Computational Stylometry in Adversarial Settings

Rémi de Zoeten

July 2015

University of Amsterdam,
Faculty of Science

For the degree of: Master of Science in Artificial Intelligence
Under the supervision of: Dr. Ilya Markov

Abstract

When computational stylometry, the study of style, is applied to typed natural language texts it becomes possible to identify authors based on the texts that they write. If the author has a preference for writing anonymously, then the use of computational stylometry is adversarial to the author. In turn, an author may behave in an adversarial way to an attributer, by writing in a way that hides her identity.

In this work, we investigate to what extent authors with a preference for anonymity are affected by the application of stylometry and how authorship may be attributed even when an author implements tactics to prevent authorship attribution.

In this work, we develop state-of-the-art authorship attribution methods for adversarial settings, using feature and language modelling based attribution methods. We show that authorship can be attributed accurately if the author does not take any precautions to prevent authorship attribution, and we investigate the effectiveness of adversarial authorship as a way to improve author anonymity. We develop a novel method capable of (partial) text de-obfuscation and demonstrate its effectiveness. We also show that imitation as an adversarial writing tactic is more effective against an adversarial authorship attribution attempt than obfuscation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Stylometry as investigated in this work

Stylometry is the study of style, usually applied to natural language texts but also to computer code [18] and possibly other mediums that can reflect the author's style. Historically, stylometry has been applied to handwritten texts, but our current work focuses on typed natural language texts. Stylometry is often used for answering the question "Who wrote this text?". This type of stylometry is specified as authorship attribution. However, there also exist other kinds of stylometric tasks including gender and age attribution. In this work we contribute to state-of-the-art authorship attribution.

Stylometry may help to reveal facts (like identity or age) about an author based solely on the texts that she writes. This conflicts with the interests of the author if she prefers not to reveal any information beyond the message that the text carries. If the author wishes to remain anonymous, then the stylometric method is applied in an adversarial setting and is considered an adversarial application of stylometry. Some organizations have an incentive to use adversarial stylometry in order to identify dissidents, and individuals may also be interested in the use of adversarial stylometry. **The goal of this research** is to create a better understanding of how organizations and individuals can use stylometry in an adversarial way, how authors are affected by this and how they might effectively defend against an adversarial application of stylometry. To this end, we will assume that the stylometric tasks that we investigate throughout this research take place in an adversarial setting.

An author can write in her natural writing style, or she might write differently in order to try to subvert stylometric analyses. In the context of stylometric analysis, such behavior is said to implement an attack, and we say the author is writing in an adversarial way. Analysis of these adversarial texts means performing stylometry in an adversarial setting. At least two types of adversarial writing tactics have been described in the literature [16]: the obfuscation attack and the imitation attack, which are described in Section 2.2. We analyse how effective the implementation of these tactics are. Then we develop adversarial stylometric methods that are designed specifically to counter these adversarial writing tactics.

1.2 Motivation

There are at least two sides to computational stylometry. On the one hand, there are organizations such as corporations and governments that have an incentive to identify individual authors that interfere with the interests of the organization. On the other hand, authors may want to escape a targeted reaction from those organizations.

1.2.1 Stylometry and government

In 2009 the FBI stated in their Technology Assessment for the State of the Art Biometrics Excellence Roadmap (SABER), “As non-handwritten communications become more prevalent, such as blogging, text messaging and emails, there is a growing need to identify writers not by their written script, but by analysis of the typed content. [12]” This shows that (American) law enforcement has an interest in applying stylometry to typed texts.

Courts of law have accepted stylometric evidence that has helped clear people of charges [2]. In one court case in the United States, stylometric evidence showed that it was improbable that the accused had written a particular confession.

In another court case, stylometry has been used to convict a man in 2009 for murdering his wife. The stylometric evidence in this case showed that the husband was more likely to have authored his wife’s suicide note than the wife herself [3].

1.2.2 Contemporary anonymous authors

Belle de Jour Belle de Jour is a pseudonym of the author of a blog called ‘Diary of a London Call Girl’. The blog describes the life of a prostitute in London in 2003. By her own initiative, Brooke Magnanti decided to step forward as the true author of the blog in November 2009 [5].

Employee Employee is a hypothetical author who wishes to write anonymously. Employee has decided to write down and leak information that is known only within the organization that she works for. Within the organization all employees submit work in the form of text to the organization. The organization could therefore decide to attempt to attribute the leak to the correct employee based on texts that it has collected from all its employees.

John Twelve Hawks The author John Twelve Hawks wrote a distopian triology and in 2014 the non-fiction book *Against Authority* [6]. In *Against Authority*, John Twelve Hawks writes the following passage about his choice to write anonymously:

> *For the first drafts of the book, I kept my birth name off the title page. The old me wasnt writing this book. Something was different. Something had changed. I had always admired George Orwell, and had read his collected essays and letters countless times. When Eric Blair became Orwell, he was set free, liberated from his Eton education and colonial policeman past. And there was another factor about the title page that troubled me. I was telling my readers that this new system of information technology was going to destroy our privacy, and that they should resist this change. It seemed hypocritical to go on a book tour or appear on a talk show blabbing about my life when our private lives were under attack.*

JRandom A person under the pseudonym ’JRandom’ was the principal author of the Invisible Internet Project (I2P) until she (or he) vanished in November 2008 [4, 7].

Mr Anonymous *Bourbon Kid* is a thriller series written by Mr Anonymous. The first part of the series was published in 2000. In an interview [9] Mr Anonymous said the following about his choice to remain anonymous (translated):

> *It was amusing to see if anyone would recognize me based on the text in the novel. And also to see if anyone would buy the book without knowing who the author is.*

Satoshi Nakamoto Satoshi Nakamoto is the creator of the bitcoin protocol and the first reference implementation in 2008 [1]. She (or he) wrote a white paper and other correspondence up until late 2010 before she (or he) stopped communicating [13]. At the time of the release of bitcoin, it was unclear if its creation was legal. As the first participant in the bitcoin network, Nakamoto is believed to be in control of about 1.000.000 bitcoin, which is $\frac{1}{21}$ of the total supply of bitcoins [13]. The 2014 price average of a bitcoin was 529USD. The question of legality as well as the wealth that Nakomoto is likely to have accumulated could have contributed to her (or his) decision to remain anonymous.

The secret social democrat An unidentified member of the Danish social democratic party has written a book called *Den Hemmelige Socialdemokrat*, in which she (or he) describes power struggles within the party and wrongdoings by party members while it was part of the Danish government [11]. Because the author is known to be an elected member of the parliament for the Danish social democratic party, there are not many possible authors.

1.2.3 Different reasons for authors to stay anonymous

Section 1.2.2 shows that there are contemporary authors who wish to remain anonymous. The reasons for their desire for anonymity are varied. Some authors (**Mr Anonymous**) may prefer anonymity for the experience of writing and publishing anonymously. Other authors (**Hawks, JRandom**) choose to write anonymously because it is in line with their political philosophy. Still other authors (**Jour, Employee, Nakamoto, Socialdemokrat**) can reasonably expect adverse reactions from the social environment they are in if their identity were revealed. These adverse reactions could be actions undertaken by the governments under which authors live (**Nakamoto**), being disapproved of by their peers and becoming an outcast in their social or work environment (**Jour, Employee, Socialdemokrat**) or in some cases intimidation and/or extortion by parties that wish to preserve or enrich themselves (**Nakamoto**).
We can now conclude that there are authors who prefer to remain anonymous and that these authors' preferences have different motivations. The application of stylometry as defined in 1.1 is therefore in conflict with the preference of some authors.

1.3 Contributions

Contributions of this work include:

Chapter 3

- We report accurate measurements of the performance of feature based authorship attribution methods, which improve our understanding of author anonymity.
- We report how homogeneity of author characteristics affects the performance of authorship attribution methods.
- We improve the state of the art in authorship attribution techniques for one specific data set.

Chapter 4

- We report accurate measurements of the performance of feature based authorship attribution methods against authors implementing an adversarial writing tactic, which improve our understanding of author anonymity.
- We report differences in writing style between obfuscated and non-adversarial texts.
- We develop a method for the identification of the obfuscation attack and report how effective this method is.
- We develop a method for (partial) de-obfuscation of obfuscated texts and report how effective this method is.
- We report and motivate what adversarial tactic an author with a preference for anonymity may want to use.

Chapter 5

- We improve understanding of how language models can be used for gender attribution by accurately reporting what language model we have employed for this task.
- To show that a feature based method for the attribution of gender is similarly effective as language modelling.
- We show that language models can successfully be applied for the attribution of authorship.
- We show there is ample future work on authorship attribution using language models, and explain why this future work is likely to improve the state of the art in authorship attribution.

1.4 Thesis outline

Chapter 2 is a literature overview and provides a summary of other research in the field of computational stylometry. In Chapter 3 we describe how we develop our own method for authorship attribution, based on an existing, state of the art method, and report the effectiveness of both. In Chapter 4 we investigate how an author's adversarial behavior affects her anonymity, and what stylometric methods can be employed specifically to target authors who write in an adversarial way. In Chapter 5 we investigate gender attribution using both feature based methods and language models, and how these language models can also be used for the attribution of authorship. Still in Chapter 5, we list future work on authorship attribution using language models. Finally, our conclusions are listed in Chapter 6.

Chapter 2

Related Work

In 2008 Patrick Juola wrote a 102-page description of the state of the field of authorship attribution [14]. Based on books and papers that were published at the time, Juola found it difficult to compare results described in the various publications because different data were used in different studies. Authorship attribution in 2008 had already been applied for analysing a great variety of texts: short and long texts, formal and informal writings, mixed-domain and domain-specific texts, and texts from different languages. Juola noted that the lack of comparable results might hamper recognition and progression of the field. The studies that are cited by Juola in [14] all report successful attribution attempts, indicating that authorship attribution is possible under many different conditions. However, the publications discussed by Juola do not assume an adversarial setting. This contrasts with our work which focuses on authorship attribution under adversarial conditions.

First, we will discus methods for non-adversarial authorship attribution. These methods form the basis for existing methods of authorship attribution under adversarial conditions and methods that we develop in this work. Second, we discuss existing work on authorship attribution under adversarial conditions to explain what methods produce the baseline performance to which we will compare our methods.

2.1 Non-adversarial Stylometry

Although many different methods for non-adversarial stylometry have been developed, in this section we will only discuss the methods most relevant to the adversarial attribution methods that we will be developing in our work.

Abbasi et al. introduced the writeprints feature set for authorship attribution in 2008 [15]. The writeprints feature set consists of tens of thousands of features, but by using sparse encoding, this feature set can typically be represented using only a few thousand features. These features include letter- and word-level lexical features, word-level syntactic features, text-structural features, and idiosyncratic features which capture common misspellings. Abbasi et al. used a support vector machine (SVM) with unspecified kernel for their attributions.

The writeprints feature set by Abbasi et al. was an inspiration for the creation of the writeprints-static feature set by Michael Brennan, Sadia Afroz and Rachel Greenstadt [16]. They have compiled and released[1] the Extended Brennan-Greenstadt Corpus, which contains texts by 45 authors. We will use this data set in our experiments and compare the performance of our method to that of theirs. The authors used an SVM with polynomial kernel for making their predictions.

Stylometric techniques for authorship attribution have been applied to small groups of authors [17] and to groups of up to 100.000 authors [20]. This shows that authorship attribution methods can be used at large scales.

Authorship is not the only attribution that has been made on the basis of texts. Schler et al. [24] have shown that gender can be attributed to authors based on their blog texts. The authors have

[1] `https://psal.cs.drexel.edu/index.php/Main_Page`

made their data set available to the public[2]. In [23], Sarawgi et al. used the blog data set by Schler et al. to attribute gender using n-gram language models. They built character and part-of-speech (POS) tag models for the texts of both genders, and then attributed gender based on the model under which a text is most likely to occur.

2.2 Adversarial Stylometry

To apply stylometric authorship attribution methods to texts of authors who want to remain anonymous is adversarial towards these authors, and it is therefore an adversarial application of stylometry. Similarly, if authors take any stylometric precautions against stylometric analyses, the authors' behavior is considered adversarial towards the entity that wishes to perform an analysis. Texts that are produced while the author made an attempt to thwart a possible future attribution attempt are said to implement an 'attack', and these texts are called 'adversarial texts' [17]. If the application of a stylometric analysis is in conflict with an author's preference, then the analysis lies in the domain of adversarial stylometry. This research is about adversarial stylometry, and we will now discuss what has already been done in this domain.

The first paper to provide insight into the domain of adversarial stylometry is by Brennan et al. [17]. They record and release the first public data set containing both natural and adversarial texts, called the Brennan-Greenstadt Adversarial Stylometry Corpus with texts by 12 authors. Two types of attacks are implemented by participants and recorded in their data set: the *obfuscation attack* and the *imitation attack*. In the obfuscation attack, each participant is instructed to produce an obfuscated text. The obfuscation should make authorship attribution of the text difficult, but no specific instruction for text obfuscation is provided. To implement the imitation attack, each participant is asked to mislead attribution attempts and to try to have the text be attributed to a well-known author of whom example texts are provided.

In 2012 Brennan et al. [16] released the Extended-Brennan-Greenstadt Corpus, which contains texts by 45 authors with texts that implement the obfuscation attack, the imitation attack and no attack (natural texts, produced by natural writing behavior). The authors implement the writeprints-static feature set, which is based on the full writeprints feature set by Abbasi et al. [15]. While the writeprints feature set contains a variable number of features, dependent on the text that is analysed, the writeprints-static feature set contains a constant 557 features. An SVM with polynomial kernel is used to attribute texts based on the 557 writeprints-static features. Brennan et al. show that their method performs better than other authorship attribution methods, including the original writeprints method, when applied to both the obfuscation and imitation attack as well as non-adversarial texts. They also conclude that both the obfuscation and imitation attacks are still highly effective against their method of authorship attribution, reducing their attribution accuracy to that of random chance or below.

In this work, we analyse and improve upon the methods proposed by Brennan et al. and Sarawgi et al. to create a better understanding of how organizations and individuals can use stylometry in an adversarial way. We investigate how authors with a preference for anonymity can be affected by authorship attribution, how they might effectively defend against an authorship attribution attempt, and how these defenses against authorship attribution can once again be overcome.

[2] `http://u.cs.biu.ac.il/~koppel/BlogCorpus.htm`

Chapter 3

Basic Authorship Attribution

Authorship attribution can be performed on the basis of features that are extracted from a text [17]. In this chapter, we investigate which features are most informative for the application of authorship attribution. We also create a better understanding of the performance of authorship attribution than was previously possible by providing richer performance measurements.

3.1 Objectives

In this chapter, as in the rest of this work, we investigate stylometric methods in adversarial settings, as defined in 1.1. However, in this chapter we will work with 'natural', non-adversarial texts exclusively, before focussing on adversarial texts in Chapter 4.

In most research into authorship attribution, the data set that was used was sampled from the general population. These data sets contain authors with a mix of genders, ages, and occupations that reflects that of the general population. Such a set of is said to be *heterogeneous* with respect to author gender, age, and occupation. While these data sets are useful, this situation does not reflect many cases of stylometry applied in adversarial settings. If the anonymity set consists of female university students, for example, then the anonymity set is more homogeneous than a set that is sampled from the general population. No scientific results about the effects of anonymity set homogeneity on authorship attribution have been reported yet.

Our main objectives in this chapter are as follows:

- Improve the method proposed by Brennan et al. [16], which is a state-of-the-art method for authorship attribution, by proposing new features and testing the effectiveness of existing and proposed features.

- Provide additional measurements of the success of authorship attribution methods on the Extended-Brennan-Greenstadt Corpus and the Blog Authorship Corpus that go beyond those already reported in the literature. These additional measurements will allow for a better understanding of the performance of authorship attribution and a better understanding of how authors are affected by authorship attribution.

- Provide measurements on how the gender, age, and occupation homogeneity of the anonymity set affects the success of authorship attribution. A homogeneous set of authors better represents the situation of real-world anonymous authors, but authorship attribution has never before been applied to homogeneous groups of authors.

Occupation	Frequency
Student	297
Arts	73
Education	69
Technology	66
Media	45
Non-Profit	29
Internet	28
Engineering	23
Publishing	21
Other	212

Table 3.1: Occupations of authors selected from the blog authorship corpus.

3.2 Experimental setup

3.2.1 Public data sets

Stylometric analysis needs to be applied to texts that come from a data set. As of July 2015, there is no distinct data set which is considered the most important set for benchmarking. To prevent further frustrating the emergence of a recognized benchmark data set and to be able to show the value of the contributions of our research, we will not create our own data set but will work with existing data sets for which results have already been published in other studies.
In this work, we will be using two public data sets. The first data set is the largest data set to contain adversarial texts, which will be investigated in Chapter 4. The second data set allows us to split authors based on gender, age and occupation, which we will do in Section 3.5. Others [16, 23, 24] have published results of their stylometric analyses on these two data sets, and we will compare our results with theirs.

Extended-Brennan-Greenstadt Corpus The first data set that is used was published by Brennan et al. [16]. 45 authors wrote a total of 757 texts. The distribution of gender and age of the authors reflects that of the general adult population. Each author wrote at least 13 texts that reflected their own writing style. These texts will be referred to as natural texts, or non-adversarial texts. In addition to non-adversarial texts, each author implemented the obfuscation and the imitation attack. The Extended-Brennan-Greenstadt Corpus is the largest publicly available corpus (in terms of authors, texts and number of words) that contains samples of both natural and adversarial texts. The corpus is publicly available.[1] We will refer to this data set as the EBG corpus or EBG data set.

Blog Authorship Corpus Schler et al. published [24] a data set containing 681.288 blog entries made by 19.320 authors. Each blog author declared an age, gender and occupation. From this data set, we filtered out all blog entries that were less than 500 words long and also only kept in authors with 14 or more blog posts. The reason for filtering out short texts and authors with few texts is to create a data set with a similarly sufficient number of words per text and texts per author as in the Extended-Brennan-Greenstadt Corpus. By using this filtering method, we selected 30.020 texts by 863 authors. The effects of author age, gender and occupation on the success of stylometric methods will be investigated in this chapter. Table 3.1 shows the distribution of occupations of the selected authors. The Blog Authorship Corpus is publicly available.[2] We will refer to this data set as the BA corpus or BA data set.

[1] `https://psal.cs.drexel.edu/index.php/Main_Page`
[2] `http://u.cs.biu.ac.il/~koppel/BlogCorpus.htm`

3.2.2 Metrics

The most common method to measure the success of an authorship attribution method has been recall@1 [14, 16, 17]. The recall@1 or R@1 evaluation method measures the success of authorship attribution after the attribution method has made one guess. The R@1 measure does not take into account a (partial) ordering of the possible authors except for which single author is most likely to have authored the text. In an adversarial setting, the R@1 measure is not necessarily the most informative measure. The anonymous author Employee, described in 1.2.2, might be in a set of, for example, 40 possible authors. These 40 authors constitute the *anonymity set* of the true author. A larger anonymity set can potentially provide more anonymity to the author than a smaller anonymity set, because the larger set offers more possible authors to hide amongst. In an adversarial setting, the objective of the attributer might not be to find the single most likely author. Instead, the attributer's objective might be to reduce the anonymity set size by selecting the n most likely authors. These n authors could then be subjected to further scrutiny while the rest of the anonymity set is (in practice) cleared of suspicion. In the Employee scenario, the initial anonymity set might consist of 40 authors. If the organization has resources to scrutinize 5 employees, then it might depend on stylometric tactics for deciding which 5 employees to further investigate. In this scenario, R@5 is the most relevant measure. If, based on scientific experiments, the recall@5 (for 40 authors) is known to be, for example, 0.95, then investigating the 5 most likely authors will imply investigating the true author with a probability of 0.95. Therefore, an author in the Employee scenario might prefer a lower rank in an attribution attempt in order to avoid scrutiny, even if the author is not ranked at position 1. We think that the R@n is of similar importance to R@1 in adversarial settings. We will therefore report R@1 in order to compare our results with those in the literature, but also report on $\{R@n \mid n \in 1...N\}$ as well as the average R@n, which is $\frac{1}{N}\sum_{n=1}^{N} R@n$.

3.2.3 Baseline

One recent and very successful method for authorship attribution was to use the Writeprints Static Feature Set (shown in Table 3.2) and a Support Vector Machine with polynomial kernel. This method was developed by Brennan et al. and published in [16] and is itself a simplification 2.1 of the original Writeprints approach [15].

Group	Category	No. of Features	Description
Lexical	Word level	3	Total words, average word length, number of short words
	Character level	3	Total char, percentage of digits, percentage of uppercase letters
	Letters	26	Letter frequency
	Digit	10	Digit frequency 0-9
	Character bigram	39	Percentage of common bigrams
	Character trigram	20	Percentage of common trigrams
	Vocabulary Richness	2	Ratio of hapax legomena and dis legomena
Syntactic	Function Words	403	Frequency of function words
	POS tags	22	Frequency of Parts Of Speech tags
	Punctuation	8	Frequency and percentage of colon, semicolon, qmark, period, exclamation mark, comma

Table 3.2: Writeprints-static feature set as described by Brennan et al.

It is unclear how or why the authors of [15] and [16] decided to use the features in Table 3.2 as opposed to leaving out one or more of them. There are many potentially useful features in this feature set, but it is not documented whether or not the usefulness of these features was tested. Also unclear is which features are normalized and how. It is not stated whether the frequency of an occurrence (for example, letter frequency) is measured in absolute terms or relative to the number of characters or words in a text. Value normalization of different features is a very important pre-processing step when using an SVM, because without it, some features become significantly more influential than others in the decision process. There are different strategies for feature normalization, which have different normalizing properties. In [16] by Brennan et al. there is no mention of feature normalization. In the writeprints-static feature set 3.2, there are 403 features that represent the frequency of function words. These features are necessarily sparsely populated in the research of Brennan et al. [16]. This is because in their data set, the average number of words per text sample varies between 503 and 667, with an average of 558, and there are at most 25 samples per author.

The approach to authorship attribution that we develop in this chapter is based on that of Brennan et al. [16]. In order to compare our results to theirs and find out if there is an improvement, we implement the approach by Brennan et al. and extend it to also report R@n.

Feature pre-processing is not discussed in [16], and although they do report using an SVM with polynomial kernel, the authors do not report the degree of the polynomial. We use Z-score as a feature pre-processing step and experiment with different low-order polynomial kernels, ultimately finding that a linear kernel yields the best performance in our experiments.

3.3 Method

The starting point for developing our method is the baseline described in 3.2.3. We will propose additional features, select a machine learning method, and then perform feature selection using the selected machine learning method. The method that we develop in this section will be similar to the baseline method, but with selected or pruned features.

3.3.1 Features

Some features, like sentence length, may have significant outliers and might be best described by some richer measure than the average. We have opted to use a Rich Vector Descriptor, RVD, that takes a real valued vector (e.g. a vector of observed sentence lengths) and produces 4 values that describe the vector: [average, median, average - median, standard deviation]. The average and median provide two different measures to indicate ‘typical’ values. Average - median provides an indication of the number of outliers times the size of the outliers in the observation, as well as the direction of these outliers. Standard deviation describes the spread of the observation, when the observation is modelled as a Gaussian distribution.

Table 3.3 shows all the features that we have considered for our approach. As seen in Table 3.3, none of the features that we use take into account the vocabulary used in the text. The use of vocabulary features might produce higher accuracy when performing authorship attribution. However, it could also be disadvantageous when an anonymous author writes about a subject that she would not write about under another identity. Not making use of the vocabulary as a feature can also increase topic invariance of the attribution method.

3.3.2 Selection of Machine Learning Method

The features in table 3.3 were extracted from the EBG data set. These features were fed to a number of machine learning algorithms. Standard Score or z-score normalization [19] and Principal Component Analysis (PCA) [21] were tested as pre-processing steps for each machine learning method individually. The Standard Score is a signed value indicating how many standard deviations a value is from the mean. This kind of normalization is more robust to outliers than simply dividing by the observed maximum, allowing for discrimination between typical values as well as extremes.

PCA is a statistical procedure for dimensionality reduction which reduces the feature vector to a pre-determined number between 1 and the number of features.

In this section, we only report the R@1 for the best combination of pre-processing step and machine learning method. Table 3.4 shows the recall of the two best performing machine learning methods. Recall was calculated using a 13-fold cross validation on all 45 authors. During each fold, each

Category	Feature count	Description
Unigram Character Distribution	47	Relative frequency for the characters a-z, space and special characters: .,!?()-/&<>[]:; Relative frequency of following three types: Special, a-z, uppercase
Bigram Character Distribution	81	Most frequent character bigrams. Together cover 68.2% (one σ) of bigrams.
Trigram Character Distribution	59	Most frequent character trigrams. Together cover 25% of trigrams
Unigram POS tag Distribution	12	Simplified tags from NLTK
Bigram POS tag Distribution	78	Most frequent POS tag bigrams. Together cover 68.2% (one σ) of bigrams. Includes symbols indicating the start/stop of a sentence.
Unigram Chunk Distribution	5	Relative distribution over the sentence chunks: NP, VP, PP, ADVP, ADJP
Bigram Chunk Distribution	6	Most frequent chunk bigrams. Together cover 68.2% (one σ) of bigrams. Includes symbols indicating the start/stop of a sentence.
Sentence Length Distribution	4	RVD of sentence lengths
Word Length Distribution	16	RVD of word lengths, Relative word length distribution for frequencies {1,2,...,11, 12+}
Legomena Fractions	5	Number of $2 - 6$ legomena over the number of hapax legomena
Readability	2	ARI and LIX readability estimators

Table 3.3: Table showing all features that we propose.

author had one text sample left out of training that had to be attributed to the correct author. All texts were non-adversarial.

3.3.3 Feature selection

Feature selection is a way to verify if the proposed features actually contribute to the quality of a machine learning method. We will use the SVM and Nearest Neighbors methods during feature selection. During feature selection, we always optimized for the R@1 of the machine learning method. We consider two main approaches to feature selection. The first is an additive approach to feature selection where we iteratively add more features to be used by the machine learning method. The second approach we will refer to as eliminative feature selection. When applying eliminative feature selection, we start out by using all the features and then iteratively remove features, still optimizing for recall. These two methods are an example of hill-climbing search [22] with two different start positions. Lastly, we also considered a ‘hybrid’ or fuzzy approach where we combine feature selection from both approaches and alternate between feature addition and elimination, still consistent with a hill-climbing search. The fuzzy approach did not improve results for this particular stylometric

Method	R@1	Pre-processing	Options
SVM	0.86	Standard Score	RBF kernel
Nearest Neighbors	0.77	Standard Score	4NN, weight based on distance
Random guessing	$0.0\overline{2}$		

Table 3.4: Recall of SVM and Nearest Neighbors methods after minor parameter optimization.

task, in contrast to another stylometric task described in Chapter 4. Table 3.5 lists the selected features for the SVM and Nearest Neighbors methods.

Method	Selected features	Recall	Recall increase
SVM	*Selected after feature removal:* Unigram, Bigram Character Distribution Unigram, Bigram POS Tag Distribution Unigram Chunk Distribution Word Length Distribution Legomena Fractions Readability *Removed:* Bigram Chunk Distribution Trigram Character Distribution Sentence Length Distribution	0.88	0.02
SVM	*Selected after feature addition:* Unigram, Bigram Character Distributtion Bigram POS Tag Distribution Word Length Distribution	0.87	0.01
Nearest Neighbors	*Selected after feature removal:* Unigram, Bigram Character Distribution Unigram, Bigram POS Tag Distribution Unigram Chunk Distribution Word Length Distribution Legomena Fractions Readability Sentence Length Distribution *Removed:* Bigram Chunk Distribution Trigram Character Distribution	0.80	0.03
Nearest Neighbors	*Selected after feature addition:* Unigram, Bigram Character Distributtin Bigram POS Tag Distribution Bigram Chunk Distribution Sentence Length Distribution Readability	0.81	0.04

Table 3.5: Results of feature selection for authorship attribution on non-adversarial texts.

The first thing to note in table 3.5 is that the Nearest Neighbors method has had a more significant improvement than the SVM method. This is likely because the Nearest Neighbors method had more room for improvement because of its lower initial recall. The eliminative search method has removed the features ‘Bigram Chunk Distribution’ and ‘Trigram Character Distribution’ for both

SVM and Nearest Neighbors. In the case of the SVM, the 'Sentence Length Distribution' was also removed. Interestingly, the 'Bigram Chunk Distribution', which had been consistently removed, was selected as an informative feature using the additive feature selection method for the nearest neighbors method. Overall, the feature selection did not yield a large improvement over the baseline recall.

3.4 Measurements

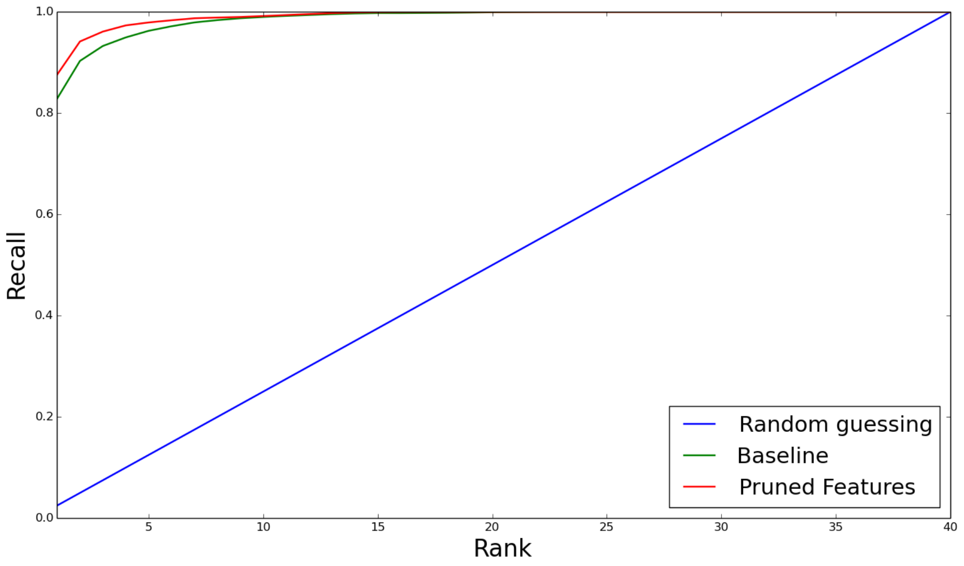

Figure 3.1: Recall@rank for attribution of non-adversarial texts of 40 authors.

Data Set	Method	Number of Authors	R@1	Average Recall
EBG	Baseline	20	0.88	0.99
		40	0.83	0.99
	Pruned Features	20	0.92	0.99
		40	0.88	0.99
BA	Baseline	800	0.38	0.95
	Pruned Features	800	0.38	0.95

Table 3.6: Measurements of R@1 and average recall.

In our experiments on the EBG data set, the highest recall is achieved using the following features: Unigram and Bigram Character Distribution, Unigram and Bigram POS Tag Distribution, Unigram Chunk Distribution, Word Length Distribution, Legomena Fractions. These features were Z-Score normalized and an SVM with RBF kernel was used for classification. This method is similar to the method that we use as a baseline, but the features have been pruned. Therefore, we refer to our method as the pruned features method. While optimizing for R@1, we found a recall of 0.88 on 45 authors using 13-fold cross validation.

Table 3.6 provides an overview of the measurements we performed on the baseline method and our pruned features method. We performed a 13-fold cross validation on 40 random subsets of 40 authors, for a total of 40 × 40 × 13 = 20800 attributions. The R@1 of our pruned features method for was 0.88. The previous state-of-the-art R@1 for the EBG data set was somewhere between 0.80 and 0.83 as reported in [16]. In our replication 3.2.3, we find a R@1 of 0.83 on 40 authors for the baseline method.

Figure 3.1 shows that the R@n of our method applied to non-adversarial texts is much higher than the recall of random guessing for all values of n, and also higher than the baseline method. The average recall of the pruned features method over all n for 40 authors is 0.99. At $n > 16$, our method has a recall of 1. This means that in the dataset that we used, the correct author is always identified within 17 guesses, which is less than half the number of authors. Figure 3.1 also shows the recall curve for the baseline method 3.2.3 for comparison. The baseline reaches a recall of 1 at $n > 21$. Figure 3.2 shows a comparison between the baseline method 3.2.3 performance and the performance of our pruned features method, measured using 13-fold cross validation on 40 random subsets of 20 authors.

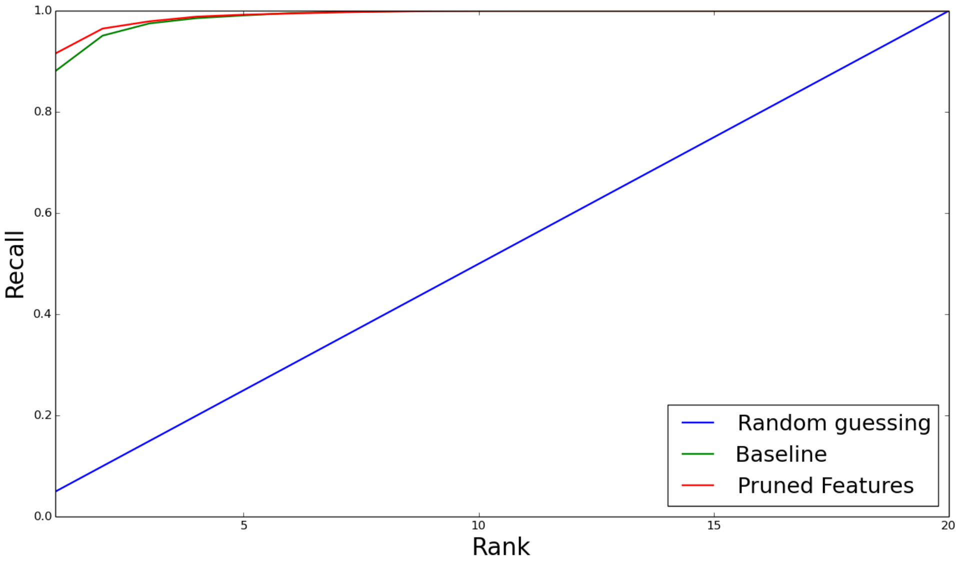

Figure 3.2: Recall@rank for attribution of non-adversarial texts of 20 authors.

When measuring the average recall of our method for sets of $n = 2-40$ authors, there is no correlation between the number of authors and the average recall; the mean average recall for $n = 2-40$ authors is also 0.99

We also measured the R@n for the blog authorship corpus using the methods that were developed on the EBG corpus. Figure 3.3 shows the recall curve. The recall for this plot is calculated by 2-fold cross validation on 10 sets of 800 randomly selected authors. This makes for a combined $2 \times 10 \times 800 = 16000$ attributions. For our pruned features method, R@1 is 0.38, recall > 0.99 occurs at $n > 478$ and the average recall is 0.95.

As shown in Figure 3.3, the baseline method by Brennan et al. slightly outperforms our pruned features method when applied to the blog authorship corpus. However, the baseline method depends on the vocabulary that the author employs. Because blogs are generally about a single topic or within a single genre, the higher recall that the baseline method has over the pruned features method when applied to the BA data set might be because of topic dependence.

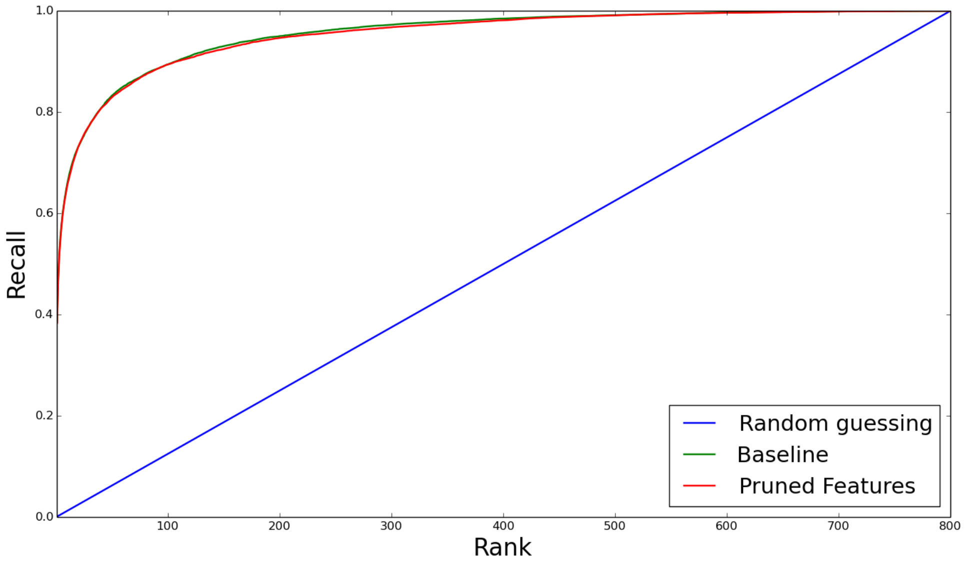

Figure 3.3: Recall@rank for attribution of non-adversarial texts of 800 authors.

3.5 Effects of homogeneity of the group of possible authors on recall

The data set 3.2.1 that we have used in Sections 3.3.2 and 3.3.3 was sampled from the general population. This means that the gender, age, and occupation of the participants also reflected that of the general population. Groups that reflect the diversity of the general population are called heterogeneous 3.1. In many of the real-world applications of adversarial stylometry, the set of possible authors is not heterogeneous. If there is a leak in a department of a company, then the group is likely to be homogeneous. If the leak comes from the engineering department, then the possible authors are (almost) all engineers with a university education. Because of the current lack of diversity in the engineering industry, it is likely that the vast majority will be men; such a group is not heterogeneous but homogeneous.

In the context of authorship attribution, the question that arises is whether the homogeneity or heterogeneity of the group of possible authors has an influence on recall. The following subsections investigate the effect on recall of having homogeneous groups in terms of gender, occupation, and age.

3.5.1 Gender

First we investigate the difference in recall for authorship attribution in mixed, male, and female groups. We perform a cross validation by sampling 160 groups of 20 authors in the mixed, male, and female categories. For each author in each group of authors, we select one text that needs to be attributed and the other texts are used to learn the authors' writing styles. Figure 3.4 shows little difference in recall for authorship attribution for male, female, or mixed groups. The only apparent difference is that male authored texts are slightly more difficult to attribute when measured in recall@1-5 on 20 authors. Authorship attribution in groups of mixed gender is as difficult as authorship attribution in female-only groups.

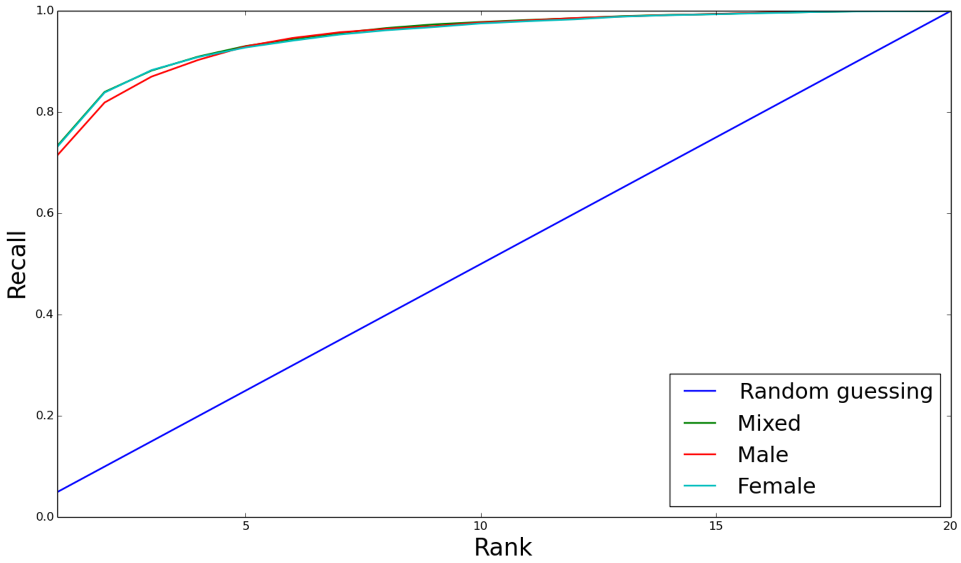

Figure 3.4: Recall@rank for attribution of texts of 20 authors comparing mixed, male, and female author groups.

3.5.2 Occupation

We have selected the 9 occupations that occur more than 20 times in our selection of authors. For each occupation, we again select 160 sets of 20 authors. In each set of authors, we leave out one text per author which will be attributed while the other texts are used to learn the authors' writing styles. A special category 'Mixed' represents a heterogeneous group, where all occupations are represented as they occur in the dataset.

Figure 3.5 shows the recall curve. The most significant outlier is the 'Publishing' group of authors. Texts by these authors within this group are the most difficult to attribute. The texts from the 'Internet' and to a lesser extent the 'Education' categories are the easiest to attribute.

3.5.3 Age

The blog authorship corpus contains authors across various ages. We measured the performance of authorship attribution for three different age groups and compared the recall with that of a mixed-age group. We sampled 160 sets of 20 authors. In each set of authors, we leave out one text per author which will be attributed while the other texts are used to learn the authors' writing styles.

Figure 3.6 shows the effect that age has on the recall of authorship attribution. When measuring recall@n, we find that authorship attribution for authors in groups of teenagers performs the lowest for all n. When the author is guaranteed to be in a group with age 30+, the recall is higher than all other categories for all values of n. Recall of authorship attribution for the group of age $20 - 30$ is similar to that of the mixed group, and both of these recall curves are above that of the teenage recall curve and under the 30+ recall curve.

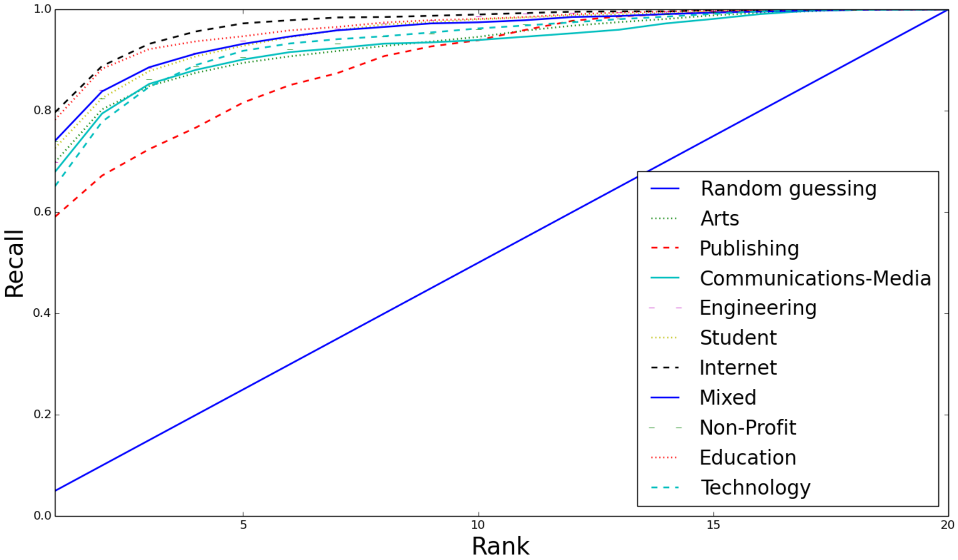

Figure 3.5: Recall@rank for attribution of texts of 20 authors comparing various occupation groups.

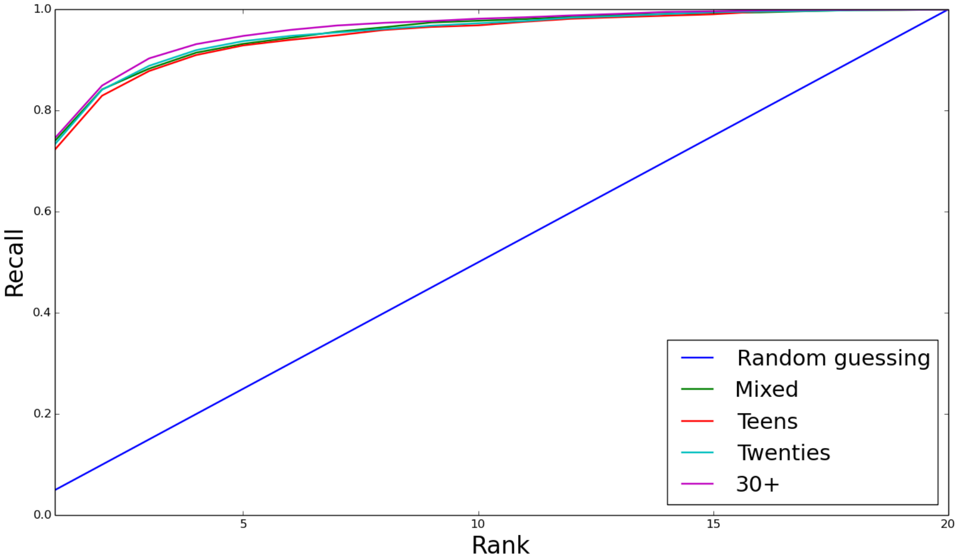

Figure 3.6: Recall@rank for attribution of texts of 20 authors comparing age groups.

3.6 Future work

When attributing authorship to a text in an effort to identify an author, the goal is to reduce the size of the anonymity set that the author likely resides in. This permits a targeted follow-up investigation into the identity of the author, which is more resource efficient.
To be able to identify characteristics of an author, like gender, occupation, age, native language (family), level of education, and possibly other characteristics, can aid an investigation into the

identity of an author because it can be used to reduce the anonymity set size that the author resides in.

Commercial- and government-protected secrets are accessible to people with different backgrounds and characteristics. To identify an author that makes these secrets public, one may start by not applying authorship attribution directly to the set of possible authors (as we will do throughout this work), but instead attribute other author characteristics to the author first in an effort to identify her. Because an organization may know characteristics like gender and native language for each employee, attributing these characteristics can quickly reduce the anonymity set size if attributed accurately.

Gender attribution applied to typed natural language texts under adversarial conditions is discussed in Chapter 5, but the attribution of other author characteristics in an adversarial setting have never been investigated. We believe that if such attributions can be made with high recall or accuracy, then this type of attribution can further reduce the anonymity set size that the author resides in.

Nota bene: the methods that we apply throughout this work could also be tested for the attribution of other author characteristics, like the above mentioned. However, an in-depth investigation into the attribution of many characteristics is beyond the scope of this research, and we therefore focus our entire work on attributing authorship, from which all other characteristics may be deduced once successfully applied; future work may focus on attributing other author characteristics as one of several steps in an authorship attribution process in order to improve the state-of-the-art authorship attribution in adversarial settings.

3.7 Conclusion

Our main conclusions in this chapter are:

- R@1 is not the only relevant measure of author anonymity. Recall at higher n should be a concern for many authors with a preference for anonymity.

- An extended and then pruned feature set can produce a higher attribution recall compared to an existing, state-of-the-art method that we implemented.

- Homogeneity in the set of possible authors with respect to some author characteristics affects recall. We have investigated the effect of the following three author characteristics:

  Gender Attributing texts when all possible authors are male is slightly more difficult than when the possible authors are female or gender mixed. This is visible in the lower n when measuring R@n. There is no clear difference in recall between attributing texts for female-only authors or authors from a mixed gender group.

  Occupation Occupation also has an effect on recall. We find that attributing authorship of texts within the 'Publishing' group is more difficult than in any other group. Attributing texts within the 'Internet' and to a lesser extent 'Education' groups is easier. A mixed-occupation group of authors shows the average recall of the recalls of occupations that comprise it.

  Age We find that within the teenage group of authors, authorship attribution recall is lowest, and within the 30+ age group it is highest. Within the age group twenties, authorship attribution recall is similar to that of the mixed age group and lies between the recall for the teen and 30+ age groups.

- We propose a new feature set and show that it outperforms an existing feature set on which our feature set is based.

- We improve the state of the art in authorship attribution techniques for one specific data set.

Chapter 4

Authorship attribution for adversarial texts

In this work, we investigate stylometric authorship attribution under adversarial conditions. In Chapter 3, we focused on adversarial conditions wherein the author did not write her texts in an adversarial way. In this chapter, we investigate what the implications are for authorship attribution when an author does write in an adversarial way.

As described in Section 3.2.1, the EBG corpus is the largest corpus to contain both natural and adversarial texts. Each of the 45 authors has implemented the obfuscation attack and the imitation attack. In the obfuscation attack, each participant is instructed to produce an obfuscated text. The obfuscation should make authorship attribution of the text difficult, but no specific instruction for text obfuscation is provided to the authors. To implement the imitation attack, each participant is asked to mislead attribution attempts and to have the text be attributed to a well-known author of whom example texts are provided to the participant.

Because this chapter is exclusively about adversarial texts, we will only work with the EBG data set. This is the largest data set to contain adversarial texts.

4.1 Motivation

In [16] and [17], Brennan et al. have reported that when texts are written in an adversarial way (through obfuscation or imitation), then the attribution recall drops significantly. Table 4.1 shows recall in authorship attribution against both types of stylometric attacks, using the pruned features method described in Chapter 3 and originally developed for attributing non-adversarial texts.

Method	Attack	R@1
SVM with RBF kernel. Features selected for attribution of non-adversarial texts	None	0.88
	Obfuscation	0.11
	Imitation	0.01

Table 4.1: R@1 of authorship attribution of adversarial texts.

Table 4.1 shows that the R@1 of authorship attribution is greatly reduced when the author implements the obfuscation or imitation attack. This is consistent with the findings by Brennan et al. Figure 4.1 shows that our method applied to adversarial texts generally performs better than random guessing. The exceptions are the R@1 against the imitation attack, and R@20-25 against the obfuscation attack. When not only measuring the R@1 but all R@n, we initially find that it is

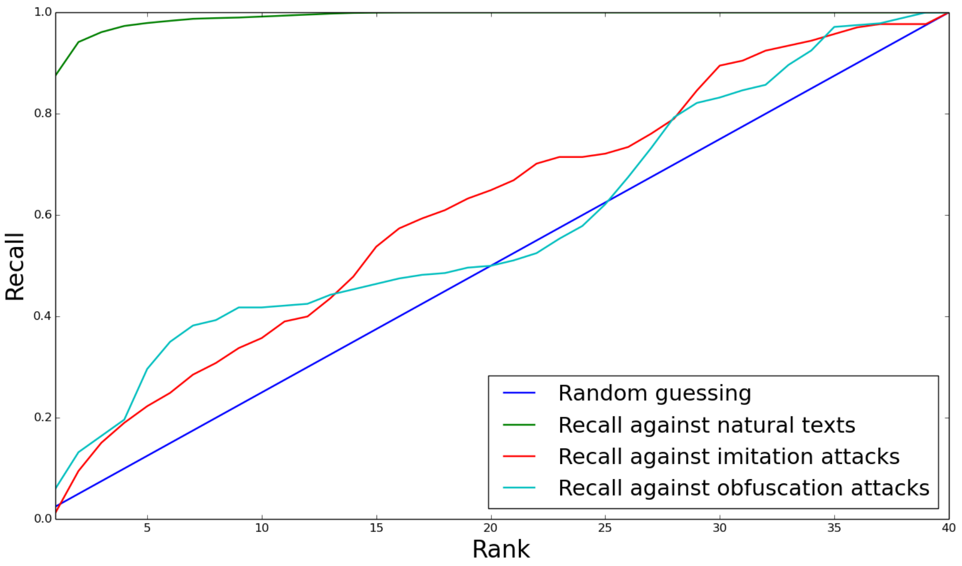

Figure 4.1: Recall@rank for attribution of adversarial texts on 40 authors.

not at all clear which attack is more successful. Until $n = 13$, the imitation attack is more successful in a set of 40 possible authors, but when $n > 13$, the obfuscation attack succeeds better at hiding authors' identities. This is in contrast with the findings by Brennan et al. in [16], which conclude that the imitation attack performs significantly better at hiding authors than the obfuscation attack. The likely reason for this difference is that Brennan et al. only measured R@1, while other recalls are relevant as well, as we have discussed in 3.2.2.

Figure 4.1 also shows that attributing authorship to adversarial texts is significantly more difficult than attributing authorship to natural texts.

When authors have a preference for anonymity, it is reasonable for them to use a writing tactic that would contribute to their anonymity. Because both types of adversarial texts are known to be effective against authorship attribution, authors with a preference for anonymity might be inclined to use one of these adversarial writing tactics in an effort to not be revealed. From the author's perspective, it is important to fully understand the effects of the tactic that she uses in order to understand the risks to her anonymity. From the attributer's perspective, adversarial texts merit more attention because they are more difficult to attribute; moreover, such texts are potentially more high-profile because employing an adversarial writing style indicates that the author has made an investment in her anonymity.

4.2 Chapter outline

In this chapter, we will further investigate stylometric authorship attribution applied to adversarial texts. The imitation attack that was recorded in the EBG data set records has all the participants imitating the same author. The accuracy as well as other findings might be significantly affected by the choice of author that is imitated. Therefore, it is not always useful to apply the same analyses of the obfuscation texts to the imitation texts. This is why in this chapter we focus on the obfuscation attack but also study the imitation attack when sensible.

We will show how the obfuscation attack can be detected. This detection can then be used to improve authorship attribution against an obfuscation attack by reversing typical obfuscation behavior on a feature level. Detection of the imitation attack would be another interesting problem, but the text samples that are readily available would not produce meaningful results. The EBG data set that

we are using contains only imitation attacks targeting the same author. This means that classifying this attack would overlap with classifying authorship by that particular author.
In Section 4.3 we apply feature selection specifically for attributing obfuscated texts. Section 4.4 describes how the obfuscation attack can be detected. The feature-level de-obfuscation is described in Section 4.5. After we have shown that text de-obfuscation is possible, we compare our method to the baseline method described in Section 3.2.3, and thereby show that we consistently outperform the baseline method against both attacks. This comparison can be found in Section 4.6. Smaller surveys and observations regarding adversarial texts are described in our 'Miscellaneous' Sections 4.7.1 - 4.7.4. Finally we outline our conclusions in Section 4.8.

4.3 Attribution of adversarial texts

We re-run the procedure described in Section 3.3 with respect to feature selection, but instead of attributing only non-adversarial texts, we now attribute only adversarial texts. This has two purposes. The first is to try to achieve high recall when attributing adversarial texts. The second is to discover which features are (most) robust against the obfuscation attack, and possibly other stylometric attacks.
The EBG data set contains only one or two samples per author for the obfuscation and imitation attacks, in contrast to at least 13 samples of non-adversarial texts. In order to do a cross validation, we selected 7 random sub-samples of 40 authors out of the set of 45 authors. This means the prior recall (random guessing) is $\frac{1}{40}$. We trained only on non-adversarial texts.
Table 4.2 shows the results of feature selection and the recall against the two types of attacks. The elimination feature selection was not included in this table because it had no significant differences with the results for non-adversarial feature selection shown in table 3.5. As can be seen in table 4.2, our feature selection selects no more than 3 feature categories when optimizing for adversarial texts. This is much less than the 7 feature categories that were selected for the attribution of non-adversarial texts in Section 3.3.1. When using the pruned feature set for attribution of obfuscated texts, the attribution R@1 is 0.13 instead of 0.12. Attribution under an imitation attack benefits much more from feature selection specific to the attack, increasing the R@1 from 0.01 to 0.12, which is again similar to the obfuscation attack.

Attack	Method	Selected Featurs	R@1
Obfuscation	SVM	Unigram, Bigram Character Distribution	0.13
	NN	Sentence Length Distribution Unigram Character Distribution Word Length Distribution	0.11
Imitation	SVM	Bigram Character Distribution Bigram Chunk Distribution Legomena Fractions	0.12
	NN	Bigram Character Distribution	0.07
	Random guessing		0.03

Table 4.2: Results of feature selection for authorship attribution for adversarial texts.

Feature selection specific to adversarial texts shows that the Bigram Character Distribution feature category is the most salient indicator of the true author.

4.4 Identification of the obfuscation attack.

Although the detection of an obfuscation attack is only a two-class problem, the approach we use in this section for this problem is similar to our approach for the multi-class problem of authorship attribution described in Chapter 3.

4.4.1 Selection of machine learning method

Method	Accuracy	Preprocessing	Options
Always classifying as non-adversarial	0.9439		
AdaBoost	0.9863	Standard Score	80 estimators, 0.998 learning rate
SVM	0.9751	Standard Score	Linear kernel

Table 4.3: Accuracy of machine learning methods using all features for the detection of an obfuscation attack.

Our cross validation method for development is as follows:
We split our data set of 45 authors into 45 training - test sets. Each training set contains the obfuscated and un-obfuscated texts of 44 authors, while one author is left out for testing. The texts of the author that is left out are then used to measure accuracy of the non-adversarial versus obfuscated classification that was trained on the 44 other authors.
There are at least 13 non-adversarial texts per author, and 1 obfuscated text. In total there are 45 obfuscated texts that are classified and 757 non-adversarial texts. The prior accuracy (always classifying as non-adversarial) is $\frac{757}{757+45} \approx 0.9439$. Only the best performing parameters and preprocessing steps for each method are documented in table 4.3. We find that the AdaBoost and Support Vector Machine methods produce the highest accuracy in our experiment.

4.4.2 Feature selection

We perform feature selection in a similar way as explained in 3.3.3. We use hill climbing with three strategies: additive feature selection, eliminative feature selection, and 'fuzzy' feature selection where we perform both elimination and addition on the current feature selection and interchange the selection between the AdaBoost and SVM methods. Table 4.4 shows the results of feature selection. The highest increase for AdaBoost is reached using fuzzy feature selection, which gave an accuracy increase of 0.0050. This increase might seem low, but relative to the theoretically possible increase it is high. The best possible increase from feature selection is from 0.9863 to 1, and relatively our increase is $\frac{0.9913-0.9863}{1-0.9863} \approx 0.36$ of what is possible.
Table 4.4 shows which features are most useful for discrimination between non-adversarial and obfuscated texts in our experiments. The SVM and AdaBoost methods are very distinct computations, but they mostly agree on which features are useful for discrimination. The most important features for the identification of the obfuscation attack are Bigram and Trigram Character Distribution and Unigram POS tag Distribution. The highest accuracy was achieved by using the AdaBoost machine learning method with the following features: Bigram and Trigram Character Distribution, Unigram POS Tag Distribution, Sentence Length Distribution, Word Length Distribution, Legomena Fractions, Readability.

4.4.3 Effectiveness of obfuscation detection

In the two class classification problem, recall is not the only measure necessary to understand the behavior of the classifier. Table 4.5 shows the obfuscation attack classifications in more detail. This measure was performed on 11 subsets of 40 authors. In each subset, a classification was performed

Method	Selected features	Accuracy	Accuracy Increase
AdaBoost	*Additive feature selection:* Bigram Character Distribution	0.9813	-0.0037
	Eliminative feature selection: Unigram, Bigram, Trigram Character Distribution Unigram, Bigram POS Tag Distribution Unigram, Bigram Chunk Distribution Sentence Length Distribution Word Length Distribution Legomena Fractions *Removed features:* Readability	0.9875	0.0012
	Fuzzy feature selection: Bigram, Trigram Character Distribution Unigram POS Tag Distribution Sentence Length Distribution Word Length Distribution Legomena Fractions Readability	0.9913	0.0050
SVM	*Additive feature selection:* Bigram, Trigram Character Distribution Unigram POS Tag Distribution	0.9875	0.0125
	Eliminative feature selection: Bigram, Trigram Character Distribution Unigram, Bigram Chunk Distribution Unigram, Bigram POS Tag Distribution Word Length Distribution Sentence Length Distribution Readability *Removed features:* Unigram Character Distribution Legomena Fractions	0.9888	0.0137

Table 4.4: Feature selection for detection of the obfuscation attack.

for each author, where learning was applied to 39 authors and the texts of the author that was left out were classified. This resulted in the classification of a combined 7853 texts, of which 440 were obfuscated and 7413 were non-adversarial. The prior accuracy is $\frac{7413}{7413+440} \approx 0.9440$.

	Absolute	Relative
Accuracy	7719	0.9830
True Positives	324	$0.73\overline{63}$
True Negatives	7395	0.9976
False Positives	18	0.0024
False Negatives	116	$0.26\overline{36}$

Table 4.5: Performance of obfuscation detection measured in accuracy, true-false positives and true-false negatives.

The AdaBoost method erroneously classified 18 out of 7413 non-adversarial texts as obfuscated texts, which is less than a quarter of a percent. Of the 440 obfuscated texts, 116 were classified as

non-adversarial while over 73 percent was classified correctly. These results are used in Section 4.7.4 to investigate if one famously anonymous author implemented the obfuscation attack.
In Section 4.5, we show that it is possible to learn obfuscation behavior and to undo the obfuscation on a feature level, to some extent.

4.5 Feature-level de-obfuscation of obfuscated texts

4.5.1 Intuition

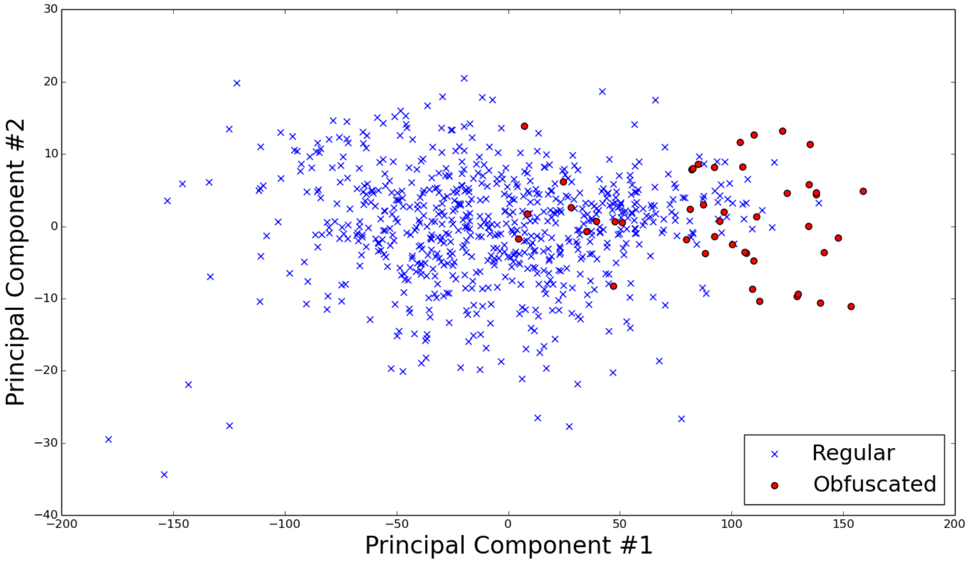

Figure 4.2: Obfuscated versus non-adversarial texts projected onto two dimensions.

To visualize the obfuscated texts in comparison to the non-adversarial texts, we projected the text features onto two dimensions (Figure 4.2). The projection was performed using principal component analysis [21] on what we found to be the most discriminative features 4.4.2. It should be noted that authors were not instructed how they should obfuscate their texts and they did not discuss text obfuscation with each other. However, as can be seen in figure 4.2, the authors behave in a typical way when writing obfuscated texts. Obfuscated texts can be found in the right hand side of figure 4.2 with not a single deviation. Moreover, this typical behavior is already visible when projecting the specially selected 180 dimensional feature space on a (well chosen) two dimensional space. Because the projection is done using principal components, it cannot be understood from the figure exactly how the obfuscation behavior is different from non-adversarial behavior. For a better understanding of how obfuscation behavior is manifested, see Section 4.7.1. If the human subjects that participated in creating the data set implemented a typical operation that changes features from non-adversarial texts to obfuscated texts, then this operation might be reversible to some extent. Figure 4.2 indicates that there is a typical operation that people perform when creating an obfuscated text.
De-obfuscation of obfuscated texts can be considered a *counter-attack* against an author.

4.5.2 De-obfuscation

The de-obfuscation is performed using data from 40 authors. The non-adversarial texts of all 40 authors are used for authorship attribution as described in Chapter 3. From 39 authors, we use the obfuscated and non-adversarial texts, from which we learn the obfuscation behavior. The text that will be de-obfuscated and attributed is the obfuscated text from which we do not learn the obfuscation behavior.

Each author has an average feature value for each feature that can be extracted from their non-adversarial texts. These average feature values may differ from the feature values of their obfuscated texts. We learn the average feature value differences between the obfuscated and non-adversarial texts of all 39 authors and average these differences. These differences can then be added to a remaining obfuscated text, which needs to be attributed.

In more abstract detail, this procedure involves the following steps:

1. For each author $a \in A$, collect all $N = |a|$ feature vectors $\vec{f_a^n}$ of non-adversarial texts written by author a.
2. For each author $a \in A$, collect the feature vector $\vec{O_a}$ of the obfuscated text written by author a.
3. Find the average feature vector $\vec{F_a}$ of the non-adversarial texts for every author a: $\vec{F_a} = \frac{1}{|a|} \times \sum_{n=1}^{|a|} \vec{f_a^n}$
4. Find the average distance vector $\vec{d}$ between the average feature vectors of the authors' non-adversarial texts and their obfuscated text feature vectors: $\vec{d} = \frac{1}{|A|} \times \sum_{a \in A} \vec{F_a} - \vec{O_a}$
5. To calculate the de-obfuscated feature vector $\overrightarrow{f_{de-obf}}$ from any obfuscated feature vector $\overrightarrow{f_{obf}}$, add the average distance vector $\vec{d}$ to $\overrightarrow{f_{obf}}$: $\overrightarrow{f_{de-obf}} = \overrightarrow{f_{obf}} + \vec{d}$

Nota bene: in the EBG corpus, each author wrote exactly one obfuscated text. If there is more than one obfuscated text per author, the vector $\vec{O_a}$ should be constructed in way similar to the construction of $\vec{F_a}$, detailed in Item 3.

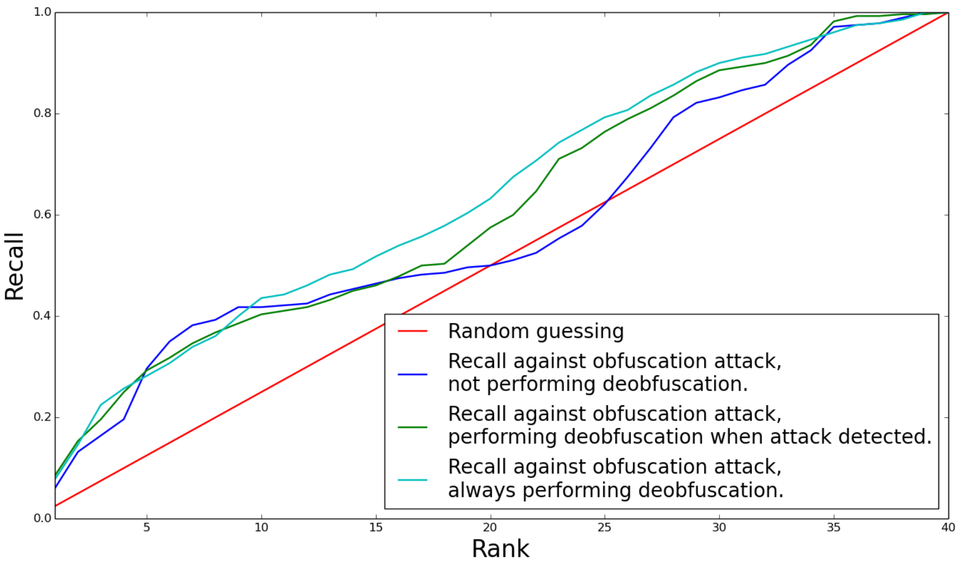

Figure 4.3: Recall@rank for attribution of obfuscated texts using de-obfuscation.

Figure 4.3 shows how the authorship attribution of obfuscated texts is improved by this de-obfuscation method. The recall was calculated on 11 sets of 40 authors, for a total of 440 obfuscated text attributions.

Note that there are two lines describing the recall of de-obfuscation in figure 4.3. One line represents the recall when always de-obfuscating the text. This serves to prove the hypothetical effectiveness of de-obfuscation when all obfuscation attempts would be correctly identified. The second line displaying recall using de-obfuscation represents a more realistic scenario where de-obfuscation is only performed when the obfuscation is detected. The obfuscation detection is trained on the 39 texts that are also used to learn de-obfuscation.

We have tested the effects of giving different weights w to the de-obfuscation method, where w would be used as follows: $\vec{f_{de-obf}} = \vec{f_{obf}} + w \cdot \vec{d}$ With weight $w = 1.0$, the de-obfuscation method would add exactly the average feature value differences between the obfuscated texts and non-adversarial texts. With weight $w = 0.5$, only half the average difference would be added. We tested the weights $0.0, 0.5, 0.9, 1.0$, and 1.1 and found that the highest average recall is reached using a weight of $w = 1.0$.

The de-obfuscation improves the recall of authorship attribution against obfuscated texts, both when the de-obfuscation is always performed and when de-obfuscation is only performed when obfuscation is detected. The increase in recall gained from de-obfuscation does not completely undo the effectiveness of the obfuscation attack. However, it does show that to some extent the obfuscation behavior follows a typical pattern. The de-obfuscation method that we implemented was ad-hoc; more advanced de-obfuscation methods might be able to further increase the recall against the obfuscation attack.

Future work on feature-level text de-obfuscation One improvement to the de-obfuscation method might be to generate more obfuscation data, as 39 samples might be a small set from which to accurately learn the obfuscation behavior. Another possibility would be to look at the difference of obfuscated and non-adversarial texts in a lower dimensional space. This could be done by first applying PCA to compress the features to a space of 30 or so dimensions and learn the obfuscation behavior in that space. Then, the feature differences could be transformed back to the original, higher dimensional space. The transformation from the lower to the higher dimensional space would be inaccurate (because of the nature of PCA), but it allows for a generalization of the behavior. This generalization would be based on features from both obfuscated and non-adversarial texts, as the learned PCA transformation can be based on features of texts from both categories. This method of lower dimensional de-obfuscation might compensate for the fact that not many obfuscated texts are available.

4.6 Comparisons to baseline

In this section, we compare the R@n of our pruned features attribution method with that of the baseline method described in Section 3.2.3.

Figure 4.4 shows our pruned-features method, with de-obfuscation when obfuscation is detected, in comparison to the baseline method for the attribution of obfuscated texts, measured in R@n. The pruned-features method performs better than the baseline method for all n, except for $15 < n < 20$, where both methods show similar recall. This shows that our method for the attribution of obfuscated texts performs better than the baseline method.

Figure 4.5 shows the R@n of our pruned features method for the attribution of imitation texts in comparison with the R@n of the baseline method. The pruned features method shows a higher recall than the baseline method, for all values of n. This shows that our method for the attribution of imitation texts performs better than the baseline method.

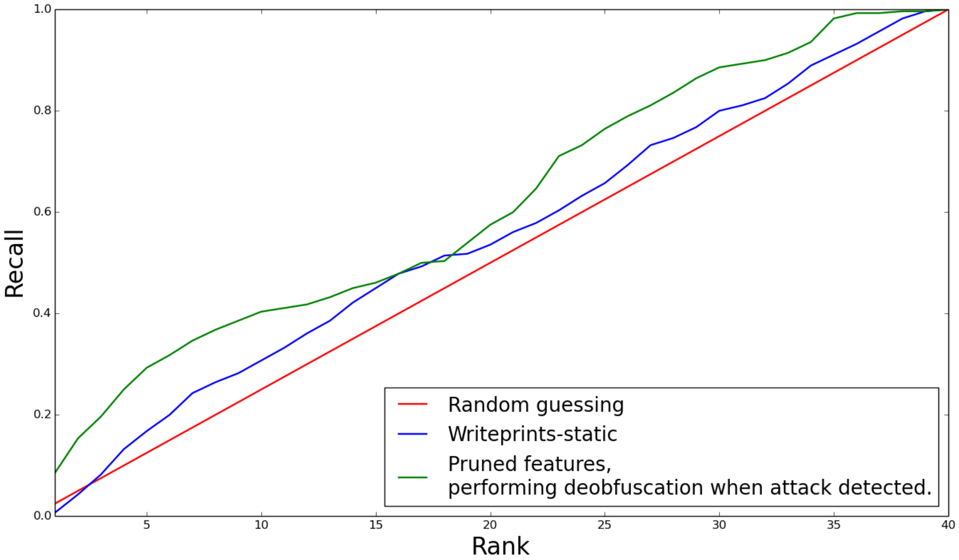

Figure 4.4: Recall@rank baseline comparison for attribution of obfuscated texts.

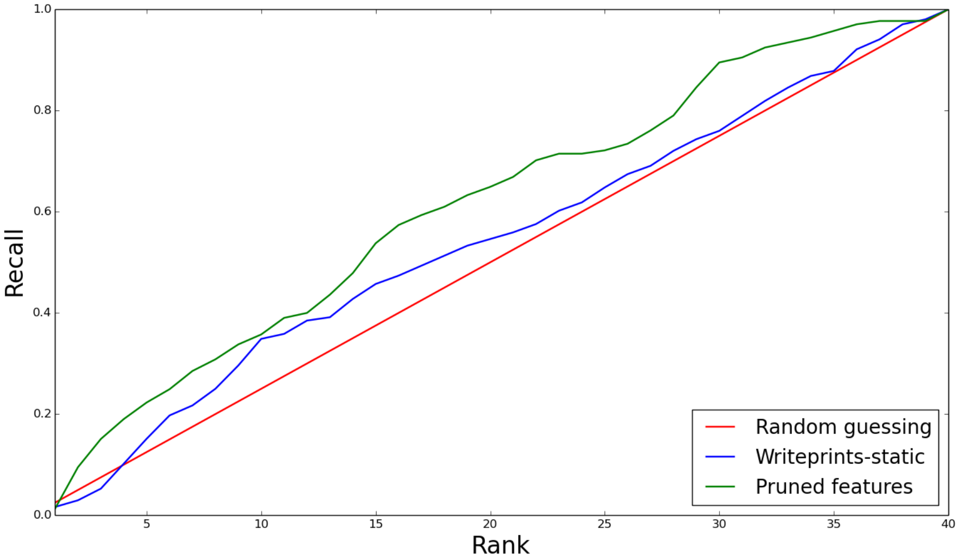

Figure 4.5: Recall@rank baseline comparison for attribution of imitation texts.

4.7 Miscellaneous

What follows are four subsections. The first two sections aim to better understand the adversarial behavior of authors using the techniques that we have developed so far. Section 4.7.3 investigates what tactics an author might want to use if she wishes to remain anonymous. In Section 4.7.4, we investigate whether a contemporary and famously anonymous author implemented the obfuscation attack.

4.7.1 Characteristics of obfuscated texts

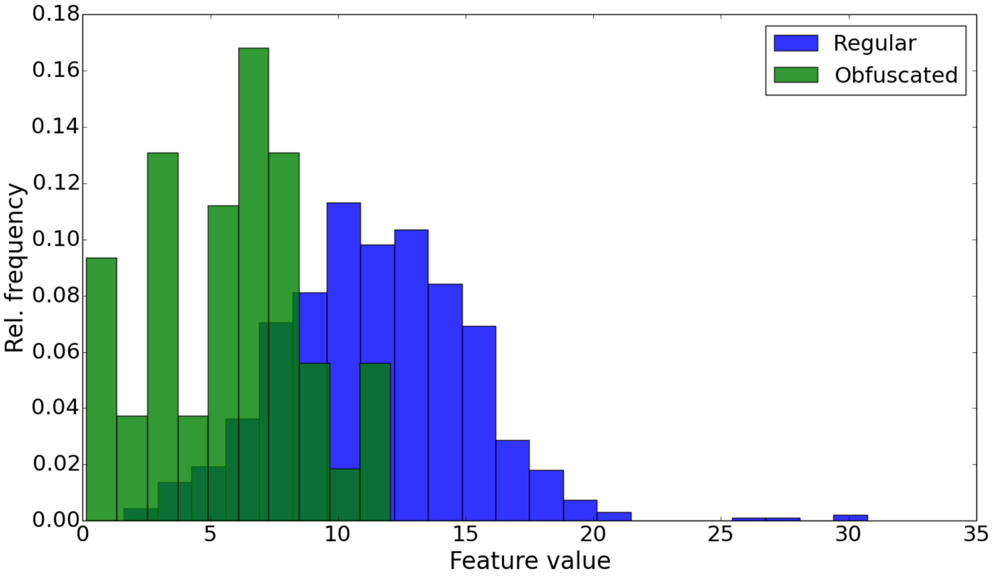

Figure 4.6: Difference between the Automated Readability Index (ARI) scores of obfuscated and non-adversarial texts.

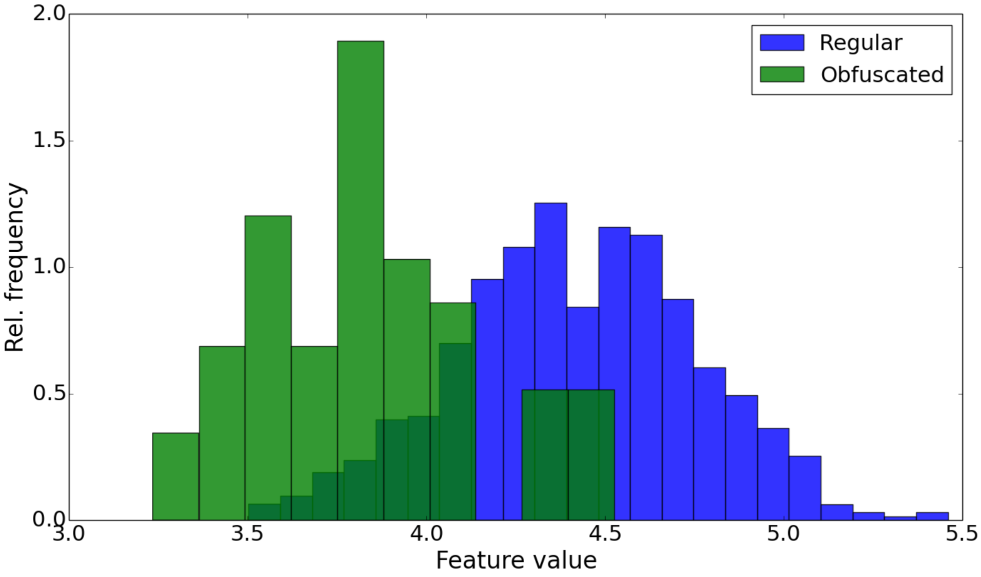

Figure 4.7: Difference between the average word lengths in obfuscated and non-adversarial texts.

We have manually investigated the differences between features of obfuscated and non-adversarial texts. For this, we used all 757 non-adversarial texts and all 45 obfuscated texts. We have chosen to present figures of three features 4.6, 4.7, 4.8 that have different typical values for obfuscated and non-adversarial texts. These are features with the greatest difference between adversarial and non-adversarial texts.

Figure 4.6 indicates that obfuscated texts are less complex, measured in Automated Readability Index (ARI). When measuring readability in LIX, we find a similar difference. The ARI and LIX readability measures are partially based on word lengths. We find that the average word length of obfuscated texts is less than that of non-adversarial texts, as can be seen in figure 4.7. Figure 4.8 shows that the use of adverbial phrases (ADVP) is much more frequent in obfuscated texts; the use of an adverbial phrase is about two times less frequent in non-adversarial texts.

Note that a similar analysis for the imitation texts in the EBG data set would not be sensible because only a single author was targeted for imitation.

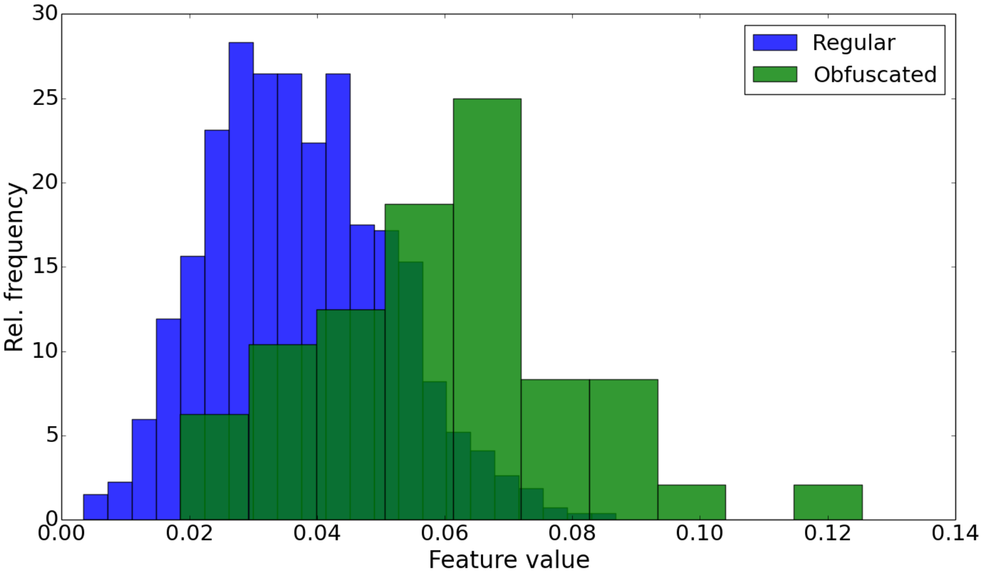

Figure 4.8: Difference between the relative frequency of the ADVP sentence chunk in obfuscated and non-adversarial texts.

4.7.2 Possible conflation of adversarial tactics

Figure 4.9 shows attribution of adversarial texts, but with an important difference to Figure 4.1. In Figure 4.1, the texts that are used during training of the attribution classifier are only the non-adversarial texts for each author. In 4.9, the training texts also contain all non-adversarial texts, but in addition to these, the adversarial texts of the other category are included in the training set. This means that when attributing obfuscated texts, the training data for each author contains non-adversarial texts and imitation attack texts. When attributing imitation attack texts, the training data for each author contains the non-adversarial texts and also the obfuscated texts. We call this *cross attack learning*. As can be seen in 4.9, the R@n for almost all n increases by adding samples from the other writing attack strategy. This result indicates that different adversarial behaviors of authors might be conflated to some extent. However, these results are not conclusive because only a single author was targeted during the imitation attacks.

4.7.3 Considerations for authors with a preference for anonymity

In this chapter, we investigated the effectiveness of two adversarial tactics: obfuscation and imitation. After performing de-obfuscation, the recall of our method was very similar against both adversarial tactics. However, there are considerations beyond recall. While it remains unclear how to identify the imitation attack, we found that obfuscation can be identified with high accuracy. An author may

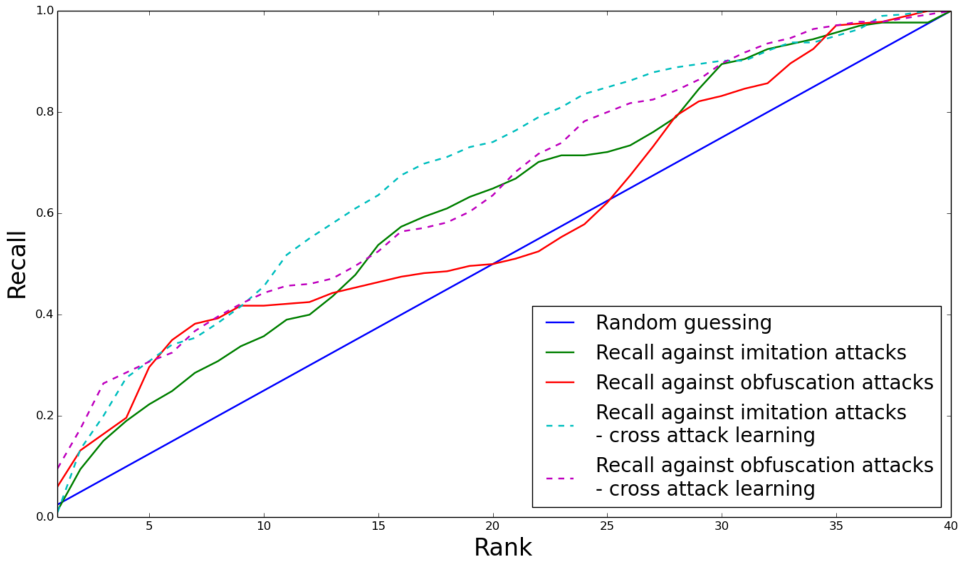

Figure 4.9: Recall@rank for attribution of adversarial texts on 40 authors. Cross-learning from adversarial texts in other category.

want to hide the fact that she implemented an adversarial writing style. If adversarial behavior is identified by an adversary, then investigative resources might be distributed to more authors, which increases the risk of exposure for the anonymous author. Also, the de-obfuscation method that we described in this chapter was ad-hoc. Better de-obfuscation methods might exist and should be anticipated. Because there are many potential authors to imitate, reversing the imitation attack is likely more complicated than reversing the obfuscation attack.

In the EBG data set that we used in this chapter, there was only one author who was being imitated, and this author had a very distinct writing style. The effectiveness of the imitation attack might decrease if the author to be imitated does not have a very distinct writing style.

In the usage of the imitation attack, when writing multiple messages under the same pseudonym, it may be important for authors to consistently target the same author for imitation. Otherwise, the average feature values of the pseudonym targeting different authors for imitation might reveal features that are more close to representing the true writing style of the author wishing to remain anonymous. This is a counter-attack to de-imitate multiple imitation texts at once by averaging feature values from all the texts that the anonymous author wrote. This attack on the author's anonymity is closely related to the de-obfuscation attack described in Section 4.5. However, as there is no method to identify the imitation attack, it is yet unclear when this counter-attack should be implemented.

With these considerations in mind, we think it wiser for authors with a preference for anonymity to implement the imitation attack instead of the obfuscation attack and to consistently imitate a single author who has a distinct writing style.

4.7.4 The case of Satoshi Nakamoto

One contemporary anonymous author, or possibly group of authors, is known under the pseudonym 'Satoshi Nakamoto' 1.2.2. Nakamoto is the designer and author of the first bitcoin protocol and client. Her (or his) creation was accompanied by a whitepaper [1]. Subsequently, Nakamoto engaged in online discussions. While there has been much speculation about the identity of Nakamoto [13], her (or his) identity remains unknown to the public. At least two attempts to identify Nakamoto used stylometry as part of the attempt [8, 10].

If someone were to use stylometric techniques as part of the identification of Satoshi Nakamoto, then it would be relevant if Nakamoto has made an attempt to obfuscate her (or his) writing style 4.7.3. We have applied our obfuscation detection method to writings produced by Nakamoto. We sampled texts from her (or his) white paper and forum posts. The exact samples that we used are found in appendix A. It should be noted that our method for obfuscation detection assumes that the author is singular. That is, the method only discriminates between non-adversarial texts by single authors and texts by single authors that obfuscated their writing style. If Nakamoto is a group, then this would be a third category or class for which we have not yet developed a classifier and for which our classifier does not apply.
All the 6 Nakamoto texts that we investigated were classified by our predictor as non-obfuscated texts. As shown in Table 4.5, the probability of our method misclassifying a single obfuscated text is $0.26\overline{36}$, and misclassifying 6 obfuscated texts is $0.2636^6 \approx 0.0003$. The chance of our method correctly classifying 6 non-obfuscated texts is $0.9976^6 \approx 0.9857$. We believe that the accuracy of our method, combined with the fact that all 6 texts were classified as non-obfuscated, provides a very strong indication that Nakamoto was not a single author implementing the obfuscation attack. This would leave open the possibility that Nakamoto is either a group or a single author that did not implement the obfuscation attack. If Nakamoto is a single author, she (or he) might have written in a non-adversarial way, or she (or he) might have implemented the imitation attack or some other attack.

4.8 Conclusion

Our main conclusions in this chapter are:

- The obfuscation and imitation attacks greatly reduce the performance of authorship attribution measured in R@n, for all n, and are therefore both very effective methods against authorship attribution.

- The obfuscation attack can be identified with high accuracy, even when there are just 39 obfuscation examples provided to the classifier.

- The obfuscation attack is reversible to some extent. While our method for text de-obfuscation was ad-hoc, better methods for text de-obfuscation are likely to exist.

- The pruned features method for the attribution of obfuscated texts, as well as imitation texts, performs better than the baseline method by Brennan et al., measured in R@n.

- Although the recall against the obfuscation and imitation attacks is similar, there are indications that the imitation attack has better anonymizing properties than the obfuscation attack. In addition, authors with a preference for anonymity may benefit from their attack going unrecognised, to avoid a more equal distribution of investigatory resources and to not incite counter-attacks. These two factors lead us to believe that a correct implementation of the imitation attack (always targeting the same author, with distinct writing style) may be more beneficial to the anonymity of an author than the implementation of the obfuscation attack.

- Satoshi Nakamoto is likely not a single author who implemented the obfuscation attack.

Chapter 5

Gender Attribution and Language Models for Authorship Attribution

In Chapters 3 and 4 we have attributed authorship to natural and adversarial texts, by improving a state of the art method by Brennan et al., resulting in the pruned writeprints attribution and de-obfuscation methods. In this chapter, we start by investigating how language models have been used for attributing the gender of an author, by replicating the Ruchita et al. gender attribution method [23]. Then, we show that a pruned writeprints approach to gender attribution is equally successful for the attribution of gender. Finally, we show that language modelling can be used not only for gender attribution, but also as an effective method for authorship attribution.

5.1 Motivation

Schler et al. [24] show that gender attribution based on blog data can be performed with high accuracy by using content words. The authors show that female bloggers are much more likely to use the words 'cute' and 'pink', while male authors are more likely to use the words 'linux' and 'gaming'. These features clearly capture cultural stereotypes of both genders. Although a dependence on cultural stereotypes might not be problematic in certain settings, this work is focussed on attribution in adversarial settings. In adversarial settings, we cannot depend on women using the words 'cute' and 'pink' and males writing about 'linux' or 'games'. Even if these words were used in an adversarial text, they may well be false clues. An alternative to using vocabulary for gender attribution is to use other features, like the ones we first proposed in Chapter 3 in Table 3.3, or to use language modelling other than word modelling.

In [23], Ruchita et al. show that gender attribution is possible using language models. In their approach a different language model is trained for each of the genders based on the training data for each respective gender. When attributing gender to a text from the test set, the likelihood under each model is calculated. The language model under which the text is most likely to occur indicates the gender. The authors have used the following language models: Probabilistic Context Free Grammars (PCFG) that do not incorporate lexicon in the final probability output, and $1, 2$ and 3-gram language models over the characters and Parts Of Speech tags (POS-tags). If language models can be successfully applied for the attribution of gender, the question that naturally arises is whether language models can also be used for the attribution of authorship. We will try to perform authorship attribution, using the same language modelling approach that we develop for gender attribution, to see if language modelling is a viable alternative approach to a feature based authorship attribution method.

5.2 Chapter outline

In this chapter, we start by replicating the work by Ruchita et al. on n-gram language models for gender attribution in Section 5.3. Then, in Section 5.4, we show that a feature based, writeprints-style approach to gender attribution is equally successful for the attribution of gender as language models are. In Section 5.5, we continue to use the same language model for the attribution of authorship of natural, and adversarial texts. Section 5.6 is dedicated to future work on language models for authorship attribution. We conclude this chapter in Section 5.7.

5.3 Language models for gender attribution

Unfortunately, Ruchita et al. have failed to specify the details of any of their language models in their paper [23]. Therefore, we come up with our own language model, which is also based on n-grams, and report the details of our language modelling approach. The data set on which Ruchita et al. measured the performance of their language models has not been made public yet. However, the authors report using texts from various blogs, and therefore, we think the use of the blog authorship corpus which we also used in Chapter 3 is appropriate.

5.3.1 Employed language model

Probability estimation The language model M is trained by feeding it all the n-grams that can be observed in the training data. The model M can then be queried the following: $|M|$, the size of the model defined by the total number of n-grams that were observed in the training data; $|M_g|$, the frequency of an n-gram g in the training data.
The likelihood of a text T occurring, given a model M is estimated as follows:

$$P(T|M) = \prod_{g \in G(T)} \frac{|M_g|}{|M|} + \frac{1}{|M|}$$

Where $G()$ is a generator that generates all n-grams g from a text T.
The first term, $\frac{|M_g|}{|M|}$, represents the un-smoothed probability estimation of observing an n-gram g, under the model M. If g is an n-gram unobserved by the model, the un-smoothed estimated probability is zero.
The second term, $\frac{1}{|M|}$ is a smoothing constant to prevent probabilities of zero. This is a simple, ad-hoc, ‘plus one’ smoothing solution. An improvement may be found by testing other smoothing values, or to use a more advanced smoothing technique. We did not investigate either of these options.

n-grams In [23], Ruchita et al. extract 1,2 and 3-grams for characters and POS-tags from each text. We also extract n-grams from each text, of lengths 1 until 5, and besides character and POS-tag n-grams we also extract chunk n-grams. For characters, we include spaces and any special characters we encounter, without substitutions; for POS-tags and sentence chunks, we add a single start/stop symbol between sentences, and n-grams cross sentence boundaries. This means that if the sentence [a,b] is followed by the sentence [c,d], this is represented as [a,b,⟨s⟩,c,d], which contains the following 3-grams: (a,b,⟨s⟩), (b,⟨s⟩,c), (⟨s⟩,c,d).

5.3.2 Measurements

In our experiments, we use the blog authorship corpus to train two language models, one for each gender that is represented in the data set.
For each model that we tested, we selected 7 random subsets of 150 authors. From each subset, the texts of 100 authors were used for training the models, while the texts of 50 authors were used as a test set. The total number of gender attributions varied between 10000 and 13000, and in

each test set there were an equal amount of male and female authors. We tested 1-5-grams. Table 5.1 shows the accuracy of the best performing character, POS-tag, and chunk n-grams. Ruchita et al. performed similar experiments using n-gram language models, also on a blog-based data set. In their experiments, they find that 2-gram language models perform best for both character and POS-tag based language models, and did not test chunk based language models. They report an accuracy of 0.71 when using character 2-grams, and 0.66 using POS-tag 2-grams. The accuracy we measure is a few percent lower than that of Ruchita et al. , but it should be noted that we tested on a different data set. While the data set that we used is publicly available 3.2.1, Ruchita et al. failed to publish the data they used. Similarly, we have provided a detailed documentation of the language model we used, while Ruchita et al. only specify their language model as an n-gram language model over the characters that outputs probability estimates, failing to provide other details.

Category	Best performing n	Accuracy
Characters	2	0.66
POS-tags	1	0.64
Chunks	1	0.54

Table 5.1: Accuracy of gender attribution using n-gram language models.

5.4 Feature based gender attribution

In Section 5.3, we show that language models can be used for the attribution of gender, confirming work by Ruchita et al. In this section, we show that gender attribution is also possible using a feature based method.

Method	Accuracy
SVM with RBF kernel	0.65
AdaBoost with 15 estimators	0.64

Table 5.2: Accuracy of gender attribution of machine learning methods using all features.

We used the feature set that we first introduced in Section 3.3.1 to select two machine learning methods in a similar way as described in Section 3.3.2. We used the same test set-up as described in Section 5.3.2. Table 5.2 shows the performance of the two best performing machine learning methods. The accuracy is 0.65 and 0.64 for the SVM and AdaBoost methods respectively. We performed feature selected in the same way as described in 3.3.3, but found that feature selection did not provide any significant improvements.

The accuracy of the feature based methods is very similar to that of the n-gram based language models. This shows that a feature-based attribution technique can successfully be applied to the task of gender attribution.

5.5 Authorship attribution using language models

In Section 5.3 we have successfully replicated the experiments performed by Ruchita et al., showing that n-gram based language models can be used for gender attribution. We also showed in Section 5.4 that a feature based approach, developed for authorship attribution, can be employed successfully for gender attribution too. In this section, we investigate to what effect language models can be used for authorship attribution.

5.5.1 Language models for the attribution of non-adversarial texts

Through experiments we found that a 2-gram character model outperforms all other language models. For comparing the effectiveness of language models for authorship attribution on the EBG data set, we performed a 13-fold cross validation on 40 random subsets of 40 authors, for a total of $40 \times 40 \times 13 = 20800$ attributions. We also measured the performance of language models for authorship attribution on the blog authorship corpus. There, we applied 2-fold cross validation on 10 sets of 800 randomly selected authors. This makes for a combined $2 \times 10 \times 800 = 16000$ attributions. Table 5.3 shows a comparison between the language models used for authorship attribution, baseline method developed by Brennan et al., and our pruned features method, both of which we reported on in Chapter 3. Table 5.3 only reports R@1, but figures 5.2 and 5.1 show the entire recall curve.

	R@1 against 800 Authors	R@1 against 40 Authors
Brennan et al. baseline	0.38	0.83
Pruned Features	0.38	0.88
2-gram character model	0.39	0.91

Table 5.3: R@1 of different authorship attribution methods against non-adversarial texts.

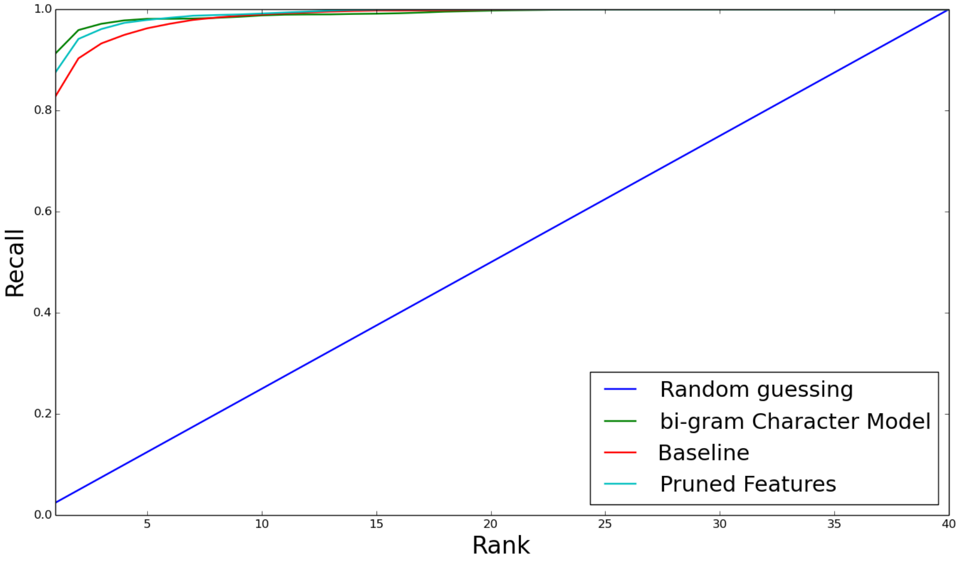

Figure 5.1: Recall@rank for attribution of non-adversarial texts of 800 authors.

As can be seen in table 5.3, the language model for authorship attribution results in a higher R@1 than the other two authorship attribution methods, on both the EBG and blog authorship corpora. Figure 5.1 shows that on the EBG data set the language model is also performant on other recall measures. However, when viewing the recall curves for the blog authorship corpus, we see that the language model generally performs lower than the feature based models, even though the R@1 of the language model is higher.

Typically, a blog covers a limited number of topics in which the blog author has an interest; different blog entries from the same blog cannot be expected to be cover different domains. In contrast to the blog authorship corpus, EBG data set is actually cross domain in that authors were asked to write about completely different topics. The fact that the language model performs on-par with the feature based models on the EBG data set, but not on the blog authorship corpus, indicates that

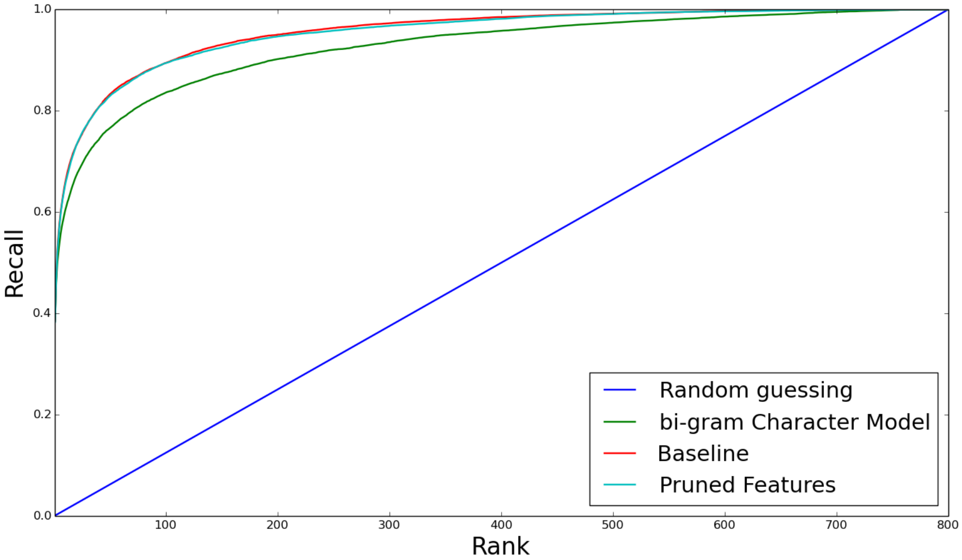

Figure 5.2: Recall@rank for attribution of non-adversarial texts of 40 authors.

language models are less suited for cross-domain authorship attribution.

In Section 3.2.2, we explained why R@1 is not the only relevant performance measure, and R@n should also be considered. With the results that we have presented in this section, we have shown that the R@1 is not necessarily an indicator for which method performs best at higher n, thereby providing a real-world example of the necessity of performance measures beyond R@1.

Side Note. We have now found that the two-class classification problem of gender attribution can be performed with a typical accuracy of around 0.65. The multi-class problem of authorship attribution can be performed with a R@1 of around 0.90 when there are 40 classes, each class representing an author. (Both numbers based on natural writing styles.) This is the case for both feature and language modelling based attribution methods.

The fact that authorship attribution for a set of many authors has a higher R@1 than gender attribution is accurate as a binary classifier, indicates there is a significantly greater variety in the language models and text features amongst individual authors, regardless of their gender, than there is between the two genders.

5.5.2 Language models for the attribution of adversarial texts

	R@1 against obfuscation attack	R@1 against imitation attack
Random guessing	0.03	0.03
Pruned Features	0.09	0.01
2-gram character model	0.10	0.05

Table 5.4: R@1 of different authorship attribution methods against adversarial texts.

For measuring the performance of language models for the attribution of adversarial texts, we selected 7 random sub-samples of 40 authors out of the set of 45 authors. For each author, we train a model based on the non-adversarial texts, and then try to attribute the adversarial texts.

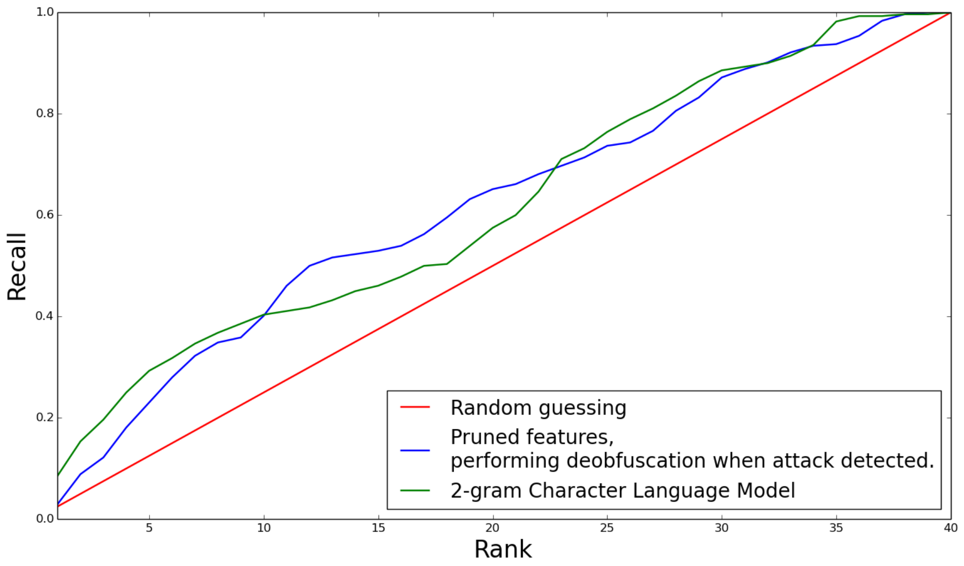

Figure 5.3: Recall@rank for authorship attribution against the obfuscation attack.

In our comparison between the language model and the pruned features method, we employed our feature-level de-obfuscation method, described in Section 4.5, on the obfuscated texts. Table 5.4 shows the R@1 for different attribution methods. Figures 5.3 and 5.4 show the recall curves for the attribution of texts implementing the obfuscation and the imitation attack. There is no clear 'best' attribution method when observing the recall curves.

5.6 Future work

Smoothing. The language model that we used in this chapter to estimate the likelihood of a text occurring is ad-hoc, and no effort was made to try a more advanced smoothing method. A more advanced smoothing method might help the language model to perform better.

Combining attribution methods By combining the outputs of two different authorship attribution methods one might be able to create a better, third attribution method. There are different strategies for combining different rankings, or the output of different language models. What follows are two examples that might be especially useful for authorship attribution.

Combining language modelling methods. In a language modelling approach, the likelihood of a text under each model is calculated. These likelihoods can be normalized to create a probability distribution over the classes. Different language models can then be combined by combining the probabilities that the language models assign. For example, a POS-tag and a character based n-gram model for one class can each output a probability by applying a normalization of the likelihood that the models output. These probabilities can then be combined to assign a final likelihood to the text being a member of a class. Different models can be assigned different probability mass when combining creating the final likelihood estimation. By this mechanism, a character model ans a POS-tag model can work together to form a likelihood.

Interleaving of dissimilar methods. As can be seen in Figure 5.1 and Table 5.3, the language modelling approach performs better than the feature based approaches when observing lower n in

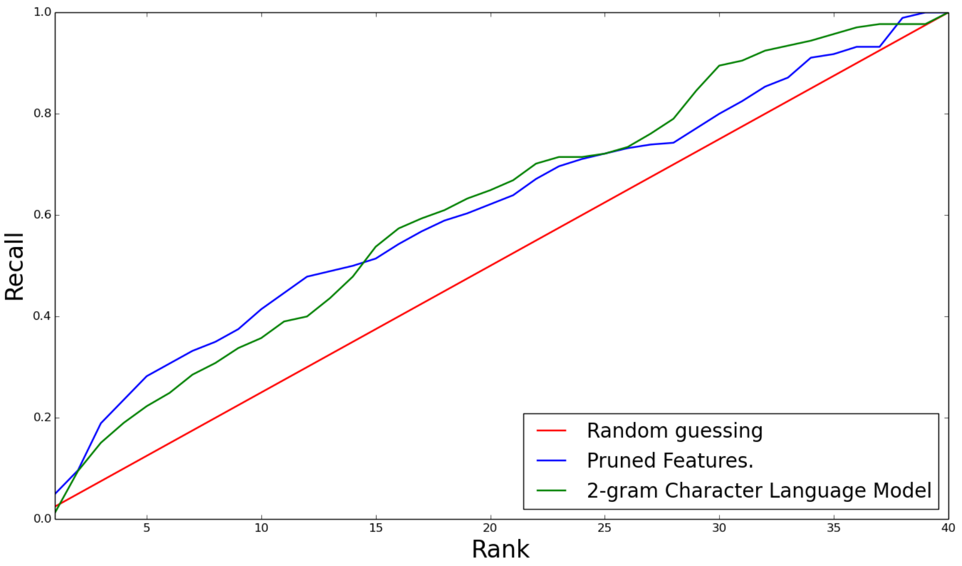

Figure 5.4: Recall@rank for authorship attribution against the imitation attack.

measuring R@n, while feature based methods outperform language models for higher values of n. One possibility is to interleave the outcome of both attribution methods. This can be done using an interleaving method that assigns equal value to both methods, for all n, but in this case another interleaving method might produce better results; because the language modelling approach performs better, only at the lower n, an appropriate interleaving method might be to draw the first, for example, $\frac{1}{5} \times N$ guesses from the language modelling attribution method, and have the pruned writeprints method order the guesses for higher n.

Language models applied against the obfuscation attack. In Chapter 4 we have shown that a feature based method can be used to create counter-attacks against the obfuscation attack. In particular, we looked at identification of the obfuscation attack, and feature-level de-obfuscation. Similar methods might be constructed using language models instead of features.

Identification of the obfuscation attack. In Section 4.4 we explained a novel technique for the identification of the obfuscation attack, which exhibits high accuracy. Using language models, it is be possible to create a model for natural texts and for obfuscated texts. Then, the likelihood of a text can be calculated for each model, and based on this a classification can be made. This is a different approach to the identification of the obfuscation attack than the method described in Section 4.4, with one significant benefit. The method described in Section 4.4 can only output a class, and not a probability distribution over the classes. However, with a language modelling approach, a probability distribution over the classes arises naturally, since the probability of the text is already calculated under each model. This allows for the introduction of a decision boundary that is not just based on maximum likelihood, which might result in the highest accuracy, but on other considerations such as the reduction of false positives or the incorporation of a prior probability, established by some other method, such as the outcome of a binary obfuscation detector.

Language models for text de-obfuscation. In the following, we assume that obfuscated texts are correctly recognized as such.

In Section 4.5 we explained our novel technique for (partial) feature-level text de-obfuscation, and proved its effectiveness. A similar approach can be applied to language models by applying a process we call n-gram text de-obfuscation. The idea behind n-gram text de-obfuscation is, similar to feature-level text de-obfuscation, to discover in what way (on an n-gram level) obfuscated texts differ from natural texts. If it is discovered that an n-gram has a higher relative frequency in adversarial texts than in natural texts, a proportional discount to the relative frequency of that n-gram should be applied, if an obfuscation attack is identified in the text; if, on the other hand, it is discovered that an n-gram has a lower relative frequency in obfuscated texts, the relative frequency of that n-gram should be increased if an obfuscation attack is detected.

5.7 Conclusion

Our main conclusions in this chapter are:

- We show that language models can be used for the attribution of gender, confirming results by Ruchita et al.
- 2-Gram character language models produce higher accuracy for gender attribution than other n-gram language models of characters, POS-tags or sentence chunks. We report exactly what kind of language model we used, improving our understanding of how language models can be used for gender attribution.
- A feature based, writeprints approach to gender attribution is similarly effective for gender attribution as our 2-gram character model is. Feature selection for gender attribution did not significantly improve accuracy for gender attribution.
- Language models can be used for authorship attribution. In particular, a 2-gram character language model performs best in our experiments.
- A 2-gram character language model for authorship attribution performs better when measuring R@n for lower n, but worse at higher n, compared to a pruned features approach.
- When observing the differences between texts via language models or (pruned) features, there is a greater variety between individual authors than there is between male and female authors in general.
- There are many opportunities to improve and expand the application of language models for authorship attribution and other stylometric tasks.

Chapter 6

Conclusion

In this chapter, we first give a per-chapter overview of the conclusions and contributions of our work. Then, we reflect on these contributions and outline the implications for authors with a preference for anonymity, authorship attributers, and future work on authorship attribution in adversarial settings.

6.1 Overview

Chapter 3

- An author's anonymity is affected when an adversary applies an authorship attribution method, such as the pruned features method, to her text.
- The R@1 measure for authorship attribution is not a sufficient measure of anonymity. R@n for $n = 1, 2, ..., N$, where N is the number of possible authors, provides a better understanding of an author's anonymity.
- We report accurate measurements of the performance of authorship attribution methods, which improve our understanding of author anonymity.
- Homogeneity in the set of possible authors with respect to some author characteristics affects recall. We investigated the effect of the following three author characteristics:

  Gender Attributing texts when all possible authors are male is slightly more difficult than when the possible authors are female or gender mixed. This is visible in the lower n when measuring R@n. There is no clear difference in recall between attributing texts for female-only authors or authors from a mixed gender group.

  Occupation Occupation also has an effect on recall. We find that attributing authorship of texts within the 'Publishing' group is more difficult than in any other group. Attributing texts within the 'Internet' and to a lesser extent 'Education' groups is easier. A mixed-occupation group of authors shows the average recall of the recalls of occupations that comprise it.

  Age We find that within the teenage group of authors, authorship attribution recall is lowest, and within the 30+ age group it is highest. Within the age group twenties, authorship attribution recall is similar to that of the mixed age group and lies between the recall for the teen and 30+ age groups.

- We propose a new feature set and show that it outperforms an existing feature set on which our feature set is based.
- We improve the state of the art in authorship attribution techniques for one specific data set.

Chapter 4

- An author's anonymity is affected when an adversary applies computational stylometry to her text, even if she implements the obfuscation or the imitation attack.
- We report accurate measurements of the performance of authorship attribution methods against authors implementing the obfuscation and imitation attack, which improve our understanding of author anonymity.
- The obfuscation and imitation attacks greatly reduce the performance of authorship attribution methods measured in R@n, for all n, and are therefore both effective methods against authorship attribution.
- The obfuscation attack can be identified with high accuracy.
- We develop a method for (partial) de-obfuscation of obfuscated texts and show its effectiveness, thereby proving the following:
    1. The obfuscation attack is implemented in a similar way by different authors; obfuscation behavior follows a pattern.
    2. The obfuscation attack can be reversed to some extend.
- We improve the state of the art in authorship attribution techniques against the obfuscation and the imitation attacks.
- Although the recall against the obfuscation and imitation attacks is similar, we show that the imitation attack has better anonymizing properties than the obfuscation attack. This is because the obfuscation attack can both be identified and reversed, while there are no publicly known methods to identify or reverse the imitation attack.
- We report differences in writing style between obfuscated and non-adversarial texts.
- We show that Satoshi Nakamoto is likely not a single author who implemented the obfuscation attack.

Chapter 5

- We show that language models can be used for gender attribution in adversarial settings and we report exactly what kind of language model we use for this task, thereby improving current understanding of language models for gender attribution.
- We show that a feature based method for gender attribution has similar accuracy as a language modelling method for gender attribution.
- We prove that language models can be used for authorship attribution and report R@n measurements. A language modelling approach to authorship attribution can be similarly effective as a pruned features approach, and performs better when measuring R@n for lower n, but worse at higher n, compared to a pruned features approach.
- We show there are various promising research directions to make further use of language models for authorship attribution in adversarial settings.

6.2 Reflection

In this work, we investigated the use of stylometric methods for authorship attribution in adversarial settings. We showed that **authorship attribution can be performed with high recall when the author does not employ an adversarial writing style.** We found that homogeneity in the set of possible authors does have some effect on the performance of authorship attribution, that the

recall is influenced by the type of group, but that the average recall against different homogeneous groups of authors is similar to that of a heterogeneous group.
We showed that the employment of the obfuscation or imitation attack by an author greatly affects the recall of different authorship attribution methods, including our own state-of-the-art authorship attribution method. Although at the time of writing the R@n against the obfuscation and imitation attacks are very similar, **we believe that the imitation attack, if correctly implemented, has better anonymizing properties than the obfuscation attack.** This is because the obfuscation attack can be detected, which in itself can be disadvantages to the anonymity of an author, and because we successfully developed a method for text de-obfuscation while showing there are other, potentially more accurate, methods for text de-obfuscation. We also explained that a similar approach for the identification of the imitation attack and for text de-imitation are much more difficult to develop because there exist many potential authors to imitate.
We showed that language models, in the context of stylometry first applied to gender attribution, can be successfully used for authorship attribution in an adversarial setting too. We believe there are opportunities to improve the performance, and broaden the use of language models for authorship attribution, including using a language model with better smoothing; identification of the obfuscation attack; text de-obfuscation using language models; and a smart interleaving of a language modelling method for authorship attribution, which has higher R@n for small n, and a pruned-features based method, which has higher R@n for large n.

Because the imitation attack has better anonymizing properties than the obfuscation attack, we believe that more research into the attribution of imitation texts and imitation counter-attacks should be prioritized in future research into authorship attribution under adversarial conditions. We also believe that the first step for future research into the attribution of imitation texts is to generate or aggregate more imitation text data. Currently, available data for research into the imitation attack only contains samples of people imitating the same author. To be able to identify imitation behavior that generalizes over different targeted authors, to better understand the anonymizing properties of the imitation attack, and to identify what author (both imitating and imitated) characteristics make an imitation attack successfull, text samples targeting a larger variety of authors is needed. Also, if there would be data of individuals targeting different authors for imitation, our proposed imitation counter-attack could be tested. While the imitation attack forms the greatest danger to the interests of an authorship attributer, the obfuscation attack is also a thread to those same interests. Additional data for the obfuscation attack would allow more research into this attack and a better understanding of the anonymizing properties of the obfuscation attack.
While we have focussed on authorship attribution and, to a lesser extend, gender attribution, there are other characteristics that may be attributed to an author. Other characteristics could include age, occupation, level of education, and native language (family). These characteristics, if attributed with high recall or accuracy, may aid an investigation into the identity of an anonymous author by reducing the anonymity set size that the author resides in, or could be used as an intermediate step of an author attribution process.

Appendices

Appendix A

Texts by Satoshi Nakamoto

Satoshi Nakamoto text 1

I wanted to let you know, I just released the full implementation of the paper I sent you a few months ago, Bitcoin v0.1. Details, download and screenshots are at www.bitcoin.org
I think it achieves nearly all the goals you set out to solve in your b-money paper.
The system is entirely decentralized, without any server or trusted parties. The network infrastructure can support a full range of escrow transactions and contracts, but for now the focus is on the basics of money and transactions.

Satoshi Nakamoto text 2

I've developed a new open source P2P e-cash system called Bitcoin. It's completely decentralized, with no central server or trusted parties, because everything is based on crypto proof instead of trust. Give it a try, or take a look at the screenshots and design paper:
The root problem with conventional currency is all the trust that's required to make it work. The central bank must be trusted not to debase the currency, but the history of fiat currencies is full of breaches of that trust. Banks must be trusted to hold our money and transfer it electronically, but they lend it out in waves of credit bubbles with barely a fraction in reserve. We have to trust them with our privacy, trust them not to let identity thieves drain our accounts. Their massive overhead costs make micropayments impossible.
A generation ago, multi-user time-sharing computer systems had a similar problem. Before strong encryption, users had to rely on password protection to secure their files, placing trust in the system administrator to keep their information private. Privacy could always be overridden by the admin based on his judgment call weighing the principle of privacy against other concerns, or at the behest of his superiors. Then strong encryption became available to the masses, and trust was no longer required. Data could be secured in a way that was physically impossible for others to access, no matter for what reason, no matter how good the excuse, no matter what.
It's time we had the same thing for money. With e-currency based on cryptographic proof, without the need to trust a third party middleman, money can be secure and transactions effortless.
One of the fundamental building blocks for such a system is digital signatures. A digital coin contains the public key of its owner. To transfer it, the owner signs the coin together with the public key of the next owner. Anyone can check the signatures to verify the chain of ownership. It works well to secure ownership, but leaves one big problem unsolved: double-spending. Any owner could try to re-spend an already spent coin by signing it again to another owner. The usual solution is for a trusted company with a central database to check for double-spending, but that just gets back to the trust model. In its central position, the company can override the users, and the fees needed to support the company make micropayments impractical.
Bitcoin's solution is to use a peer-to-peer network to check for double-spending. In a nutshell, the

network works like a distributed timestamp server, stamping the first transaction to spend a coin. It takes advantage of the nature of information being easy to spread but hard to stifle. For details on how it works, see the design paper at website.
The result is a distributed system with no single point of failure. Users hold the crypto keys to their own money and transact directly with each other, with the help of the P2P network to check for double-spending.

Satoshi Nakamoto text 3

In the absence of a market to establish the price (of bitcoin, estimates) based on production cost is a good guess and a helpful service (thanks). The price of any commodity tends to gravitate toward the production cost. If the price is below cost, then production slows down. If the price is above cost, profit can be made by generating and selling more. At the same time, the increased production would increase the difficulty, pushing the cost of generating towards the price. In later years, when new coin generation is a small percentage of the existing supply, market price will dictate the cost of production more than the other way around.
When someone tries to buy all the worlds supply of a scarce asset, the more they buy the higher the price goes. At some point, it gets too expensive for them to buy any more. Its great for the people who owned it beforehand because they get to sell it to the corner at crazy high prices. As the price keeps going up and up, some people keep holding out for yet higher prices and refuse to sell. The Hunt brothers famously bankrupted themselves trying to corner the silver market in 1979.
I believe itll be possible for a payment processing company to provide as a service the rapid distribution of transactions with good-enough checking in something like 10 seconds or less.
While I dont think Bitcoin is practical for smaller micropayments right now, it will eventually be as storage and bandwidth costs continue to fall. Whatever size micropayments you need will eventually be practical. I think in 5 or 10 years, the bandwidth and storage will seem trivial.
Creating an account on a website is a lot easier than installing and learning to use software, and a more familiar way of doing it for most people. The only disadvantage is that you have to trust the site, but thats fine for pocket change amounts for micropayments and misc expenses. Its an easy way to get started and if you get larger amounts then you can upgrade to the actual bitcoin software.

Satoshi Nakamoto text 4

It is a global distributed database, with additions to the database by consent of the majority, based on a set of rules they follow:
- Whenever someone finds proof-of-work to generate a block, they get some new coins
- The proof-of-work difficulty is adjusted every two weeks to target an average of 6 blocks per hour (for the whole network)
- The coins given per block is cut in half every 4 years

You could say coins are issued by the majority. They are issued in a limited, predetermined amount.
As an example, if there are 1000 nodes, and 6 get coins each hour, it would likely take a week before you get anything.
To Sepp's question, indeed there is nobody to act as central bank or federal reserve to adjust the money supply as the population of users grows. That would have required a trusted party to determine the value, because I don't know a way for software to know the real world value of things. If there was some clever way, or if we wanted to trust someone to actively manage the money supply to peg it to something, the rules could have been programmed for that.
In this sense, it's more typical of a precious metal. Instead of the supply changing to keep the value the same, the supply is predetermined and the value changes. As the number of users grows, the value per coin increases. It has the potential for a positive feedback loop; as users increase, the value goes up, which could attract more users to take advantage of the increasing value.

Satoshi Nakamoto text 5

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non- reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone. They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

Satoshi Nakamoto text 6

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non- reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

Bibliography

[1] Bitcoin: A peer-to-peer electronic cash system. Accessed: 2015-07-06.

[2] Corpus linguistics and stylometry. `http://pers-www.wlv.ac.uk/$\sim$in4326/papers/$U50.pdf`. Accessed: 2015-07-06.

[3] Dear garry. i've decided to end it all: The full stop that trapped a killer. `http://www.dailymail.co.uk/news/article-1197187/Dear-Garry-Ive-decided-end-The-stop-trapped-killer.html`. Accessed: 2015-07-06.

[4] I2p. `https://en.wikipedia.org/wiki/I2P`. Accessed: 2015-07-06.

[5] Im belle de jour. Accessed: 2015-07-06.

[6] John twelve hawks the official site. Accessed: 2015-07-06.

[7] Jrandom's announcement. `https://geti2p.net/en/misc/jrandom-awol`. Accessed: 2015-07-06.

[8] Linguistic analysis says newsweek named the wrong man as bitcoin's creator. Accessed: 2015-07-06.

[9] Mr anonymous: "tous les éditeurs avaient rejeté mon manuscrit". Accessed: 2015-07-06.

[10] Occams razor: who is most likely to be satoshi nakamoto? Accessed: 2015-07-06.

[11] The secret social democrat. Accessed: 2015-07-06.

[12] Technology assessment for the state of the art biometrics excellence roadmap. `http://www.biometriccoe.gov/SABER/index.htm`. Accessed: 2015-07-06.

[13] Who is satoshi nakamoto? Accessed: 2015-07-06.

[14] *Authorship Attribution.* now - the essence of knowledge, 2008.

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

[16] Michael Brennan, Sadia Afroz, and Rachel Greenstadt. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Trans. Inf. Syst. Secur. 15, 3*, 2012.

[17] Michael Brennan and Rachel Greenstadt. Practical attacks against authorship recognition techniques.

[18] Aylin Caliskan-Islam, Richard Harang, Andrew Liu, Arvind Narayanan, Clare Voss, Fabian Yamaguchi, and Rachel Greenstadt. De-anonymizing programmers via code stylometry. Technical report, Drexel university and Princeton university and University of Goettingen, 2014.

[19] E Kreyszig. *Applied Mathematics.* Wiley Press, 1979.

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

[21] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine 2*, 1901.

[22] Stuart J. Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Pearson, 2003.

[23] K. G. Ruchita Sarawgi and Y. Choi. Tracing stylometric evidence beyond topic and genre. *Proceedings of the 15th Conference on Computational Natural Language Learning*, 2011.

[24] J. Schler, M. Koppel, S. Argamon, and J. Pennebaker. Effects of age and gender on blogging. *Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*, 2006.