposed scenario will not be as easy as the same-topic, introduces new topics at test time, and can help explain the entanglement of the topic and the writing style.

4.3.2 The Proposed Setup

Compared to the standard cross-topic setting, this task can help us understand how a topic affects certain features by showing whether the error is caused by the topic or the features themselves. While the cross-topic setting would give a more realistic performance compared to the same-topic, it lacks any insights on why we got such

results.

We propose a new task to measure the performance of authorship attribution techniques given a confounding topic–author setting. The key characteristic of this task is how we associate the topics and the authors in the training, validation, and testing sets. Given a set of writing samples written by N authors on T topics where the number of authors $N \geq 4$, the number of topics $T \geq 3$, and each author has, approximately, the same number of writing samples on each topic T.

First, we divide the authors into two equal-sized groups: group 1 and group 2. Next, to create the training set, we select two random topics and use writing samples on topic 1 for the authors in group 1 and writing samples on topic 2 for the authors in group 2. For the testing set, we flip the configuration of the topics that we used for the training set. We use writing samples on topic 2 (instead of 1) for the authors in group 1 and samples on topic 1 (instead of 2) for the authors in group 2. Finally, we use the remaining writing samples on the unused topics for the authors in both groups for the validation set. Figure 4.3 shows the setup for the proposed task as an example of having four authors and four topics.

With this setup, we can sub-divide the errors that the model makes on the validation and test sets. In particular, we count the following three cases:

1. **Correct (%)**: The ratio of correctly classified samples to the total number of predicted samples.
2. **Same-group error (%)**: The number of misclassified samples to authors within the same group as the true author divided by the total number of predicted samples.
3. **Cross-group error (%)**: The number of misclassified samples to authors in the other group divided by the total number of predicted samples.

Distinguishing these types of errors allows us to investigate whether features in a classifier tend to be indicative of writing style or topic. In particular, features that

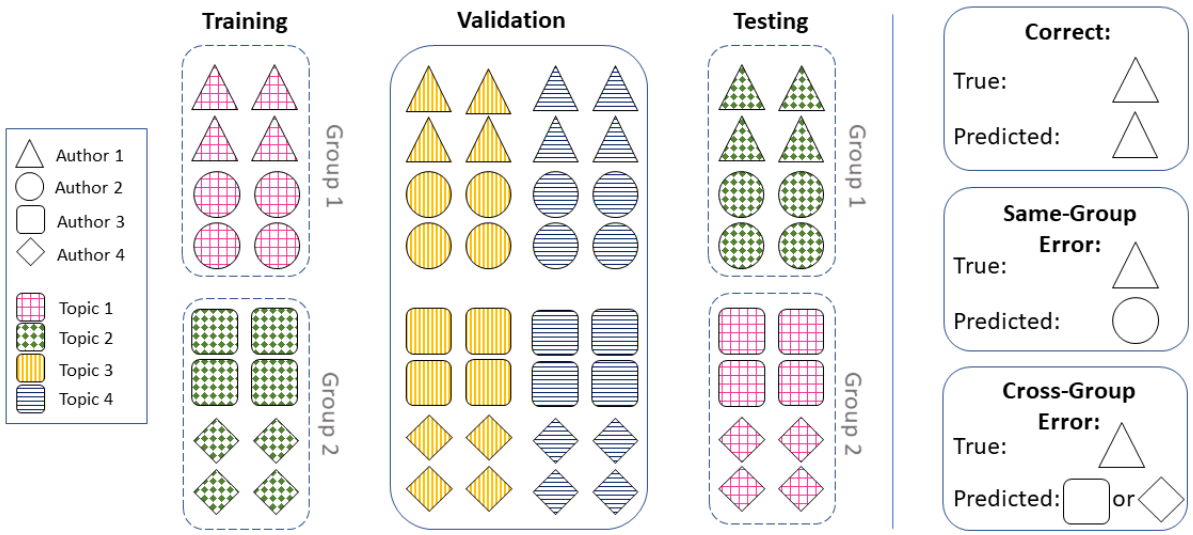


Figure 4.3 Topic confusion task. We use two topics for training and switch them for testing. Two topics are used for hyperparameter tuning. The topic labels are not available for the classifier during training and are only used to distribute the samples over the subsets and calculate the scores.

are invariant to the topic and only capture the authors' writing styles should lead a model to correctly identify the author in the test set. Conversely, features that capture the topic instead of the writing style would lead a model to classify according to the topic, resulting in cross-group errors. Finally, a model that fails for other reasons—either because the writing styles are too similar or because the used features can only partially capture the writing styles—will misclassify samples to authors within the same group.

4.4 Dataset

4.4.1 Data Collection

First, we curated the existing dataset by retrieving the 381 original documents from the Guardian's website. Next, we inspected the authors' names and the topics associated with each article. We excluded the articles that had the wrong topic (e.g., labeled as

"Politics" in the dataset while having a "Society" tag on the website), the ones that appeared under more than one of the previous topics, or were co-authored by multiple authors.

Next, we used the Guardian's API[3] to get all the articles written by each author, filtered them based on the topic, and collected the URLs of these articles and new articles aiming for 10 documents per author per topic. This resulted in a total of 40 documents per author. Note that while some authors have been writing in the Guardian for more than 20 years, they would mostly focus on one topic while occasionally writing on the other four. As a result, we still could not get 10 articles per author on the Society topic. The supplementary material contains full instructions, and the necessary script to get the data and preprocess it.

4.4.2 The Extended Guardian Dataset

We present an extended, curated, and relatively balanced version of the Guardian dataset. One motivation is that the articles in the commonly used version of the dataset contained some HTML artifacts and meta-data from the Guardian's website, and had a number of its articles either on the wrong topic, or written by authors that are not in the dataset. Because of that, we retrieved the original articles and added more articles to balance the number of writing samples per author on each topic. We maintained the same upper limit on the number of documents per author as the original dataset.

Another reason is that as we try to understand the effect of the topic on the attribution process, we need to isolate any external factors that may affect the performance and make the results noisy. For example, in the topic confusion task, we have to use topics with writing samples from all the authors. Otherwise, the model could learn to favor one topic versus the other during training, while on test time, it will have author samples that it did not see during training. Based on that, it will be hard to tell whether

[3] https://open-platform.theguardian.com

these samples will be misclassified due to a lack of training samples or due to a strong topic effect on the attribution process. Although datasets in real life can be imbalanced, this issue can be addressed by randomly excluding some writing samples to make the dataset imbalanced or using proper performance metrics for imbalanced datasets such as weighted accuracy, precision, recall, and F-Score. The number of collected articles and additional descriptive statistics are provided in Table 4.1.

Total number of:		**Number of articles per topic**	
Topics	4	Politics (P)	130
Authors	13	Society (S)	118*
Articles	508	UK (U)	130
Words	3,125,347	World (W)	130
Average number of:		**Number of articles per author**	
Articles / Author	= 39.1 ($SD = 1.5$)	M.K.	35
Articles / Topic	= 127 ($SD = 5.2$)	H.Y.	37
Words / Author	$\approx$ 41 K ($SD \approx$ 6.9 K)	J.F.	38
Words / Topic	$\approx$ 781 K($SD \approx$ 13.0 K)	M.R. and P.P.	39
Words / Document	$\approx$ 1050.2	**The remaining 8**	40

Table 4.1 Descriptive statistics for the extended Guardian dataset (* Has less than 10 articles per author).

4.4.3 Data Splitting and Preprocessing

The Cross-Topic Scenario. In all our experiments, we split the dataset into training, validation, and test sets. For the cross-topic experiments, we followed the same setup in (Stamatatos, 2017). We used one topic for training, another topic for validation and hyperparameter tuning, and the remaining two topics for testing. The number of articles was 127 articles when training on Society and 130 articles otherwise. This setup resulted in 12 different topic permutations. We reported the average overall accuracy on all the 12 configurations.

The Same-Topic Scenario. We combined the 508 articles from all the topics, then split them as follows: 26% for training, 26% for validation, and the remaining 58% for testing. This corresponds to 132 articles for training, 132 articles for validation, and 244 articles for testing. This ensures that the difference in performance between the same-topic and the cross-topic scenarios is not caused by the difference in the number of samples that are used for training/testing. We repeated this process 12 times and reported the average overall accuracy.

4.5 Authorship Attribution Models

In this section, we discuss two groups of authorship attribution models. The first group contains a set of classical models that use hand-engineered features and a classification algorithm. The second group comprises a set of neurally-inspired models motivated by recent advancements in many natural language processing tasks. Such models are considered end-to-end systems where the feature representation is learned by the model as opposed to being hand-crafted and provided to the model.

4.5.1 Classical Features with SVM

This approach uses a set of classical, hand-engineered features with a non-neural classification algorithm similar to what was discussed in Section 2.2.2. We experiment with a wide spectrum of features that include both stylometric features and n-gram features (See Section 2.1). Early work on authorship attribution proposed using stylometric features to represent an author's writing Style. On the other hand, n-gram features were used with most text classification tasks until recent neural representations replaced them.

With all the following features, we used the instance-based approach (Stamatatos, 2009) where a writing style is extracted from every sample separately. A classifica-

tion model is trained on the extracted features to predict the authors of new, unseen samples. We used Pedregosa et al. (2011)'s implementation of linear Support Vector Machines (SVM) as the classification algorithm[4], which is a common choice in authorship attribution (Stamatatos, 2017).

Different classification algorithms can be used with these features. Examples are Naïve Bayes, decision trees, and SVM. We chose to use SVM with linear kernel based on its favorable performance in previous work (Sapkota et al., 2014, 2015; Ding et al., 2015; Stamatatos, 2017, 2018).

Stylometric Features (Iqbal et al., 2008, 2013). We evaluate 371 features including syntactic features and lexical features on both character- and word-level. These features are listed in Table 2.1.

Character-, Word- and POS-level N-Grams (Stamatatos, 2013; Sapkota et al., 2014, 2015). Using n-grams is a common approach to represent documents in authorship attribution. In most text classification tasks, tokenization is done on either the word or the character level. We use both character and word level n-grams in addition to POS-level [5] n-grams which are proven to be an essential indication of style (Ding et al., 2015; Sundararajan and Woodard, 2018).

Masking (Stamatatos, 2017, 2018). This preprocessing technique replaces every character in words to be masked with a (*) and replaces all the digits with a (#). Masked words are chosen based on their frequency in the British National Corpus (BNC), an external dataset. After Masking, tokens are put back together to recreate the original document structure before extracting n-gram features.

[4] Appendix 4.5.3, Tables 4.2 and 4.3 show the range of values and the average optimal parameters that are fine-tuned on the validation set, respectively.

[5] We used the POS tagger from (Manning et al., 2014a).

Combining features. One advantage to using hand-engineered features on the sample level is that these features can easily be combined. First, we evaluated the combination of the stylometric features and POS n-grams. Next, we combined both these features with the other classical features mentioned above.

4.5.2 Pretrained Language Models

Few-Shot BERT and RoBERTa. This is an example of a few-shot (FS) classification with pretrained language models. We used a sequence classification model with a pretrained embeddings layer from the transformer-based non-autoregressive contextual language models BERT Devlin et al. (2019) and RoBERTa (Liu et al., 2019) followed by a pooling layer then a classification layer. We refere to these models in the experiments as FS BERT and FS RoBERTa, respectively.

Given the huge size of these models and the small number of training samples, we decided to freeze the embeddings and train only the classification layer. We used the implementation provided by the HuggingFace (Wolf et al., 2020) library[6].

Author Profile (AP) BERT and RoBERTa (Barlas and Stamatatos, 2020, 2021). We trained a separate neural language model for each author in the dataset where the embedding layer is initialized with embeddings from BERT and RoBERTa. To predict the author, we used each language model —or author profile— to calculate the average perplexity of the model for an investigated document. Before attribution, however, the perplexity scores are normalized using a normalization vector (n) to make up for the biases in the output layer of each language model, where n_i equals the average perplexity of profile A_i on the normalization corpus.

Barlas and Stamatatos (2020, 2021) used two normalization corpora during inference: the training set (K) and the testing set without labels (U). The author with the

[6] https://huggingface.co

lowest normalized perplexity score is the most plausible author of the investigated document. Note that assuming the availability of a test set rather than a single document is unrealistic in authorship attribution even if labels were not provided. We evaluated both cases for the sake of completeness.

4.5.3 Hyperparameters

Hyperparameter	Range
k	100, 200, 300, 400, 500, 1000, 2000, 3000, 4000, 5000
f_t	5, 10, 15, 20, 25, 30, 35, 40, 45, 50
n_{ch}	3, 4, 5, 6, 7, 8
n_w	1, 2, 3
$epochs$	2, 5
$vocab\_size$	2000, 5000

Table 4.2 Hyperparameters for masking and n-gram feature representations. k is the threshold for masking, n_w is the word-level and POS n-grams, n_{ch} is the character-level n-gram, and f_t is the minimum frequency threshold in the whole dataset.

Tables 4.2 and 4.3 show the range of hyperparameters and the optimal hyperparameters values, respectively. For FS BERT and RoBERTa, we used the pretrained sequence classification models. These pretrained models do not have hyperparameters for the model structure but only have pretrained configurations. We used the base uncased models, where base refers to the models' size (not large, and not distilled), and trained on all-lower-case text. For the training procedure, we used the following: AdamOptimizer, lr=0.1, Epochs=500, EarlyStopping(min_delta=1e-3, patience=100). Despite the large Epoch value, most models would stop after less than 150 epochs.

We implemented Barlas and Stamatatos (2020) ourselves. The code was made available online in a later version Barlas and Stamatatos (2021). We performed a grid-search hyperparameter tuning for the number of epochs and the vocabulary size. For the topic confusion task, we used epochs=2 and vocab_size=2000 based on the ablation studies on BERT reported in Barlas and Stamatatos (2020).

Method	k	n	f_t	Feat.
Masking (W.)	1,616.7	1.9	7.9	3,265.8
Masking (Ch.)	1,691.7	5.5	18.8	6,416.3
Stylometric + POS	-	1.3	31.3	484.2
Stylometric + POS + n-grams (W.)	-	2.0	12.5	2,481.0
Stylometric + POS + n-grams (Ch.)	-	3.8	38.3	5,355.6

Table 4.3 The average optimal parameters for each feature representation, with the resulting number of features under these settings (k: masking threshold, n: number of tokens in n-grams, f_t: minimum frequency threshold in the dataset, W.: word-level, Ch.: character-level).

4.6 Evaluation Procedure

For each set of features, we used the setup explained in Section 4.3 to create 100 different configurations. For each configuration, we randomly ordered the topics, selected 12 out of the 13 available authors, and distributed the authors to the groups. This setting is considered one single experiment. To account for randomness in the classification algorithm, we repeated every single experiment ten times[7] and reported the average balanced accuracy score and standard deviation.

We decided to omit one author and use the remaining twelve out of the available 13 authors to balance the groups. With this split, the probability of picking the correct author is $\frac{1}{12}$, the likelihood of choosing a wrong author in the same group is $\frac{5}{12}$, and the probability of picking a wrong author in the other group is $\frac{6}{12}$. This case applies if the true author was in either group 1 or group 2. However, suppose we were to use all the 13 authors and divide them into two groups of six and seven authors, respectively. In that case, the probabilities will differ depending on whether the actual author is in the group with six authors or seven authors. In that case, we will need to re-weight the errors based on their probability, and that will complicate the results as we will not be talking about the exact number of samples.

After creating the training, validation, and testing sets we train models for author-

[7]We trained FS BERT and FS RoBERTa only once.

ship attribution. First, the features are extracted from the writing samples. Second, a classification model is trained on the training samples, tuned on the validation set to pick the best hyperparameters, and tested on the testing set. Note that the classifier does not have access to any information about the setup, such as the group configuration or the topic labels.

4.7 Results and Discussion

4.7.1 Topic Confusion Task

Table 4.4 shows the results on the topic confusion task using the proposed measured in section 4.3.2. Correct is the percentage of samples that were correctly classified, same-group error is the percentage of samples that were attributed to the wrong author but within the same group as the correct author, and finally cross-group error is the percentage of samples that were attributed to the wrong author and to the author group that does not contain the correct author —caused by the change in the topic—.

Classical Features with SVM. Compared to stylometric features, a classifier using character n-grams would correctly classify more samples. However, splitting the error shows that using stylometric features will lead to a lower cross-group error, which is associated with the topic shift. Here, the topic shift does not cause the low performance of stylometric features but rather because they partially capture the writing style.

When looking at character- vs word-level n-grams, we see that they have comparable same-group errors while cross-group error is much higher for word n-grams. Our results are in line with the literature on the classical cross-topic authorship scenario, which shows that character n-grams outperform word n-grams while still capturing the topic, which makes character n-grams less influenced by the topic in the attribution task.

Models	Topic Confusion %			Cross-topic %
	↑ Correct	↓ Same-group Error	↓ Cross-group Error	↑ Accuracy
Stylo.	63.1 (4.2)	15.7 (2.7)	21.2 (3.0)	61.2 (3.1)
POS n-grams	72.0 (4.5)	11.5 (2.9)	16.6 (3.3)	71.0 (3.2)
+ Stylo	79.6 (4.0)	8.4 (2.6)	12.1 (2.8)	79.2 (2.7)
Char n-grams	70.1 (6.5)	6.8 (2.4)	23.2 (6.5)	77.3 (2.8)
+ Stylo	73.0 (6.4)	6.5 (2.6)	20.5 (6.1)	-
+ Stylo & POS	76.8 (6.1)	**6.0** (2.3)	17.2 (5.6)	82.8 (2.7)
Word n-grams	62.5 (7.4)	7.9 (2.7)	29.6 (7.4)	77.7 (2.7
+ Stylo	72.4 (6.4)	7.3 (2.3)	20.3 (6.2)	-
+ Stylo & POS	80.3 (5.0)	7.1 (2.7)	12.6 (4.2)	**83.3** (2.6)
Masking (Ch.) *	79.5 (5.6)	6.8 (2.7)	13.8 (5.0)	80.9 (2.6)
+ Stylo & POS	83.1 (4.8)	6.4 (2.7)	10.4 (3.5)	83.2 (3.3)
Masking (W.)	76.8 (5.7)	7.9 (2.9)	15.3 (5.7)	77.9 (4.0)
+ Stylo & POS	**83.3** (4.4)	6.7 (2.7)	**10.0** (3.2)	82.8 (3.3)
FS BERT	33.1 (5.7)	19.9 (5.6)	47.0 (9.0)	37.5 (3.5)
BERT AP (K)	51.6 (7.5)	8.2 (3.1)	40.2 (8.6)	67.3 (4.4)
BERT AP (U)	52.1 (7.3)	8.4 (3.2)	39.6 (8.5)	71.1 (3.3)
FS RoBERTa	39.8 (7.5)	13.1 (5.1)	47.1 (10.9)	51.1 (3.4)
RoBERTa AP (K)	57.8 (7.1)	7.1 (2.9)	35.1 (8.5)	70.8 (2.0)
RoBERTa AP (U**)	58.9 (7.1)	6.8 (2.8)	34.3 (8.3)	75.8 (3.8)
"random chance"	8.3	41.7	50.0	7.7

Table 4.4 Average results (SD) on the topic confusion task and the cross-topic scenario. The last row is random performance. (**Boldface**: Best result per column. ↑ Higher is better. ↓ Lower is better. %: Percentage. *State of the art. ** Has access to the (unlabeled) test set.)

Next, we look at the effect of masking as a preprocessing technique. Specifically, we compare character- and word-level n-gram features before and after masking. Masking infrequent words is evident in the cross-group error between character n-grams and masking on the character level as well as the word n-grams and masking on the word level. Table 4.4 shows the same-group error remained fixed while cross-group error decreased by around 10% and 15% for the character- and the word-level, respectively.

Combining features. We evaluated the effect of combining both stylometric features and POS n-grams with character- and word-level n-grams with and without masking. The results of combining both stylometric features and POS n-grams with all the other features have decreased the cross-group error significantly, which resembles less confusion over the topic. On the other hand, the same-group error was reduced by merely one sample at max in most cases.

Pretrained Language Models. Surprisingly, such models performed very poorly on this topic confusion task regardless of the attribution approach being used with them. According to the results, these models have a much larger cross-group error which is associated with the topic shift.

One potential explanation for this behavior is that in authorship attribution, two words would have similar embeddings if they appear in a similar context *and* are used by the author in their writing samples. Consider the words 'color' and 'colour' for example. These are essentially the same word but with different spelling based on whether American or British English is being used. Ideally, these two words would have very similar embeddings, if not identical ones. The distinction between the two is critical because it indicates the author's identity or the language system they use. Authorship attribution techniques highlight these differences and use them to identify the most plausible author of an anonymous document.

4.7.2 Comparing the Performance on the Cross-Topic Scenario

We use the cross-topic scenario on the Guardian dataset to compare the performance of different attribution models to that on the topic-confusion task. Note that it is common to do the evaluation on one of the two cross-topic authorship attribution datasets (Goldstein-Stewart et al., 2009; Stamatatos, 2013) similar to (Goldstein-Stewart et al., 2009; Sapkota et al., 2014; Stamatatos, 2018; Barlas and Stamatatos, 2020, 2021)

The last column of Table 4.4 shows a similar trend to the topic-confusion task where combining stylometric features and POS-level n-grams to other classical features results in better authorship attribution. Notably, the combination of stylometric features, POS- and word-level n-grams outperforms the state-of-the-art. Additionally, adding stylometric features and POS-level n-grams to masking (Ch) and masking (W) achieved better performance than state-of-the-art, but the difference was statistically significant only when compared with masking (Ch.) ($P = 0.04$).

Finally, consider two completely different approaches to authorship attribution, namely BERT AP (U) and a linear SVM with POS-level n-grams. Now, note how the accuracy alone on the cross-topic scenario does not provide any insights on why these two models perform very similarly. In contrast, the cross-group error in the topic-confusion task shows that a linear SVM with POS-level n-grams has a much lower error, hence, less affected by the change in topic compared to BERT AP (U).

4.7.3 Cross-Topic Authorship Attribution

As shown in Table 4.5, by combining the stylometric features and POS tags with n-gram features we achieve the highest accuracy of 83.3%. This is in line with our findings in the topic confusion task in Sec. 4.3. The difference between using all the features ($mean = 83.26$, $SD = 2.63$) and the character-based masking approach ($mean = 80.89$, $SD = 2.59$) is statistically significant ($P = 0.04$)[8].

It is also worth noting that by using only stylometric features with POS n-grams we can achieve similar results to the masking approach with character-level tokenization. The difference of 1.7% in favor of the masking approach is statistically insignificant ($P = 0.15$) with a ($mean = 80.89$, $SD = 2.59$) for masking versus a ($mean = 79.22$, $SD = 2.70$) when using stylometric features with POS n-grams.

Furthermore, Table 4.5 shows a 3% increase in the accuracy of the masking ap-

[8]We used a t-Test: Two-Sample Assuming Unequal Variances at the $\alpha = 0.5$ level.

Features	Accuracy
Stylo. + POS	79.2 ± (2.7)
Stylo. + POS + n-grams (W.)	**83.3** ± (2.6)
Stylo. + POS + n-grams (Ch.)	82.8 ± (2.7)
Masking (W.)	77.9 ± (4.0)
Masking (Ch.)	80.9 ± (2.6)
Masking (W.) + Stylo. + POS	82.8 ± (3.3)
Masking (Ch.) + Stylo. + POS	83.2 ± (3.3)
FS BERT	37.5 ± (3.5)
BERT AP (K)	67.3 ± (4.4)
BERT AP (U)	71.1 ± (3.3)
FS RoBERTa	51.1 ± (3.4)
RoBERTa AP (K)	70.8 ± (2.0)
RoBERTa AP (U)	75.8 ± (3.8)

Table 4.5 Average cross-topic classification accuracy (%) on the extended Guardian dataset (W.: word-level, Ch.: character-level).

proach when using character-level tokenization. This outcome is in line with the findings in (Stamatatos, 2017). The difference between word-level n-grams ($mean = 77.90$, $SD = 4.03$) and character-level ($mean = 80.89$, $SD = 2.59$) is statistically insignificant ($P = 0.05$). Similarly, the difference between combining the stylometric features and POS-grams with word-level n-grams ($mean = 83.26$, $SD = 2.63$) versus with character-level n-grams ($mean = 82.83$, $SD = 2.7$) is statistically insignificant ($P = 0.71$).

Finally, the difference between the state-of-the-art approach which is masking on the character-level from one side, versus stylometric features and POS tags combined with either character-level n-grams ($mean = 80.89$, $SD = 2.59$), masking on the word-level ($mean = 82.80$, $SD = 3.34$) or masking on the character-level ($mean = 83.17$, $SD = 3.33$) is statistically insignificant ($P = 0.10$, $P = 0.98$, and $P = 0.80$, respectively.). The only statistically significant difference ($P = 0.0.4$) was with stylometric features and POS tags combined with word-level n-grams ($mean = 83.26$, $SD = 2.63$)

4.7.4 Ablation Study on the Cross-Topic Scenario

We conclude our experiments with an ablation study to see the contribution of each set of features to the overall accuracy. Similar to the experiments above, we perform a grid search over all the hyperparameters f_t and n.

Features	**Accuracy** $\%$
Stylo.	61.2 $\pm$ (3.1)
POS	71.0 $\pm$ (3.2)
W. n-grams	77.7 $\pm$ (2.7)
Ch. n-grams	77.3 $\pm$ (2.8)
Stylo. + POS	79.2 $\pm$ (2.7)
Stylo. + POS + n-gram (W.)	**83.2** $\pm$ (2.6)

Table 4.6 Ablation study: classification accuracy (%) on cross-topic scenario. (Stylo.: Stylometric, W.: word-level)

As shown in Table 4.6, each feature set on its own does not achieve the same performance as by combining all of them. We also confirm the previous results where, even in the cross-topic scenario, n-grams outperformed stylometric features by a large margin. We evaluated the significance of the difference between the top three accuracy groups. The results show that the difference between Set (3) ($mean = 77.7$, $SD = 2.69$) and Set (4) ($mean = 79.3$, $SD = 2.7$) is statistically insignificant ($P = 0.21$) while it is significant ($P < 0.01$) between Set (4) and Set (5) ($mean = 83.3$, $SD = 2.6$).

4.8 Conclusion

In this work, we proposed the topic confusion task, which helps us characterize the errors made by the authorship attribution models with respect to the topic. Additionally, it could help in understanding the cause of the errors in authorship attribution. We verified the outcomes of this task on the cross-topic authorship attribution scenario. We showed that a simple linear classifier with stylometric features and POS tags could

improve the authorship attribution performance compared to the commonly used n-grams. We achieved a new state-of-the-art of 83.3% on the cross-topic scenario by resurrecting stylometric features and combining them with POS tags and word-level n-grams, 3% over the previous state-of-the-art, masking-based, character-level approach. Surprisingly, neurally-inspired techniques did not perform well on the authorship attribution task. Instead, they were outperformed by a simple, hand-crafted set of stylometric features and POS-level n-grams and an SVM classifier with a linear kernel.

Chapter 5

A Multifaceted Framework to Evaluate Evasion, Content Preservation, and Misattribution in Authorship Obfuscation Techniques

In the previous chapter, I showed that the newly proposed topic confusion scenario can help evaluate different authorship attribution techniques. Using the analysis from this task, I showed that stylometric features and part-of-speech tags are a strong indicator of the author's writing style while being less affected by the topic changes. I have also shown that pretrained language models are highly susceptible to topic variations. In contrast to the common trend, authorship attribution is one of the few tasks on which these pretrained language models do not perform well.

In an effort to understand which features are more effective in capturing the writing style than others, one can look at the most effective obfuscation techniques. Ideally, such a technique would obfuscate the features with a strong indication of an author's writing style. However, it is critical for an anonymization technique to hide the writing

style of an author while preserving the content of the text that the author intended to share.

In this chapter, I evaluate state-of-the-art obfuscation techniques on detection evasion and content preservation and show that these techniques are inferior to a back-translation baseline. In addition, I investigate a side-effect of successful evasion in the obfuscation task, namely misattribution, and propose a *fairness* measure to characterize this behavior. As a result, I reveal a key weakness in existing state-of-the-art obfuscation techniques in terms of effectiveness and content preservation. Furthermore, I show that a baseline using a pretrained language model is on par with existing techniques given that such techniques write in a generic writing style.

5.1 Introduction

Advances in authorship attribution have been misused against the public to suppress freedom of speech or to persecute whistle-blowers. As a result, a new task, namely, authorship obfuscation, was introduced (Kacmarcik and Gamon, 2006). Authorship obfuscation is the task of masking the writing style of an author of a document to prevent authorship identification techniques from using stylistic patterns to reveal the author's identity.

When a new authorship obfuscation technique is proposed, it is crucial to compare its performance to state-of-the-art obfuscation tools in settings that accurately depict the real-life environment where such a tool may be used. One important assumption that should be made is that a de-anonymizer is likely to use the most competitive authorship identification tool available to identify the author of a document. Using an inferior or brittle authorship attribution technique, or a weak obfuscation baseline will overstate the performance of such obfuscation tools. For example, previous work on obfuscation has evaluated obfuscation techniques against an identification tool that

uses the exact same features and classifier that were used to obfuscate it in the first place (Mahmood et al., 2019). This could lead to misleadingly high performance.

Similarly, obfuscation techniques must convey the same intended message both before and after obfuscation. Therefore, when a new obfuscation tool is proposed, it is evaluated on both the quality of obfuscation and its ability to preserve the content. With the recent development in language models and their ability to generate text, many automatic measures have been introduced to evaluate the quality of this generated text (Novikova et al., 2017), and some of these measures have been used in obfuscation techniques. The problem, however, with existing content-preservation measures is that they only provide an abstract, numerical score that limits the user's ability to pinpoint the part of the text that suffered from loss of information and requires re-modification. Recently, question answering-based approaches were proposed and shown to provide meaningful feedback in the form of questions that tell the user which information in the text has been changed and in which part (Durmus et al., 2020).

The concept of evasion in authorship obfuscation has a critical, potential harmful side-effect that we raise for the first time. In a classification setting, the typical setting in which evasion of obfuscation techniques are evaluated, a classifier has to pick one author from the set of candidate authors. An obfuscation technique may obfuscate a document by imitating another potential author, in effect unfairly "framing" another person. To investigate this behavior, we use an information-theoretic approach to evaluate the potential for misattribution of the obfuscating technique. Specifically, we propose a new evaluation measure; namely, misattribution harm where the goal is to characterize the confidence in the attribution algorithm rather than its output.

In this work, we highlight a number of issues with the existing work on obfuscation with respect to two dimensions: obfuscation effectiveness, and content preservation, and we propose a new evaluation dimension namely, fairness. Our key contributions are the following:

- We show that a carefully selected baseline can outperform state-of-the-art obfuscation techniques.
- We use question answering as an evaluation measure for content preservation instead of token- and embedding-based approaches.
- Using information theory, we conduct a detailed analysis of the harming effect of misattributing a document to a different author to achieve detection evasion.

5.2 A Multifaceted Evaluation Framework

Ideally, authorship obfuscation should only modify the author's writing style in a document while retaining all the original information. However, due to the topic–writing style entanglement, modifying the document is likely to cause information loss, i,e., some content is not preserved. Based on that, obfuscation techniques are evaluated in two dimensions: evasion, and content preservation.

In Section 2.3.2, I introduced the two existing aspects on which authorship obfuscation is evaluated. In the following subsections, I formally describe obfuscation, evasion, and content preservation and discuss the state of the tools used to evaluate them. Finally, I propose a novel evaluation dimension to characterize a potential side effect of successfully evading detection namely, misattribution.

5.2.1 Obfuscation

Let d be a document written by author a^*. To hide their identity, $a*$ uses an obfuscation technique $O : d \rightarrow \hat{d}$ that takes a document d as an input, modifies it, and outputs an obfuscated version of this document, namely, $\hat{d}$ such that $d \neq \hat{d}$.

For example, suppose we have a document d, where d = "The decision caused the team a big loss!", which was written by author $a^* = Q$. Q uses an obfuscation technique O that modifies the document d by changing it to $\hat{d}$, where $O(d) = \hat{d}$ = "The advice

caused the team a huge loss".

5.2.2 Evading detection

Let I be an authorship identification technique $I : (d, T) \rightarrow a_i$ that takes a document d and a set of candidate authors of this document T as input, and outputs a_i as the most plausible author of this document d. Let $T = [a_1, a_2, \ldots, a_n]$ and $n =\mid T \mid$. We say that author a^* has evaded detection using the obfuscation tool O if $I(d, T) = a*$, $I(O(d), T) = a_i$, where $a^* \neq a_j$; and $a^*, a_j \in T$. Note that, if $I(d, T) \neq a*$ then d does not require obfuscation against I.

To evaluate the obfuscation performance over a whole test dataset D, let $S : (a_i, a^*) \rightarrow \mathbb{Z} \in [0, 1]$ be an indicator function given by Eq. 5.1.

Finally, let $Accuracy = \sum_i^m S(I(d_i, T), a_i^*)/m$, where $m =\mid D \mid$ is the number of test documents.

$$S(a_i, a^*) = \begin{cases} 1, & \text{if } a_i = a^* \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

Continuing from the example in Sec. 5.2.1, let T be ["G", "Q", "B", "M", "W"], the predicted author before obfuscation, i.e., $I(d, T)$ be Q, and the predicted author after obfuscation, i.e., $I(O(d), T)$ be G. Here, the obfuscation tool O has evaded detection successfully.

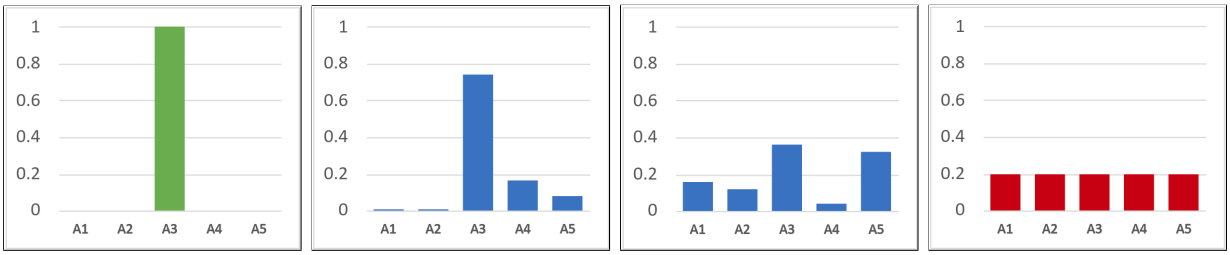
5.2.3 Preserving the content

After evaluating evasion, content preservation is evaluated to investigate whether loss of information has occurred due to obfuscation. An authorship obfuscation technique should maximize the content preservation, or equally minimize the loss of information. After evaluation, the result of this evaluation is communicated to the author to decide whether to accept the obfuscation outcome or reject it if the information loss is drastic.

Formally, let $P : (d, O(d)) \rightarrow \mathbb{R}$ be a content-preservation evaluation tool that takes an original document d and an obfuscated document $O(d)$ as input, compares their content and outputs a content-preservation score that represents the amount of information preserved from the original document d after obfuscation.

For example, suppose that the content-preservation tool of choice is based on the word-level uni-grams overlap between the original document d, and the obfuscated document $O(d)$, where d = "The decision caused the team a big loss!" and $O(d)$ = "The advice caused the team a huge loss". Suppose that splitting d, and $O(d)$ into word-level uni-grams yields ["The", "decision", "caused", "the", "team", "a", "big", "loss!"] and ["The", "advice", "caused", "the", "team", "a", "huge", "loss"], respectively. Here, the goal is to maximize content-preservation score where $P : (d, O(d)) = 5$.

5.2.4 Fairness, and the potential of mis-attribution harm



(a) Max. confidence. (b) Relatively high. (c) Relatively low. (d) Min. confidence.

Figure 5.1 Examples on confidence levels for a model represented as the probability distribution over all the authors. (The x-axis is $P(A_t|x)$, where x is a test instance)

In real-life applications of authorship identification, misattribution can have severe outcomes. For example, if the obfuscated text is a threatening message, then it is important to identify the real culprit to avoid persecuting and incarcerating an innocent person.

Confidence in the identification outcome is a core concept in authorship identification that has not been emphasized in the obfuscation literature.

Instead of imitating the writing style of one of the candidate authors, an obfuscation technique may output a generic writing style that is difficult to attribute to a specific author. In that case, the identification technique will still provide a candidate author, but its confidence in this output would be low. As a result, if an obfuscation technique can lower the confidence of an identification method, then its outcome will be in a position of doubt, hence, neither the original author nor the identified one will have to suffer.

Formally, let C:(I,d,T) $\rightarrow \mathbb{R}^n$ be a tool that takes as input an authorship identification tool I, a document d, and a set of candidate authors for that document T and outputs a probability distribution over the candidate authors $[c_1, c_2, \ldots, c_n]$, where $c_i = P_I(a_i|d)$ is the likelihood of author a_i being the original author of document d when authorship identification tool I is used, $0 \leq c_i \geq 1$, and $\sum_i^n c_i = 1$.

In this work, we consider the model's confidence to be high when the probability distribution for one author is much higher compared to the other authors. For example, a model is the most confident when it predicts author A_t with probability 1. In contrast, a model is the least confident, or rather clueless, when the probability distribution over all the authors is uniform, i.e., when the probability of each author is $\frac{1}{T}$, where T is the number of authors.

Figure 5.1 shows a synthetic example on a model generating the same author prediction but with different confidence levels. Note that the model can predict the wrong author and have high confidence in its prediction. Luckily, this confidence can be easily measured by computing the Entropy (Eq. 5.2) for the attribution model, and the effect of misattribution harm can be characterized by the difference in Entropy before and after obfuscating a document (d).

$$H(X) = -\sum_{t=1}^{n} P(x_t) \log_2 P(x_t) \tag{5.2}$$

Furthermore, this approach can provide a finer-grained measure of performance

than identification accuracy. For example, let us assume that the attribution model had the maximum confidence in its prediction before obfuscation. Then, after obfuscation using technique A, the model's confidence dropped to the same level as Fig. 5.1b. Alternatively, after obfuscation using technique B, the model's confidence dropped to the same level as Fig. 5.1c. Clearly, both techniques generated the same author prediction, and so, both techniques are either equally right or equally wrong. However, technique B would be considered better because it caused the attribution model to have lower confidence in its prediction.

5.3 Experimental Setup

In brief, we conducted the evaluation as follows. We started by establishing the authorship identification accuracy on the original datasets. Note that, the training and testing split is predefined for each dataset as shown in Table 5.1. For validation, however, we shuffled the training set and took 20% of the samples for validation.

We followed that by creating different obfuscated copies of the test sets, one for each obfuscation technique. Next, we evaluated the detection evasion and misattribution on each obfuscated copy in one step. We concluded our evaluation with content preservation. We provide below the details of each step separately.

5.3.1 Corpora

For this work, we use two different corpora: the Extended Brennan–Greenstadt Corpus (EBG) dataset (Brennan et al., 2012) and the Reuters Corpus Volume 1 (RCV1) (Teahan, 2000; Khmelev, 2000; Kukushkina et al., 2001), commonly referred to as the C50 dataset. For each dataset, we use two authors configurations: five authors, and 10 authors. We provide corpus statistics in Table 5.1.

	C50				EBG			
Authors	**5**		**10**		**5**		**10**	
Training set								
Docs	75		150		55		110	
Docs / authors:	15	(0.0)	15	(0.0)	11	(0.0)	11	(0.0)
Avg. doc Len (W)	478	(46.4)	452	(60.8)	496	(6.1)	494	(4.8)
Avg. doc Len (C)	3007	(273.1)	2861	(366.9)	3157	(24.0)	3120	(41.8)
Testing set								
Docs	75		150		55		110	
Docs / authors:	15	(0.0)	15	(0.0)	7	(4.0)	6	(3.2)
Avg. doc Len (W)	480	(86.2)	479	(77.6)	496	(14.1)	497	(12.5)
Avg. doc Len (C)	3032	(567.2)	3036	(473.9)	3068	(102.7)	3046	(130.8)
Total docs	150		300		90		169	

Table 5.1 Corpus statistics. (Doc: Document, W: Words, C: Characters) numbers are reported using the rounded Mean and (SD).

5.3.2 Authorship Obfuscation

The evasion performance of an obfuscation technique is compared to a set of baselines as well as state-of-the-art obfuscation techniques. Here, the role of a baseline is to set a lower bound on the performance while requiring little knowledge about the problem, and a fairly low effort to use.

In this work, we use well-tuned baselines that are expected to be competitive with obfuscation techniques as opposed to using simple and primitive ones. An example of excluded baselines is Random Replacement which tries to obfuscate a document by replacing words in that document with a random word from the author's vocabulary set, or with a synonym from a dictionary. Such baselines have been explored heavily in the literature and are known for their poor obfuscation performance and incoherent output.

More specifically, we use a neural machine translation model in the back-translation baseline to replace statistical models that were used in previous studies (Brennan et al., 2012; Keswani et al., 2016). Additionally, we use a contextual language model namely,

BERT to replace words based on their context, instead of replacing them with synonyms or random words from the author's vocabulary set.

Back Translation (BT) uses Facebook's many-to-many translation model (El-Kishky et al., 2020; Fan et al., 2021; Schwenk et al., 2021) implemented by the HuggingFace (Wolf et al., 2020) library [1]. This model has two advantages. Firstly, it is open-source and its results can be replicated in contrast to commercial translation products that are costly and can be replaced at any time.

Secondly, this model translates between languages directly without using English as a reference/pivot language. Many of the existing neural machine translation models use English as a *pivot language* where translation is done either *from* English or *to* English. For example, if the task is to translate from French to Chinese, one has to translate from French to English, then from English to Chinese. This approach defeats the whole point of multi-hop translation where the goal is to use the differences between languages in phrasing the same idea to change the writing style of a sentence.

Lexical substitution using BERT (LSB) (Mansoorizadeh et al., 2016) masks random words in a sentence, then use the BERT language model to replace these words with ones that fit the context.

Mutant-X (Mahmood et al., 2019) replaces words based on their GloVE word embeddings given that the candidate replacement has the same sentiment. This technique requires knowledge of the authorship attribution classifier, specifically, the probability of each author, to do the obfuscation.

Heuristic Obfuscation Search (A*) (Bevendorff et al., 2019) was originally developed as an imitation approach to obfuscation. The algorithm requires a target author

[1] https://huggingface.co/facebook/m2m100_418M

profile which is the tri-grams frequency. This rule-based approach changes the text while incurring costs, and the goal is to generate a document with high similarity to a target profile with minimum cost.

Anon. Tech.	EBG dataset		C50 dataset	
	5 Authors	10 Authors	5 Authors	10 Authors
None	96.4	77.6	76.0	67.3
A*	93.5	71.1	**72.0**	**64.0**
Back Translation	**84.0**	**64.2**	73.3	65.3
Lexical Sub (BERT)	91.5	78.4	76.0	67.3
Mutant-X	86.4	73.6	74.7	66.7

Table 5.2 Obfuscation performance measured by the drop of the attribution accuracy (%). Identification using word masking and character n-grams as features, and a linear SVM classifier. (Lower accuracy is better. **Bold**: best result per column.)

5.3.3 Authorship Identification

When evaluating evasion in authorship obfuscation, it is important to use a state-of-the-art authorship identification technique. In the authorship attribution domain, it is well-established that a cross-topic authorship identification tool will have a realistic performance that mimics real-life applications (Goldstein-Stewart et al., 2009; Sundararajan and Woodard, 2018; Stamatatos, 2017, 2018; Custódio and Paraboni, 2019; Barlas and Stamatatos, 2020, 2021; Altakrori et al., 2021).

Because of that, we use the state-of-the-art (Altakrori et al., 2021) cross-topic, authorship identification technique to evaluate evasion of obfuscation techniques namely, (Stamatatos, 2018). The main idea of this approach is to mask words in a document, where masking is done by replacing the characters in the word with asterisks, then using the word- or character-level n-grams to represent as features. The choice of which words are masked is based on the hyperparameter k. In a document, any word that is not in the k-most frequent words in the British National Corpus (BNC) must be

masked. After masking and extracting the n-gram features, a Support Vector Machines (SVM) with a linear kernel is used as a classifier.

5.3.4 Content preservation

To evaluate the content preservation, we chose the EBG dataset with the ten authors' configuration. From all the original test documents and the four obfuscated versions, we randomly selected 10% of the documents. These documents were split into sentences, and a sentence was included or excluded from the evaluation samples based on a coin flip. This resulted in 212 sampled sentences, an average of 42 sentences per obfuscation technique. To avoid cherry-picking samples that favor one metric vs. another, we did not exclude any of the sampled sentences. However, we discuss the consequence of this in the results section below.

To evaluate the content preservation of these samples, we used HuggingFace implementation for both token-based and model-based evaluation tools. For the question-answering approach, we used (Scialom et al., 2021) that generated the questions from the original document instead of needing a reference[2]. In brief, we used the following measures to evaluate content preservation.

- BLEU (Papineni et al., 2002).
- METEOR (Banerjee and Lavie, 2005).
- ROUGE-1, 2, and L (Lin, 2004).
- BERTScore (Zhang et al., 2020).
- QuestEval (Scialom et al., 2021).

5.3.5 Characterizing mis-attribution

As described in Sec. 5.2.4, we calculate the change in entropy before and after training. We follow the same training procedure that was used for identification. However,

[2]The authors made their code available online.

instead of using the authors' probabilities to find the most likely author, we calculate the entropy for that output distribution.

Note that, while we are measuring the model's entropy at test time for this task, it is important to train and validate the model using the traditional accuracy-based approach, i.e., by minimizing the cross-entropy. This is important because maximizing the entropy at validation time is not guaranteed to yield the best-performing model.

To clarify, assume that model M_1 predicts author a_1 as the most plausible author of document d with probability 1.0. Similarly, assume that model M_2 predicts author a_4 as the author of the same document d with probability 1.0 as well. Clearly, only one of the two models can be right. However, both models will have the same entropy value. Therefore, in this task, we pick the model with the highest validation accuracy, then calculate the average entropy of all the test documents.

Finally, we normalize the entropy scores to make them comparable with other content-preservation scores that are bounded between zero and 1. To do that, we divide the entropy scores by the entropy of the uniform distribution with K authors, where K is the number of authors in each dataset.

5.4 Experimental Results

5.4.1 Evaluating Evasion

As mentioned earlier in Sec 5.2.2, the successful evasion of an obfuscation technique is measured [3] by the drop in authorship identification accuracy [4] after obfuscation [5]. In Table 5.2, the first row shows the identification accuracy on the original test documents, i.e., before obfuscation. The rows below it show the identification accuracy

[3]The experiments for this paper were run on a workstation with one GPU of type Quadro RTX 8000, with four CPUs and 32GB of RAM. Run time are estimated by http://www.wandb.com

[4]Authorship identification run time: ∼14.5 Hours

[5]Obfuscation run-time: ∼10 days, that is ∼256 Hrs total.

after obfuscating the test documents. Here, the lower the attribution accuracy after obfuscation the better an obfuscation algorithm at evading detection.

We make the following observations from Table 5.2. First, despite being a baseline, back translation outperforms both obfuscation techniques on the EBG dataset, and comes as a close second on the C50 dataset after A* of Bevendorff et al.. In contrast to the literature, back translation is not a weak baseline anymore.

The other general observation that we make is that identifying the original author —even without obfuscation— becomes much harder as the number of candidate authors increases. Specifically, as the number of authors increased from five authors to ten authors, the authorship attribution accuracy dropped by around %20 and %10 on the EBG and the C50 dataset, respectively.

5.4.2 Content preservation

Table 5.3 shows the result of content preservation using various evaluation metrics. Firstly, it is important to clarify why the original text does not have the highest score (1.0) on token-based metrics. Upon manually investigating the sampled sentences for evaluation, we notices that one of the sentences had only one word, that is "originally,". As a result, Rouge-2 could not find any bi-grams in this sentence and output a 0 score.

Anon. Tech.	Rouge-1	Rouge-2	Rouge-L	BLEU	METEOR	BERTSc.	QuestE
None	1.000	0.981	1.000	0.981	1.000	1.000	0.678
A*	**0.906**	**0.858**	**0.906**	**0.766**	0.867	0.966	0.582
Back Translation	0.704	0.471	0.681	0.312	0.722	0.958	**0.620**
Lexical Sub (BERT)	0.848	0.696	0.845	0.593	0.844	0.965	0.599
Mutant-X	0.902	0.814	0.902	0.746	**0.915**	**0.976**	0.555

Table 5.3 Content Preservation scores on 212 sampled sentences from the EBG-10 dataset.

Naturally, A* has the best performance on the token-based metrics given that most of the modifications are done on the character level, i.e., has a lower tendency to change

words. Similarly, Mutant-X has the highest model-best scores because words are replaced based on their embeddings.

Conversely, back translation has the worst scores in both token-based and model-based measures. In contrast, it has the closest score to the original text using the QA-based approach which, as mentioned earlier, has a better correlation with human scores than token-based and model-based metrics.

5.4.3 Characterizing unfair mis-attribution using entropy

Table 5.4 shows the normalized entropy scores that are used to characterize unfair mis-attribution. The higher the normalized entropy, the closer the probability distribution of the predicted authors is to the uniform distribution. In that case, the model has no preference for one particular author or has low confidence in its outcome.

Anon. Tech.	EBG		C50	
	5 Authors	10 Authors	5 Authors	10 Authors
None	73.9 ± 4.4	83.3 ± 3.8	79.4 ± 6.8	84.3 ± 2.9
A*	78.8 ± 4.9	86.8 ± 4.0	79.0 ± 6.5	84.4 ± 2.1
Back Translation	**82.3 ± 4.3**	**88.8 ± 2.1**	**83.1 ± 4.8**	**87.3 ± 3.1**
Lexical Sub (BERT)	80.5 ± 4.5	84.6 ± 2.5	80.2 ± 6.6	85.3 ± 2.4
Mutant-X	72.6 ± 4.7	85.2 ± 2.8	82.0 ± 5.4	83.2 ± 2.6

Table 5.4 Characterizing the misattribution using the normalized entropy score (%).

Back translation has the best performance on this evaluation metric, measured by the increase in normalized entropy from that before obfuscation (the first row). Our interpretation is that by translating to different languages, back translation is generating text in a generic style that is hard to attribute to one particular author. In contrast, A* tries to imitate a specific author's writing style to avoid detection, while Mutant-X requires a set of candidate authors and a classifier to do the obfuscation, and only stops when the obfuscated text is attributed to a different author.

Tables A.17 and A.18 show the identification accuracy on the left of the table and unnormalized entropy scores to characterize the misattribution behavior. The goal of this table is to show that raw entropy scores are less intuitive than the normalized values bound between zero and one.

5.4.4 Ablation Study

In this section, we conduct a battery of tests on different attribution features. The goal of this study is to characterize the obfuscation performance under different types of writing style features. Table 5.5 shows the result of using different writing style features for identification which vary between stylometric features and content features.

5.5 Conclusion

In this work, we demonstrated the importance of using state-of-the-art evaluation tools to measure the performance of authorship obfuscation techniques. In addition, our experiments revealed that current obfuscation techniques have key weaknesses and have been outperformed by a baseline, namely back translation in multiple evaluation aspects. Furthermore, we identified a critical issue with respect to the fairness of obfuscation techniques. Our proposed misattribution measure investigates the side-effect of a successful detection evasion by identifying another author as the most plausible author of the obfuscated text. As a result, we argue that an attack on the confidence of the identification model, by generating text in a generic style would confuse the identification model and make it unusable in real-life applications. Finally, we argue that the evaluation of authorship obfuscation tools should follow the rapidly evolving domain of evaluation tools while keeping the potential users and real-life applications when developing and evaluating novel obfuscation techniques.

		Identification technique					
		Stylo.		N-grams		Masking	
	Anon. tech.	-	+ POS	Ch.	W.	Ch.	W.
EBG 5	No Anon.	81.2	90.0	91.5	96.4	88.8	96.4
	A*	51.3	78.0	91.5	94.6	76.4	93.5
	Back Translation	59.7	67.3	89.7	96.4	83.1	84.0
	Lexical Sub (BERT)	68.2	80.2	89.7	96.4	95.3	91.5
	Mutant-X	81.2	84.7	91.5	96.4	95.5	86.4
(a)							
EBG 10	No Anon.	58.8	53.9	73.3	75.1	53.1	77.6
	A*	47.8	37.2	74.2	71.8	51.7	71.1
	Back Translation	46.2	43.5	66.3	73.2	59.5	64.2
	Lexical Sub (BERT)	55.0	52.0	71.5	73.7	58.7	78.4
	Mutant-X	55.0	52.2	74.1	77.0	52.3	73.6
(b)							
C50 5	No Anon.	65.3	68.0	84.0	84.0	61.3	76.0
	A*	65.3	66.7	81.3	85.3	58.7	72.0
	Back Translation	64.0	65.3	82.7	81.3	58.7	73.3
	Lexical Sub (BERT)	65.3	68.0	85.3	88.0	62.7	76.0
	Mutant-X	60.0	62.7	84.0	84.0	54.7	74.7
(c)							
C50 10	No Anon.	58.7	69.3	69.3	64.0	53.3	67.3
	A*	56.7	69.3	68.0	61.3	54.7	64.0
	Back Translation	55.3	64.7	65.3	62.0	52.0	65.3
	Lexical Sub (BERT)	56.7	70.7	68.7	62.0	54.7	67.3
	Mutant-X	56.0	67.3	69.3	62.7	52.7	66.7
(d)							

Table 5.5 Obfuscation performance using different sets of features with a Support Vector Machines classifier. (Stylo.: Stylometric, Ch.: Character, W.: Word)

Chapter 6

Discussion

In this chapter, I provide a discussion on the key issues that were addressed in this thesis. I start by highlighting the drastic change the authorship attribution went through starting from using it to solve disputes over authorship plays and books, to using it as evidence in the courts of law. With the application of authorship attribution in a new domain, I discuss the need to visualize the outcome of the attribution process. Next, I reiterate the importance of stylometric features when attempting to capture an author's writing style while minimizing the topic effect and follow this with a discussion on using neural models for authorship attribution and obfuscation. Finally, I conclude this chapter with a discussion of the limitations and potential future directions.

6.1 Authorship Attribution and Forensic Linguistics

To begin with, it is important to highlight how the authorship identification problem changed compared to when it started in the literature domain. For one, early researchers were working on real-life data, and not synthetic data that was collected in a lab setting. This point is important because all the issues or constraints that are currently being discussed, such as the number of candidate authors, the number of sam-

ples per author, or the effect of the topic on the attribution process were non-existent when authorship attribution was used in the literature domain. For example, if we look at the Federalist papers (Mendenhall, 1887), we see that all the letters have the same topic, genre, and media. Similarly, the dispute between Shakespeare and Marlowe was over plays. Researchers on these two datasets did not need to investigate any aspect that might affect the attribution process.

Researchers who adapted authorship attribution to new domains, languages, and media then seem to have followed the same approaches that existed in the literature. For the new domains, the main step is to collect an authorship attribution dataset, then investigate or propose writing style features to evaluate authorship attribution techniques. This phenomenon is commonly referred to as *function creep*.

Function creep describes a situation where a tool is designed to perform a certain function that is considered safe but ends up being used in a more critical application without being evaluated thoroughly similar to a typical application for such a critical domain. In authorship attribution, numerous studies have emerged proposing new writing style representations, each supported with some empirical proof to show that the new writing style representation leads to better authorship identification performance. However, the earliest work on evaluating the effect of dataset size, and the number of candidate authors is attributed to (Luyckx and Daelemans, 2011). In the same direction, Sapkota et al. (2014) and Stamatatos (2013) investigated the effect of the topic on the authorship attribution process.

For such critical applications, it is important to work on evaluation in tandem with achieving higher performance, given that such techniques are already being used in real-life applications and this thesis is one attempt to do that.

6.2 Visualizing the outcome of the attribution process.

In Chapter 3, I investigated the authorship attribution problem for short Arabic text, specifically, Twitter posts. At that point, the larger body of literature on authorship attribution for Arabic text has been focusing on media that is relatively longer than tweets, e.g., books, poems, and blog posts. This adaptation, from English to Arabic and from longer to shorter text, necessitates that the writing style features are changed to match the language and the media regardless of whether we were using stylometric features or n-gram-based features.

When authorship attribution is used in criminal investigations, a domain expert would use the identification techniques to help law enforcement identify the most plausible author of an anonymous, threatening text (Ding et al., 2015; Rocha et al., 2016). Understanding both authorship attribution techniques and their results is crucial because the outcome of the attribution process could be used as evidence in the courts of law and has to be explained to the jury members. As a result, some researchers turned their attention to visualizing this outcome in an intuitive matter that would allow law enforcement officers or members of the jury to make an informed decision about the evidence.

With that in mind, I investigated the performance of the attribution process using both instance-based and profile-based techniques when different classification algorithms are used. The results from this investigation showed that profile-based approaches, specifically using n-grams as writing style feature representation, are competitive to the instance-based approaches.

However, such instance-based models are considered harder to visualize and be used as evidence in courts of law. Some examples of that complexity are the ratio of words to spaces or average sentence length. Profile-based approaches, on the other hand, are simpler given that each profile represents one author which makes their results easier to project on the investigated document in a more intuitive way, which

makes them more favorable to be used as evidence in courts of law.

6.3 On the Effectiveness of Stylometric Features for Authorship Identification.

According to (Holmes, 1998), Augustus De Morgan (1882) was the first to look at the differences in writing styles among authors of religious scripts. De Morgan did not conduct any formal analysis to support his claim, but rather made the observation that "*by merely looking at the words while ignoring their meaning, one can notice that one text has longer words than the other*" (De Morgan and De Morgan, 1882). Motivated by his work, several studies investigated different feature representations of the writing style that can be considered within the same category as word lengths, such as the word and sentence length (Mendenhall, 1887; Yule, 1939).

When authorship attribution techniques were investigated for other languages, many issues became apparent. For example, some languages do not have a clear sentence or word boundaries. In other cases where sentences and words can be identified, automatic parsers and tokenizers are not available for that language. To circumvent this issue, Kešelj et al. (2003) proposed the use of character-level n-grams instead of word-level ones. In that study, the evaluation of such features was done on English, Greek and Chinese books. However, the topic variation among the books in the English dataset was not considered as evident by the list of authors that comprises Shakespeare, Charles Dickens, and John Milton (Kešelj et al., 2003; Peng et al., 2003).

Replacing word-level n-grams with character-level n-grams has many benefits. In addition to solving the aforementioned technical issues with parsers and tokenizers, character-level n-grams mimic lemmatization and stemming, two preprocessing techniques that are commonly used to combine a word that appears in different inflectional forms across a document. For example, instead of treating the words: *play*, *played* and

playing as three different words, they would be counter as a three occurrences of the quad-gram *play*. Similarly, bi- and tri-grams can capture some regular verb tenses, e.g., played, playing, studied, and studying, although not as effective as counting the POS tags of all the verbs. It is important to note that the literature lacks a detailed study that investigates this behavior and only empirical evidence is provided to show that character-level n-grams outperform word-level ones for the task of authorship identification (Stamatatos, 2013).

Stylometric features continued to be used for authorship attribution, e.g., (Iqbal et al., 2008, 2013), but for the cross-topic setting which is considered harder and more-realistic character-level and word-level n-grams continued to be used as the base for state-of-the-art authorship attribution methods (Markov et al., 2017b; Stamatatos, 2017; Barlas and Stamatatos, 2020, 2021). The exclusion of stylometric features seemed counter-intuitive, given that by nature stylometric features such as average word length, the ratio of spaces to characters, or punctuation count naturally ignore the topic of the document compared to word- and character-level n-grams. The cross-topic setting, proposed by Stamatatos (2013), is created to show that existing authorship attribution techniques depend on the topic cues in a document to determine the author of a test document. This setup, however, does not allow researchers to analyze the effect of the topic on the attribution techniques.

In Chapter 4, I proposed the topic confusion task as a benchmark to measure the effect of the topic on different authorship identification techniques. Specifically, I use the topic as a confounding factor and probe the error that an attribution technique makes by dividing the error into one caused by the topic switch and one caused by the model's inability to capture the style difference. While the cross-topic setting can be used to rank techniques based on their performance and use that ranking to explain the effect of the topic on the compared techniques, this setting does not provide any insights when two models exhibit similar performance. In contrast, using the error

breakdown in the proposed topic-confusion benchmark I was able to show that despite achieving similar performance on the cross-topic scenario a model using POS n-grams with an SVM classifier and a model using BERT embeddings to create authors' profiles have different susceptibility to the topic variation.

Furthermore, I was able to show that stylometric features with an SVM classifier have lower confusion over the topic and while they do not achieve higher overall performance on the cross-topic scenario compared to other techniques, this low performance is not because of the topic effect, but rather because they are generally unable to capture the differences in authors writing styles.

I verified the outcomes of this task on the cross-topic authorship attribution scenario. The results showed that a simple linear classifier with stylometric features and POS tags could improve the authorship attribution performance compared to the commonly used n-grams. This approach achieved a new state-of-the-art of 83.3% on the cross-topic scenario, which is 3% over the previous state-of-the-art, masking-based, character-level approach. Surprisingly, neurally inspired techniques did not perform well on the authorship attribution task. Instead, they were outperformed by a simple, hand-crafted set of stylometric features and POS-level n-grams and an SVM classifier with a linear kernel. One potential reason is that stylometric features and POS-level n-grams are designed to capture non-content aspects in addition to working well with limited data. In contrast, neural approaches either require many training samples, or focus only on the content.

6.4 Using Pretrained Language Models for Authorship Identification

Pretrained language models have been shown to perform exceptionally well on various NLP tasks. For some tasks, e.g., sentiment analysis and topic classification, these models have been shown to work in a zero-shot setting, i.e., without any fine-tuning

for the end task. One reason for this success, however, is attributed to the large corpora that are used for the training process.

In Chapter 4, I compared two approaches to using these pretrained language models for authorship attribution. The first approach is to use one language model to generate word embeddings for a certain document, then apply a classifier over these embeddings to predict the author.

For the second approach, which is the more common approach to using language models for authorship attribution, I replicated the work of Barlas and Stamatatos (2021) where I used one pretrained language model per author fine-tuned only on that author's documents. Overall, this approach achieved much better performance than the first approach. Still, I argue that pretrained language models are not suitable for the authorship attribution task and I present my argument below.

I divide my argument into two parts to cover each approach separately. To begin with, I go back to the motivation for learning distributed word representation to replace the one-hot encoding. The problem with one-hot encoding is that it does not capture any similarity between words that may appear within the same context, or at least generally related. For example, in various tasks, one would want the similarity between the representation of *apple* and *orange* to be more than the similarity between each word and *chair* given that 'apple' and 'orange' are both fruits while 'chair' is not. Here, the measure of similarity is based on whether two words appear in the same context or not.

In authorship attribution, however, the requirement for words to have similar embeddings is whether they are used by the author or not, i.e., whether they can be found in that author's documents. For example, judging by only the language rules and the context, both "apples" and "oranges" are candidate words to finish the sentence "I like to eat <BLANK>". If, however, the sentence is used in the context of authorship attribution, then there is an additional requirement which is whether that particular author

likes apples or oranges. In contrast, one-hot encoding captures that exact requirement, and word frequencies can act as a weight for each word, where a higher frequency indicates how many times a word appears in the author's document.

The other approach to using pretrained language models is fairly old but more suitable for the authorship attribution task, that is using one language model per author. Indeed, the intuition behind this approach is that the language rules would change based on the author's writing preferences. Traditionally, language models for authorship attribution were trained from scratch, and for different tokenization levels. For example, Peng et al. (2003) trained a language model for character-level n-grams.

Lately, Barlas and Stamatatos (2020; 2021) fine-tuned BERT and RoBERTa models for the authorship attribution task. While this approach is more appropriate for the authorship attribution task than using one language model for all the authors, the results showed that the performance of this approach was worse than using word-level n-grams because such models are considerably more affected by the topics in the training samples.

6.5 Text Generation for Authorship Obfuscation

In an attempt to gain more insights about writing style features, I turned to the authorship obfuscation task where the goal is to hide the author's writing style to protect their identity. Naturally, the first step is to evaluate the state of the field and probe any issues with state-of-the-art obfuscation techniques.

The evaluation is commonly done for two dimensions: evasion and content preservation. Evasion describes the success of obfuscation techniques in hiding the author's writing style which is demonstrated by the author evading detection by authorship identification techniques. Content preservation, on the other hand, ensures that the message that the author intended to convey in the original document did not change

as a side-effect of the obfuscation.

While reviewing the literature, I found that existing content-preservation tools that are being used to evaluate obfuscation techniques are not up-to-date with state-of-the-art content-preservation tools that are used for other NLP tasks such as text summarization. Particularly, question-answering-based approaches have been deployed to replace token-based and model-based approaches to evaluate content preservation. In addition to having a better correlation with the human evaluation of content preservation, question-answering can provide a more intuitive way for users of obfuscation techniques to identify how is the content not preserved.

In addition to content preservation, my initial experiments showed that the evasion performance of existing obfuscation techniques is overstated when compared with a back-translation-based obfuscation technique. While using back translation is not a new approach, most of the existing work still cites the results from early papers on obfuscation where the translation model was based on outdated statistical machine translation tools. Nowadays, neural machine translation approaches are comparable to human-level performance. Upon re-investigating the performance of a back-translation model and comparing it to existing obfuscation techniques I showed that such a baseline outperforms these techniques in both evasion and content preservation while not requiring any training or fine-tuning.

To explain this result, I would like to recall the theoretical motivation for the topic confusion task discussed in Section 4.3. Specifically, I used Figure 4.2 to describe the relationship between the topic, the author's writing style, and the language when a document is written. When such language models are trained on a vast amount of documents that are on very many topics, one might argue that these documents cover a huge spectrum of writing styles as well. Because of that, when a text generation model is used to generate a translation, the style of the generated text is an aggregation or an average, of different styles.

To support this claim, we can look at the entropy of the probability distribution over the authors before, and after obfuscation. This is the same measure that is used to characterize misattribution harm in Section 5.3.5. The back-translation text-generation-based model resulted in a higher uncertainty, or entropy, compared to the other obfuscation techniques. This uncertainty can be interpreted as having the obfuscated document generally similar in style to the writing styles of all the other authors.

6.6 Limitations

One main limitation of this work is not investigating existing deep learning techniques for authorship identification and obfuscation. Such a question has been raised specifically by reviewers of the published work in this thesis. The argument that I presented in reply to that valid question is as follows.

At this point, it is a concrete fact that deep learning models that are trained from scratch require a tremendous amount of data to perform well. Researchers on authorship attribution have argued that the training data in real-life applications is very limited. They have also shown that adding more training data will increase the accuracy of authorship attribution techniques, but such performance would be overstated given that adding more samples is only possible in a lab setting, and not in a real-life scenario. On the other hand, I have provided empirical proof that zero-shot and few-shot neural models that work well for tasks such as topic classification and sentiment analysis perform poorly on the authorship attribution task.

Because of that, I decided to only cite deep learning-based models that have been evaluated on unrealistic datasets for authorship attribution while highlighting the issue of the unrealistic setting that was used to train them. An alternative approach would have been to train such models on the fairly small but realistic authorship attribution datasets and then claim that such models do not perform well on this task.

This, however, would be inaccurate and misleading.

Similarly, for the authorship obfuscation task, I evaluated neural methods that work out of the box, e.g., neural machine translation model, and only cite deep-learning-based approaches that require an unrealistically large training dataset.

6.7 Future Work

For the authorship attribution task, I have shown that pretrained language models have the highest susceptibility to capturing the topic of a document. I have also shown that the identity of the author can be a combination of their writing style and the topic bias in each document. Therefore, one potential future direction is to use these models to capture the topic of the document instead of using word-level features. In specific, it would be interesting to use these pretrained language models to learn a document embedding and then combine this embedding with stylometric features extracted from that document.

The inverse direction is using knowledge infusion techniques that are used to embed external knowledge in pretrained language models to teach these models structural information about the document, such as the number of paragraphs, the ratio of sentences to paragraphs, and other stylometric features. The advantage of this approach compared to simply combining stylometric features with pretrained embeddings could be beneficial in other domains where the structure of the document is important.

In Chapter 5, I have shown that the authorship obfuscation task is not up-to-date to the advancement in authorship identification. For example, none of the existing obfuscation algorithms were evaluated on cross-topic datasets. This is important because, as discussed in Chapter 4, the cross-topic scenario is the more realistic one, and evaluating authorship obfuscation techniques on a cross-topic dataset will give a better

indication of their performance.

Finally, in Chapter 3, I discussed the importance of explaining the outcome of authorship attribution techniques for the non-technical audience. Similar to evaluation tasks, proposing visualization and interpretability techniques for authorship attribution is an under-investigated area. In contrast, interpretability has received much attention within the deep learning community. Exploring such techniques for authorship attribution, e.g., using transfer learning, is one area of interest. More specifically, knowledge distillation techniques where a large, complex model can be used as a *teacher* for a smaller, more interpretable model.

Chapter 7

Conclusion and Summary

In this thesis, I presented an argument for evaluation techniques and why such techniques can help researchers reveal weaknesses of existing systems and then address these weaknesses to achieve better performance.

For the task of authorship attribution, I showed that an author's *writeprint* in a document is a combination of the author's writing style, captured by a combination of stylometric features and POS tags, and the author's topic bias. For the authorship obfuscation task, I showed that the performance of existing obfuscation techniques is overstated where an off-the-shelf baseline performs as well as these techniques and has better content preservation.

Below, I highlight the contributions of this thesis for the tasks of authorship attribution and authorship obfuscation.

Summary of Contributions

For the task of authorship identification, I proposed two new datasets for the task of authorship attribution. The first dataset is on Arabic tweets and was used to evaluate authorship attribution techniques that were developed for English on another

language. The second data set was used for the topic-confusion benchmark.

Furthermore, I proposed an evaluation task, namely, topic confusion, and proposed two measures to characterize the errors made by attribution techniques on this task to help understand the effect of the topic on the attribution process. Finally, I highlighted the importance of using stylometric features and POS tags in capturing a writing style that is less susceptible to topic variations in the author's writing samples and showed that pretrained language models have a key weakness when used as a writing style representation.

For the task of authorship obfuscation, I showed that existing obfuscation tools have critical weaknesses when evaluated for obfuscation effectiveness and content preservation, raised the issue of misattribution, which is a result of successful obfuscation of a document, and propose an information-theoretic measure to characterize it for different models. Finally, I showed that a baseline that is based on pretrained language models has a high potential as an obfuscation tool given that pretrained language models have a generic writing style. This is in line with the findings from the topic confusion task where I show that pretrained-language-model-based authorship attribution techniques perform poorly.

Appendix A

A.1 The Top 100 Most Frequent Function-Word Features

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

A.2 Function-Word Features With Zero Usage

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

A.3 Statistical Results

Table A.1 Using the ANOVA test to showed that the difference between the datasets a, b, c, d, and e is statistically significant.

(a) 2 Authors

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

(b) 5 Authors

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

(c) 10 Authors

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

(d) 20 Authors

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

Table A.2 Post hoc Tukey HSD test with $\alpha = 0.05$ to evaluate the effect of increasing the number of authors on the performance (ANOVA result: $[F(3, 12) = 249.57, p < 0.001]$)

(a) Mean and SD

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

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

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

Table A.3 Paired two-sample t-Tests with $\alpha = 0.05$ to evaluate the significance of the difference in the performance of n-grams vs. NB, SVM, DT ,and RF $[F(4, 12) = 19.37, p < 0.001]$

(a) Mean and SD

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

(b) t-Test Results and p-value

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

Table A.4 Mean and SD for 2, 5, 10, and 20 authors with varying number of tweets per author.

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

Table A.5 ANOVA and Post hoc Tukey tests (with $\alpha = 0.05$) to evaluate the significance of the difference in the performance of n-grams vs. NB, SVM, DT, and RF for two (a), five (b), 10 (c), and 20 authors (d)

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

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

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

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

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

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

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

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

Table A.6 Mean and SD for 2, 5, 10, and 20 authors with specifying the minimum number of words per tweet.

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

Table A.7 ANOVA and Post hoc Tukey tests (with $\alpha = 0.05$) to evaluate the significance of the difference in the performance of n-grams vs. NB, SVM, DT and RF

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

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

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

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

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

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

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

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

Table A.8 Paired two-sample t-Tests with $\alpha = 0.05$ to evaluate the significance of the difference in the performance of NB, SVM, DT, RF, and n-grams when groups of 5 tweets are merged into one artificial tweet.

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

Table A.9 Post hoc Tukey HSD test with $\alpha = 0.05$ to evaluate the effect of merging groups of 5 tweets into one artificial tweet on the performance (ANOVA result: $[F(4, 15) = 5.30, p = 0.007]$)

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

Table A.10 ANOVA t-Test with $\alpha = 0.05$ to evaluate the different n-grams modalities.

(a) Mean and SD

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

(b) ANOVA results and P values

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

Table A.11 Post-hoc Tukey test (with $\alpha = 0.05$) to evaluate the significance of the difference in using character level, word level, POS level or all modalities combined.

(a) 2 Authors

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

(b) 5 Authors

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

(c) 10 Authors

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

(d) 20 Authors

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

Table A.12 Mean and SD for using Diacritics with different n-gram modalities

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

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

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

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

Table A.13 ANOVA t-Test with $\alpha = 0.05$ to evaluate the effect of using Diacrticis with different n-gram modalities.

(a) ANOVA results and p-values

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

Table A.14 Mean and SD for using different categories of features.

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

Table A.15 5 Authors: Post hoc Tukey test to evaluate different categories of features.

(a) ANOVA results and P values

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

Table A.16 10 Authors: Post hoc Tukey test to evaluate different categories of features.

(a) ANOVA results and p-values

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

Table A.17 The EBG dataset: Identification accuracy (left) and Misattribution harm (right) characterized by raw entropy scores.

Anon. T.	Identification accuracy (%)		Entropy	
	5 Authors	10 Authors	5 Authors	10 Authors
None	96.4	77.6	1.72 ± 0.1	2.77 ± 0.1
A*	93.5	71.1	1.83 ± 0.1	2.88 ± 0.1
Back Translation	84.0	64.2	1.91 ± 0.1	2.95 ± 0.1
Lexical Sub (BERT)	91.5	78.4	1.87 ± 0.1	2.81 ± 0.1
Mutant-X	86.4	73.6	1.69 ± 0.1	2.83 ± 0.1

Table A.18 The C50 dataset: Identification accuracy (left) and Misattribution harm (right) characterized by raw entropy scores.

Anon. T.	Identification accuracy (%)		Entropy	
	5 Authors	10 Authors	5 Authors	10 Authors
None	76.0	67.3	1.84 ± 0.2	2.80 ± 0.1
A*	72.0	64.0	1.83 ± 0.1	2.81 ± 0.1
Back Translation	73.3	65.3	1.93 ± 0.1	2.90 ± 0.1
Lexical Sub (BERT)	76.0	67.3	1.86 ± 0.2	2.84 ± 0.1
Mutant-X	74.7	66.7	1.90 ± 0.1	2.76 ± 0.1

Bibliography

Ahmed Abbasi and Hsinchun Chen. 2005a. Applying authorship analysis to arabic web content. *Intelligence and Security Informatics*, pages 75–93.

Ahmed Abbasi and Hsinchun Chen. 2005b. Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 20(5):67–75.

Ahmed Abbasi and Hsinchun Chen. 2006. Visualizing authorship for identification. In *International Conference on Intelligence and Security Informatics*, pages 60–71. Springer.

Ahmed Abbasi and Hsinchun Chen. 2008. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems (TOIS)*, 26(2):1–29.

Sadia Afroz, Michael Brennan, and Rachel Greenstadt. 2012. Detecting hoaxes, frauds, and deception in writing style online. In *2012 IEEE Symposium on Security and Privacy*, pages 461–475. IEEE.

Sadia Afroz, Aylin Caliskan Islam, Ariel Stolerman, Rachel Greenstadt, and Damon McCoy. 2014. Doppelgänger finder: Taking stylometry to the underground. In *IEEE Symposium on Security and Privacy*, pages 212–226. IEEE.

Mahmoud Al-Ayyoub, Ahmed Alwajeeh, and Ismail Hmeidi. 2017a. An extensive study of authorship authentication of arabic articles. *International Journal of Web Information Systems*, 13(1):85–104.

Mahmoud Al-Ayyoub, Yaser Jararweh, Abdullateef Rabab'ah, and Monther Aldwairi. 2017b. Feature extraction and selection for arabic tweets authorship authentication. *Journal of Ambient Intelligence and Humanized Computing*, 8(3):383–393.

Mashael Alamr. 2022. *Authorship Attribution, Idiolectal Style, and Online Identity: a specialised corpus of Najdi Arabic tweets*. Ph.D. thesis, University of Leeds.

Fatimah Alqahtani and Mischa Dohler. 2022. Survey of authorship identification tasks on arabic texts. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* Just Accepted.

Malik H. Altakrori, Jackie Chi Kit Cheung, and Benjamin C. M. Fung. 2021. The topic confusion task: A novel evaluation scenario for authorship attribution. In *Findings of the 2021 Conf. on Empirical Methods in Natural Language Processing: EMNLP 2021*, pages 4242–4256, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Malik H. Altakrori, Farkhund Iqbal, Benjamin C. M. Fung, Steven H. H. Ding, and Abdallah Tubaishat. 2018. Arabic authorship attribution: An extensive study on twitter posts. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(1):5.1–5.51.

Malik H. Altakrori, Thomas Scialom, Benjamin C.M. Fung, and Jackie Chi Kit Cheung. 2022. A multifaceted framework to evaluate evasion, content preservation, and misattribution in authorship obfuscation techniques. *"Under Review" in Proc. of the 2022 Conf. on Empirical Methods in Natural Language Processing: EMNLP 2022*, pages 1–8.

Alaa Saleh Altheneyan and Mohamed El Bachir Menai. 2014. Naïve bayes classifiers for authorship attribution of arabic texts. *Journal of King Saud University-Computer and Information Sciences*, 26(4):473–484.

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

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

ArabiNames.com. 2015. Arabi names. Available at `http://arabinames.com/categories.aspx` (2015/06).

Douglas Bagnall. 2015. Author identification using multi-headed recurrent neural networks. In *Working Notes of Conference and Labs of the Evaluation Forum (CLEF)*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Georgios Barlas and Efstathios Stamatatos. 2020. Cross-domain authorship attribution using pre-trained language models. In *Artificial Intelligence Applications and Innovations*, pages 255–266, Cham. Springer International Publishing.

Georgios Barlas and Efstathios Stamatatos. 2021. A transfer learning approach to cross-domain authorship attribution. *Evolving Systems*, pages 1–19.

Victor Benjamin, Wingyan Chung, Ahmed Abbasi, Joshua Chuang, Catherine A Larson, and Hsinchun Chen. 2014. Evaluating text visualization for authorship analysis. *Security Informatics*, 3(1):10.

Janek Bevendorff, Martin Potthast, Matthias Hagen, and Benno Stein. 2019. Heuristic authorship obfuscation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1098–1108, Florence, Italy. Association for Computational Linguistics.

Janek Bevendorff, Tobias Wenzel, Martin Potthast, Matthias Hagen, and Benno Stein. 2020. On divergence-based author obfuscation: An attack on the state of the art in statistical authorship verification. *it-Information Technology*, 62(2):99–115.

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

Christopher M Bishop. 2006. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer.

David M Blei. 2012. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

Haohan Bo, Steven H. H. Ding, Benjamin C. M. Fung, and Farkhund Iqbal. 2021. ER-AE: Differentially private text generation for authorship anonymization. In *Proc. of the 2021 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3997–4007, Online. Association for Computational Linguistics.

Dasha Bogdanova and Angeliki Lazaridou. 2014. Cross-language authorship attribution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2015–2020, Reykjavik, Iceland. European Language Resources Association (ELRA).

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

Sarah R Boutwell. 2011. Authorship attribution of short messages using multimodal features. Technical report, DTIC Document.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Michael Brennan, Sadia Afroz, and Rachel Greenstadt. 2012. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Transactions on Information and System Security (TISSEC)*, 15(3):1–22.

John F Burrows. 1992. Not unles you ask nicely: The interpretative nexus between analysis and information. *Literary and Linguistic Computing*, 7(2):91–109.

Aylin Caliskan-Islam, Richard Harang, Andrew Liu, Arvind Narayanan, Clare Voss, Fabian Yamaguchi, and Rachel Greenstadt. 2015. De-anonymizing programmers via code stylometry. In *24th USENIX security symposium (USENIX Security 15)*, pages 255–270.

Yixin Cao, Ruihao Shui, Liangming Pan, Min-Yen Kan, Zhiyuan Liu, and Tat-Seng Chua. 2020. Expertise style transfer: A new task towards better communication between experts and laymen. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1061–1071.

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

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. USE: Universal sentence encoder for english. In *Proc. of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 169–174.

Carole E Chaski. 2005. Who's at the keyboard? authorship attribution in digital evidence investigations. *International Journal of Digital Evidence*, 4(1):1–13.

Na Cheng, Rajarathnam Chandramouli, and KP Subbalakshmi. 2011. Author gender identification from text. *Digital Investigation*, 8(1):78–88.

José Eleandro Custódio and Ivandré Paraboni. 2019. An ensemble approach to cross-domain authorship attribution. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 201–212. Springer.

Sophia Elizabeth De Morgan and Augustus De Morgan. 1882. *Memoir of Augustus De Morgan*. Longmans, Green, and Company.

Gülşen Demiröz and H Altay Güvenir. 1997. *Classification by Voting Feature Intervals*, pages 85–92. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Steven H. H. Ding, Benjamin C. M. Fung, and Mourad Debbabi. 2015. A visualizable evidence-driven approach for authorship attribution. *ACM Trans. Inf. Syst. Secur.*, 17(3):12:1–12:30.

Steven H. H. Ding, Benjamin C. M. Fung, Farkhund Iqbal, and William K. Cheung. 2019. Learning stylometric representations for authorship analysis. *IEEE Transactions on Cybernetics*, 49(1):107–121.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

Chris Emmery, Enrique Manjavacas Arevalo, and Grzegorz Chrupała. 2018. Style obfuscation by invariance. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 984–996, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Georgia Frantzeskou, Efstathios Stamatatos, Stefanos Gritzalis, Carole E Chaski, and Blake Stephen Howald. 2007. Identifying authorship by byte-level n-grams: The source code author profile (SCAP) method. *International Journal of Digital Evidence*, 6(1):1–18.

Zhenhao Ge, Yufang Sun, and Mark J. T. Smith. 2016. Authorship attribution using a neural network language model. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 4212–4213. AAAI Press.

Jade Goldstein-Stewart, Ransom Winder, and Roberta Sabin. 2009. Person identification from text and speech genre samples. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 336–344, Athens, Greece. Association for Computational Linguistics.

Helena Gómez-Adorno, Juan-Pablo Posadas-Durán, Grigori Sidorov, and David Pinto. 2018. Document embeddings learned on various types of n-grams for cross-topic authorship attribution. *Computing*, 100(7):741–756.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Ana Granados, David Camacho, and Francisco Borja Rodríguez. 2012. Is the contextual information relevant in text clustering by compression? *Expert Systems with Applications*, 39(10):8537–8546.

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

MA Hall. 1998. Correlation-based feature subset selection for machine learning. *Thesis submitted in partial fulfillment of the requirements of the degree of Doctor of Philosophy at the University of Waikato*.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: An update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Jiawei Han and Micheline Kamber. 2001. Data mining: Concepts and techniques. *San Francisco, CA, itd: Morgan Kaufmann*, 5.

Julian Hitschler, Esther van den Berg, and Ines Rehbein. 2017. Authorship attribution with convolutional neural networks and POS-eliding. In *Proceedings of the Workshop on Stylistic Variation*, pages 53–58, Copenhagen, Denmark. Association for Computational Linguistics.

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

David I Holmes. 1998. The evolution of stylometry in humanities scholarship. *Literary and linguistic computing*, 13(3):111–117.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7856–7870, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Houvardas and Efstathios Stamatatos. 2006. N-gram feature selection for authorship identification. In *International conference on artificial intelligence: Methodology, systems, and applications*, pages 77–86. Springer.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

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

Farkhund Iqbal, Hamad Binsalleeh, Benjamin C. M. Fung, and Mourad Debbabi. 2010a. Mining writeprints from anonymous e-mails for forensic investigation. *Digital Investigation*, 7(1-2):56–64.

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

Farkhund Iqbal, Mourad Debbabi, and Benjamin C. M. Fung. 2020. *Machine Learning for Authorship Attribution and Cyber Forensics*. Computer Entertainment and Media Technology. Springer Nature.

Farkhund Iqbal, Rachid Hadjidj, Benjamin C. M. Fung, and Mourad Debbabi. 2008. A novel approach of mining write-prints for authorship attribution in e-mail forensics. *Digital Investigation*, 5:S42–S51.

Farkhund Iqbal, Liaquat A. Khan, Benjamin C. M. Fung, and Mourad Debbabi. 2010b. E-mail authorship verification for forensic investigation. In *Proc. of the 25th ACM SIGAPP Symposium on Applied Computing (SAC)*, pages 1591–1598, Sierre, Switzerland. ACM Press.

Shunichi Ishihara. 2011. A forensic authorship classification in SMS messages: A likelihood ratio based approach using n-gram. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 47–56, Canberra, Australia.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*, volume 112. Springer.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19.

Patrick Juola. 2006. Authorship attribution for electronic documents. In *IFIP International Conference on Digital Forensics*, pages 119–130. Springer.

Patrick Juola. 2007. Future trends in authorship attribution. In *IFIP International Conference on Digital Forensics*, pages 119–132. Springer.

Patrick Juola. 2008. *Authorship Attribution*, volume 1. Now Publishers, Inc.

Gary Kacmarcik and Michael Gamon. 2006. Obfuscating document stylometry to preserve author anonymity. In *Proc. of the Int'l Conference on Computational Linguistics (COLING)*, pages 444–451, Sydney, Australia. Association for Computational Linguistics.

Vlado Kešelj, Fuchun Peng, Nick Cercone, and Calvin Thomas. 2003. N-gram-based author profiles for authorship attribution. In *Proc. of the conference pacific association for computational linguistics, PACLING*, volume 3, pages 255–264.

Yashwant Keswani, Harsh Trivedi, Parth Mehta, and Prasenjit Majumder. 2016. Author masking through translation. In *CLEF (Working Notes)*, pages 890–894.

Dmitry V Khmelev. 2000. Disputed authorship resolution through using relative empirical entropy for markov chains of letters in human language texts. *Journal of quantitative linguistics*, 7(3):201–207.

Foaad Khosmood. 2012. Comparison of sentence-level paraphrasing approaches for statistical style transformation. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer ....

Foaad Khosmood and Robert Levinson. 2010. Automatic synonym and phrase replacement show promise for style transformation. In *2010 Ninth International Conference on Machine Learning and Applications*, pages 958–961. IEEE.

Bradley Kjell, W Addison Woods, and Ophir Frieder. 1994. Discrimination of authorship using visualization. *Information Processing & Management*, 30(1):141–150.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Moshe Koppel, Shlomo Argamon, and Anat Rachel Shimoni. 2002. Automatically categorizing written texts by author gender. *Literary and linguistic computing*, 17(4):401–412.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology*, 60(1):9–26.

Olga V Kukushkina, Anatoly A Polikarpov, and Dmitry V Khmelev. 2001. Using literal and grammatical statistics for authorship attribution. *Problems of Information Transmission*, 37(2):172–184.

Sushil Kumar and Mousmi A Chaurasia. 2012. Assessment on stylometry for multilingual manuscript. *Assessment*, 2(9):01–06.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Robert Layton, Stephen McCombie, and Paul Watters. 2012a. Authorship attribution of irc messages using inverse author frequency. In *Proc. of the 3rd Cybercrime and Trustworthy Computing Workshop (CTC)*, pages 7–13.

Robert Layton, Paul Watters, and Richard Dazeley. 2010. Authorship attribution for twitter in 140 characters or less. In *Proc. of the Second Cybercrime and Trustworthy Computing Workshop (CTC)*, pages 1–8. IEEE.

Robert Layton, Paul Watters, and Richard Dazeley. 2012b. Recentred local profiles for authorship attribution. *Natural Language Engineering*, 18(03):293–312.

Quoc V. Le and Tomás Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1188–1196. JMLR.org.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Robert A Leonard, Juliane ER Ford, and Tanya Karoli Christensen. 2016. Forensic linguistics: Applying the science of linguistics to issues of the law. *Hofstra L. Rev.*, 45:881.

Jiexun Li, Rong Zheng, and Hsinchun Chen. 2006. From fingerprint to writeprint. *Communications of the ACM*, 49(4):76–82.

Mark Liberman. 2008. Ask language log: Comparing the vocabularies of different languages. Online.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv*, pages arXiv–1907.

Kim Luyckx and Walter Daelemans. 2008. Authorship attribution and verification with many authors and limited data. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 513–520, Manchester, UK. Coling 2008 Organizing Committee.

Kim Luyckx and Walter Daelemans. 2011. The effect of author set size and data size in authorship attribution. *Literary and linguistic Computing*, 26(1):35–55.

Asad Mahmood, Faizan Ahmad, Zubair Shafiq, Padmini Srinivasan, and Fareed Zaffar. 2019. A girl has no name: Automated authorship obfuscation using mutant-x. *Proceedings on Privacy Enhancing Technologies*, 2019(4):54–71.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014a. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014b. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Muharram Mansoorizadeh, Taher Rahgooy, Mohammad Aminiyan, and Mahdy Eskandari. 2016. Author obfuscation using wordnet and language models—notebook for pan at clef 2016. In *CLEF 2016 Evaluation Labs and Workshop–Working Notes Papers*, pages 5–8.

Ilia Markov, Jorge Baptista, and Obdulia Pichardo-Lagunas. 2017a. Authorship attribution in portuguese using character n-grams. *Acta Polytechnica Hungarica*, 14(3):59–78.

Ilia Markov, Efstathios Stamatatos, and Grigori Sidorov. 2017b. Improving cross-topic authorship attribution: The role of pre-processing. In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 289–302. Springer.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Andrew W. E. McDonald, Sadia Afroz, Aylin Caliskan, Ariel Stolerman, and Rachel Greenstadt. 2012. Use fewer instances of the letter "i": Toward writing style anonymization. In *Privacy Enhancing Technologies*.

Andrew WE McDonald, Jeffrey Ulman, Marc Barrowclift, and Rachel Greenstadt. 2013. Anonymouth revamped: Getting closer to stylometric anonymity. In *PETools: Workshop on Privacy Enhancing Tools*, volume 20.

Thomas Corwin Mendenhall. 1887. The characteristic curves of composition. *Science*, 9(214):237–249.

George K. Mikros and Eleni K. Argiri. 2007. Investigating topic influence in authorship attribution. In *Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection (PAN 07)*. CEUR-WS.org.

Miniwatts Marketing Group. 2013. Internet world users by language.

Frederick Mosteller and David L Wallace. 1963. Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed federalist papers. *Journal of the American Statistical Association*, 58(302):275–309.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.

Abdulfattah Omar and Wafya Ibrahim Hamouda. 2020. The effectiveness of stemming in the stylometric authorship attribution in arabic. *International Journal of Advanced Computer Science and Applications*, 11(1).

James O'Shea. 2013. Alphabetical order 277 word new function word list. https://semanticsimilarity.files.wordpress.com/2013/08/jim-oshea-fwlist-277.pdf. [Retrieved Oct. 2019].

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

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

Rebekah Overdorf and Rachel Greenstadt. 2016. Blogs, twitter feeds, and reddit comments: Cross-domain authorship attribution. *Proc. on Privacy Enhancing Technologies*, 2016(3):155–171.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101, Reykjavik, Iceland. European Language Resources Association (ELRA).

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Fuchun Peng, Dale Schuurmans, Shaojun Wang, and Vlado Kešelj. 2003. Language independent authorship attribution using character level language models. In *Proc. of 10th Conference of the European Chapter, Association for Computational Linguistics (EACL)*, EACL '03, page 267–274, USA. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Juan-Pablo Posadas-Durán, Helena Gómez-Adorno, Grigori Sidorov, Ildar Batyrshin, David Pinto, and Liliana Chanona-Hernández. 2017. Application of the distributed document representation in the authorship attribution task for small corpora. *Soft Computing*, 21(3):627–639.

Martin Potthast, Matthias Hagen, and Benno Stein. 2016. Author obfuscation: Attacking the state of the art in authorship verification. In *CLEF (Working Notes)*, pages 716–749.

John Ross Quinlan. 1993. C4.5: Programs for machine learning. *The Morgan Kaufmann Series in Machine Learning, San Mateo, CA: Morgan Kaufmann,| c1993*, 1.

Abdullateef Rabab'ah, Mahmoud Al-Ayyoub, Yaser Jararweh, and Monther Aldwairi. 2016. Authorship attribution of arabic tweets. *Proc. of the IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–6.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Roshan Ragel, Pramod Herath, and Upul Senanayake. 2013. Authorship detection of sms messages using unigrams. In *Proc. of the 8th International Conference on Industrial and Information Systems (ICIIS)*, pages 387–392. IEEE.

Josyula R. Rao and Pankaj Rohatgi. 2000. Can pseudonymity really guarantee privacy? In *USENIX Security Symposium*.

Anderson Rocha, Walter J Scheirer, Christopher W Forstall, Thiago Cavalcante, Antonio Theophilo, Bingyu Shen, Ariadne RB Carvalho, and Efstathios Stamatatos. 2016. Authorship attribution for social media forensics. *IEEE Transactions on Information Forensics and Security*, 12(1):5–33.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *Proc. of the 20th Conf. on Uncertainty in Artificial Intelligence*, UAI '04, page 487–494, Arlington, Virginia, USA. AUAI Press.

David C Rubin. 1978. Word-initial and word-final ngram frequencies. *Journal of Literacy Research*, 10(2):171–183.

Sebastian Ruder, Parsa Ghaffari, and John G Breslin. 2016. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *ArXiv preprint*, abs/1609.06686.

Hataichanok Saevanee, Nathan Clarke, Steven Furnell, and Valerio Biscione. 2015. Continuous user authentication using multi-modal biometrics. *Computers & Security*, 53:234–246.

Miguel A Sanchez-Perez, Ilia Markov, Helena Gómez-Adorno, and Grigori Sidorov. 2017. Comparison of character n-grams and lexical features on author, gender, and language variety identification on the same spanish news corpus. In *Proc. of the Int'l Conf. of the Cross-Language Evaluation Forum for European Languages*, pages 145–151. Springer.

Upendra Sapkota, Steven Bethard, Manuel Montes, and Thamar Solorio. 2015. Not all character n-grams are created equal: A study in authorship attribution. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–102, Denver, Colorado. Association for Computational Linguistics.

Upendra Sapkota, Thamar Solorio, Manuel Montes, Steven Bethard, and Paolo Rosso. 2014. Cross-topic authorship attribution: Will out-of-topic data help? In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1228–1237, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Yunita Sari, Andreas Vlachos, and Mark Stevenson. 2017. Continuous n-gram representations for authorship attribution. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 267–273, Valencia, Spain. Association for Computational Linguistics.

Michael Schmid, Farkhund Iqbal, and Benjamin C. M. Fung. 2015. E-mail authorship attribution using customized associative classification. *Digital Investigation (DIIN)*, 14(1):S116–S126.

Tal Schuster, Roei Schuster, Darsh J Shah, and Regina Barzilay. 2020. The limitations of stylometry for detecting machine-generated fake news. *Computational Linguistics*, 46(2):499–510.

Roy Schwartz, Oren Tsur, Ari Rappoport, and Moshe Koppel. 2013. Authorship attribution of micro-messages. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1880–1891, Seattle, Washington, USA. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2014. Authorship attribution with topic models. *Computational Linguistics*, 40(2):269–310.

Kareem Shaker and David Corne. 2010. Authorship attribution in arabic using a hybrid of evolutionary search and linear discriminant analysis. In *Proc. of The Computational Intelligence (UKCI) Workshop*, pages 1–6, UK. IEEE.

Kareem Shaker, David Corne, and Richard Everson. 2007. Investigating hybrids of evolutionary search and linear discriminant analysis for authorship attribution. In *Proc. of the IEEE Congress on Evolutionary Computation (CEC)*, pages 2071–2077. IEEE.

Rakshith Shetty, Bernt Schiele, and Mario Fritz. 2018. A4NT: Author attribute anonymity by adversarial training of neural machine translation. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1633–1650, Baltimore, MD. USENIX Association.

Prasha Shrestha, Sebastian Sierra, Fabio González, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 669–674, Valencia, Spain. Association for Computational Linguistics.

Rui Sousa Silva, Gustavo Laboreiro, Luís Sarmento, Tim Grant, Eugénio Oliveira, and Belinda Maia. 2011. 'twazn me!!!;('automatic authorship analysis of micro-blogging messages. In *Natural Language Processing and Information Systems*, pages 161–168. Springer.

Steve Simon. 2005. When the f test is significant, but tukey is not. Available at `http://www.pmean.com/05/TukeyTest.html` (2005-09-09).

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556.

Efstathios Stamatatos. 2013. On the robustness of authorship attribution based on character n-gram features. *Journal of Law and Policy*, 21:421–439.

Efstathios Stamatatos. 2017. Authorship attribution using text distortion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1138–1149, Valencia, Spain. Association for Computational Linguistics.

Efstathios Stamatatos. 2018. Masking topic-related information to enhance authorship attribution. *Journal of the Association for Information Science and Technology*, 69(3):461–473.

Kalaivani Sundararajan and Damon Woodard. 2018. What represents "style" in authorship attribution? In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2814–2822, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nick Taylor. 2017. Twitter and open data in academia. Available at https://twittercommunity.com/t/twitter-and-open-data-in-academia/51934 (2017-07-20).

William John Teahan. 2000. Text classification and segmentation using minimum cross-entropy. In *Content-Based Multimedia Information Access - Volume 2*, RIAO '00, page 943–961. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, Paris, FRA.

Twitter. 2017. Developer agreement and policy. Available at https://dev.twitter.com/overview/terms/agreement-and-policy (2017-06-18).

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Olivier de Vel, Alison Anderson, Malcolm Corney, and George Mohay. 2001. Mining e-mail content for author identification forensics. *ACM Sigmod Record*, 30(4):55–64.

Haining Wang, Allen Riddell, and Patrick Juola. 2021. Mode effects' challenge to authorship attribution. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1146–1155, Online. Association for Computational Linguistics.

Benjamin Weggenmann and Florian Kerschbaum. 2018. Syntf: Synthetic and differentially private term frequency vectors for privacy-preserving text mining. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 305–314. ACM.

Ian H Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, third ed edition. Morgan Kaufmann.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv preprint*, abs/1609.08144.

Wei Xu, Alan Ritter, William B Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of COLING 2012*, pages 2899–2914.

G Udny Yule. 1939. On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship. *Biometrika*, 30(3/4):363–390.

Richong Zhang, Zhiyuan Hu, Hongyu Guo, and Yongyi Mao. 2018. Syntax encoding with application in authorship attribution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2742–2753, Brussels, Belgium. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. 2006. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American society for information science and technology*, 57(3):378–393.