Deception in Authorship Attribution

A Thesis

Submitted to the Faculty

of

Drexel University

by

Sadia Afroz

in partial fulfillment of the

requirements for the degree

of

PhD in Computer Science

December 2013

Acknowledgements

At first, I want to thank my amazing adviser Rachel Greenstadt. I truly mean it when I say that it would have been difficult for me to finish my PhD if she were not my adviser. Specially in the beginning of my graduate life when I was struggling to express myself in English as well as having hard time coping with graduate life and life in the USA in general, she was always the first person to support me. I can probably write a whole book about the millions of ways Rachel inspired me, made my life easier and helped me improve. I am also grateful to my present and past committee members and mentors– Yuanfang Cai, Spiros Mancoridis, Dario Salvucci, Ali Shokofandeh and Jennifer Rode from Drexel University, Knarig Arabshian from Bell Labs, J. D. Tygar and Anthony Joseph from UC Berkeley and Ling Huang from Intel. Special thanks to J. D. Tygar and Anthony Joseph for giving me the opportunity to spend two summers at UC Berkeley. I also want to thank Damon McCoy for introducing me to the underground cybercrime research.

My labmates from PSAL and UC Berkeley made the process of doing research and writing papers much more fun than it would have been otherwise. Michael Brennan has been a great mentor to me, especially in the early days of PSAL having Mike as a labmate was very helpful. My other labmates – Aylin Çalışkan İslam, Ariel Stolerman, Andrew McDonald, Alex Kantchelian (UC Berkeley), Vaibhav Garg, and Rebekah Overdorf – have an enormous impact on my productivity. Seeing them doing useful research right after they started in the lab made me seriously reevaluate my research productivity. I will always cherish the excitement and satisfaction I felt after our last minute paper submissions. My internship at Bell Labs is an important part of my graduate life, as there I got the opportunity to work on industry research and discuss my research with many other students from different universities. I would especially thank Ranjit Kumaresan for several discussions on security and research in general that broadened my research perspectives.

I once read that it takes a village to raise a child. It is definitely appropriate to say that it takes the whole research community to raise a graduate student. This is an exciting time to work in security and privacy and I want to thank the ever enthusiastic crowd of Privacy Enhancing

Technology Symposium (PETS) and Chaos Communication Congress (CCC) for making me see the importance of research in this area.

I am fortunate to be a part of a family that values hard-work and perseverance over anything else. Providing higher education to a girl is difficult in Bangladesh and my parents often had to go the extra mile to provide me the best possible opportunities. I thank my brother Maswood for our frequent skype calls and Alimoor and my in-laws for their support. I am also grateful for having most of my undergraduate classmates in almost all the major cities in the USA. I am specially grateful to Shamia Hoque, Tanvir Farouk, Israt Fatema and Tomal bhai for opening their door to us in time of need.

Lastly, I want to thank all the people I met in the USA who were surprisingly generous to me— Kim, the food truck owner beside my lab, for talking with me every morning and also giving me free coffees and drinks; that woman at the train station for helping me lift my heavy luggage; Nadia, from Bell Labs food counter, for giving me free french fries many many times; that woman at Macy's for giving me free gift card; my roommates, for making cookies for me— and many others whose warm hearts brightened my days. It is the unexpected generosity of random strangers that made me feel at home in this foreign country.

Dedications

To Masrura Chowdhury and Mostafa Roushan Ali, my parents and role models.

Table of Contents

List of Figures

Abstract

Deception in Authorship Attribution

Sadia Afroz
Advisor: Rachel Greenstadt, PhD

In digital forensics, questions often arise about the authors of documents: their identity, demographic background, and whether they can be linked to other documents. The field of stylometry uses linguistic features and machine learning techniques to answer these questions. While stylometry techniques can identify authors with high accuracy in non-adversarial scenarios, their accuracy is reduced to random guessing when faced with authors who intentionally obfuscate their writing style or attempt to imitate that of another author. Most authorship attribution methods were not evaluated in challenging real-world datasets with foreign language and unconventional spelling (e.g. l33tsp3ak). In this thesis we explore the performance of authorship attribution methods in adversarial settings where authors take measures to hide their identity by changing their writing style and by creating multiple identities. We show that using a large feature set, it is possible to distinguish regular documents from deceptive documents with high accuracy and present an analysis of linguistic features that can be modified to hide writing style. We show how to adapt regular authorship attribution to difficult datasets such as leaked underground forum and present a method for detecting multiple identities of authors. We demonstrate the utility of our approach with a case study that includes applying our technique to an underground forum and manual analysis to validate the results, enabling the discovery of previously undetected multiple accounts.

1. Introduction

In 2013, writing style analysis revealed that J. K. Rowling was the author of "The Cuckoo's Calling" which was published under the pen name Robert Galbraith. Two linguists Peter Millican and Patrick Juola found that the distribution of word lengths in Rowling's and Galbraith's writing was very similar[1]. Previously, convicted terrorists were identified from the manifestation of their attacks. For example, the Unabomber (University and Airlines bomber) was identified as Ted Kaczynski from his anonymously published document "Unabomber Manifesto"[2]. Writing style analysis was also used in court cases to resolve identity fraud, e.g., the alleged email exchange of Mark Zuckerburg's with Paul Ceglia, a salesman from upstate New York who claimed that half of Facebook actually belongs to him, was found as fake after writing style analysis[3].

Writing style is a marker of individual identity which can be used to identify the author of a document. While these techniques existed before computers and artificial intelligence, the field is currently dominated by AI techniques such as neural networks and statistical pattern recognition. State-of-the-art stylometry approaches can identify individuals in sets of 100 authors with over 90% accuracy [2] and can be scaled to 10,000 authors [16] and even 100,000 authors [25]. These developments have made authorship attribution a necessary new ground for research in privacy and security. It has also been recognized by the law enforcement. The 2009 Technology Assessment for the State of the Art Biometrics Excellence Roadmap (SABER) commissioned by the FBI stated that, "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 [40]."

Although current authorship attribution techniques are highly accurate in detecting authors, these techniques can be deceived easily by changing writing style and by creating multiple identities

[1] `http://languagelog.ldc.upenn.edu/nll/?p=5315`
[2] `http://cyber.eserver.org/unabom.txt`
[3] `http://techland.time.com/2011/06/03/how-to-write-like-mark-zuckerberg/?hpt=te_bn1`

per author. Deception in writing style has been used recently to spread hoax and false information on the web. For example, an American blogger Thomas MacMaster wrote a sociopolitical blog "A Gay Girl in Damascus" where he posed as a Syrian homosexual woman Amina Arraf and wrote about Syrian political and social issues. To write the blog Thomas developed a new writing style which is different from his regular writing style [4]. Another incident happened in June 2012 when a Wikileaks supporter imitated Bill Keller's writing style and wrote an op-ed which was considered as true by millions including the editor of the Times' Bits technology blog[5].

Creating multiple accounts per author is more common than changing writing style, e.g., many people have multiple email addresses, accounts on different sites (e.g. Facebook, Twitter, G+) and blogs. This is problematic because the supervised authorship algorithm uses documents of known authors to model each author in the system. If any two authors were the same but were represented under different aliases, the classifier may learn a faulty model of the original author. This could increase the number of misclassifications during evaluation. The problem would be worse if the sample documents for one identity are more general than the sample documents for the other. Grouping multiple identities of an author is a powerful ability which is impossible to do using only the supervised authorship attribution methods.

In this thesis, we first show that regular authorship attribution methods fail in challenging real world datasets where authors take measures to hide their identity and then present methods to detect these deceptions by analyzing writing style. We propose methods for identifying imitated and obfuscated documents and for grouping multiple identities of an author and evaluate both of the methods using real world multilingual datasets. This is useful both for forensics analysts, as otherwise they have no way of knowing whether the suspect is deceptive or not, and for internet activists as well, who may appear in these suspect lists and be falsely accused of writing certain documents. We also present *Anonymouth* that can help an author change his writing style.

[4] http://www.telegraph.co.uk/news/worldnews/middleeast/syria/8572884/A-Gay-Girl-in-Damascus-how-the-hoax-unfolded.html

[5] `http://www.cnn.com/2012/07/30/tech/web/fake-nyt-editorial`

1.1 Problem statement

In supervised authorship attribution given a document D and a set of unique authors $\mathcal{A} = \{A_1, ..., A_n\}$ and their written documents determines who among the authors in $\mathcal{A}$ wrote D.

This thesis asks the following two questions:

1. Is D written in a changed writing style?

2. Are the authors in $\mathcal{A}$ unique? Group all the identities of an author in $\mathcal{A}$.

1.2 Prior work

Stylometry. The classic example in the field of stylometry is the Federalist Papers. 85 papers were published anonymously in the late 18th century to persuade the people of New York to ratify the American Constitution. The authorship of 12 of these papers was heavily contested [26]. To discover who wrote the unknown papers, researchers have analyzed the writing style of the known authors and compared it to that of the papers with unknown authorship. The features used to determine writing styles have been quite varied. Original attempts used the length of words, whereas later attempts used pairs of words, vocabulary usage, sentence structure, function words, and so on. Most studies show the author was James Madison. Recently, Artificial Intelligence has been embraced in the field of stylometry, leading to more robust classifiers using machine learning and other AI techniques such as neural networks and statistical pattern recognition [15, 38]. State-of-the-art stylometry approaches can identify individuals in sets of 50 authors with over 90% accuracy [2], and even scaled to over 100,000 authors [25].

Stylistic deception. The current authorship recognition methods are built on the assumption that authors do not intentionally change their writing style. These methods fail to detect authorship when this assumption does not hold[7, 18]. Brennan et al.[7] showed that the accuracy of detecting authorship decreases to random guessing when one author imitates another author or obfuscates his writing style. Other research looked at imitation of professional authors. Somers [36] compared the work of Gilbert Adair's literary imitation of Lewis Carroll's Alice in Wonderland, and found mixed results.

The area of detecting deception in writing style has not be explored much. Kacmarcik and Gamon explored detecting obfuscation by first determining the most effective function words for discriminating between text written by Hamilton and Madison, then modifying the feature vectors to make the documents appear authored by the same person. The obfuscation was then detected with a technique proposed by Koppel and Scher, "unmasking," that uses a series of SVM classifiers where each iteration of classification removes the most heavily weighted features. The hypothesis they put forward (validated by both Koppel and Scher [16] and Kacmarcik and Gamon [21]) is that as features are removed, the classifier's accuracy will slowly decline when comparing two texts from different authors, but accuracy will quickly drop off when the same is done for two texts by the same author (where one has been modified). It is the quick decline in accuracy that shows there is a deeper similarity between the two authors and indicates the unknown document has most likely been modified. This work has some significant limitations. The experiments were performed on modified feature vectors, not on modified documents or original documents designed with obfuscation in mind. Further, the experiments were limited to only two authors, Hamilton and Madison, and on the Federalist Papers data set. It is unclear whether the results generalize to actual documents, larger author sets and modern data sets.

We analyzed the differences between the control and deceptive passages on a feature-by-feature basis and used this analysis to determine which features authors often modify when hiding their style, designing a classifier that works on actual, modern documents with modified text, not just feature vectors.

Multiple identities detection. A few previous works explored the question of identifying multiple identities of an author. The Writeprints method can be used to detect similarity between two authors by measuring distance between their "writeprints." Qian et al.'s method, called "Learning by similarity," learns in the similarity space by creating a training set of similar and dissimilar documents [31] and comparing the distances between them. This method was evaluated using users on Amazon book reviews. Almishari et al. [4] also used a similar distance-based approach using reviews from yelp.com to find duplicate authors. Koppel et al. [22] used a feature subsampling approach to detect whether two documents are written by the same author. But all of these methods were evaluated by creating *artificial* multiple identities per author by splitting a single author into two parts. In our experiments we noticed that identifying users writing about similar topics is easier than when they write about different topics. We evaluated our method on a real world blog dataset where users themselves created different identities in different blogs and in many cases different blogs by the same user were not about the same topic.

Detecting Fraudulent Accounts. Perito et al. [28] showed that most users use similar usernames for their accounts in different sites, e.g., daniele.perito and d.perito. Thus different accounts of a user can be tracked by just using usernames. This does not hold when the users are deliberately trying to hide their identity, which is often the case in the underground forums.Usernames and other account information and behavior in the social network have often used to identify Sybil/spam accounts [11, 10, 5]. Our goal is different from these works as we are trying to identify duplicate accounts of highly active users, who would be considered as *honest* users in previous fraud detection papers. For example, these users are highly connected with other users in the forum, unlike spam/sybil accounts. Their account information (usernames, email addresses) are similar to spam accounts with mixed language, special characters and disposable email accounts, however, these properties hold for most users in these forums, even the ones who are not creating multiple identities.

1.3 Statement of thesis

This thesis argues the following statement:

Deception in authorship attribution can be detected.

We propose two methods to mitigate deception in authorship attribution. For detecting writing style change we identified a set of discriminating features that distinguish deceptive writing from regular writing. After determining these features, supervised learning techniques were used to train and generate classifiers to classify new writing samples. For detecting multiple accounts we use *Doppelgänger Finder* where a classifier decides whether A_i and A_j are the same based on the probability of A_j's document attributed to A_i and probability of A_i's document attributed to A_j.

1.4 Summary of contributions

Contributions of this thesis are:

1. **Authorship attribution in real-world dataset:** We perform authorship attribution in multiple real-world datasets and show the shortcomings of current methods. We propose ways to cope authorship attribution in challenging scenarios, e.g., in multilingual settings and short colloquial texts. Although some language-agnostic authorship attribution methods are available [20, 16] for this task, most of the highly accurate attribution methods [25, 1] are language specific for standard English. We show that by using language-specific function words and parts-of-speech taggers, our authorship attribution method provides high accuracy even with over 1000 authors in difficult, foreign language texts.

2. **Detecting stylistic deception:** We show that whether or not an author has changed his writing style can be detected. We show when authors deliberately change their writing style they mostly change content words, while function words remain unchanged. Our contributions include a general method for distinguishing stylistic deception from regular writing, an analysis of long-term versus short-term deception, and the discovery that stylistic deception shares similar features with lying-type deception (and can be identified using the linguistic features used in lying detection).

   We perform analysis on the Brennan-Greenstadt adversarial dataset and a similar dataset collected using Amazon Mechanical Turk (AMT)[6]. We show that linguistic cues that can detect stylistic deception in the Extended-Brennan-Greenstadt adversarial dataset can detect

[6] https://mturk.amazon.com

indication of masking in the documents collected from the Ernest Hemingway[7] and William Faulker imitation contests[8]. We also show how long-term deceptions such as the blog posts from "A Gay Girl in Damascus" are different from these short-term deceptions. We found these deceptions to be more robust to our classifier but more vulnerable to traditional stylometry techniques.

3. **Detecting multiple identities:** We propose and evaluate a method *Doppelgänger Finder* to identify multiple identities of an authors. Our method is evaluated on a real world English blog dataset where the authors themselves created different blogs and includes cases where the blogs of one author are not of similar topics. Our approach evaluates all pairs of a set of authors for duplicate identities and return a list of potential pairs, ordered by probability. This list can be used by a forum analyst to quickly identify interesting multiple identities. We validated our algorithm on real-world blogs using multiple separate blogs per author and using multiple accounts of members in different underground forums.

   Using *Doppelgänger Finder* on a German carding forum *Carders*, we show how to discover and group unknown identities in cases when ground truth data is unavailable. We discovered at least 10 new author pairs (and an additional 3 probable pairs) automatically which would have been hard to discover without time consuming manual analysis. These pairs are typically high value identities—in one case we found a user who created such identities for sale to other users on the forum.

1.5 Thesis organization

In Chapter 2 we discuss current authorship attribution methods and present two case studies of regular authorship attribution in real world datasets. We present a method for detecting stylistic deception in Chapter 3 and discuss which features an author changes to change his writing style. The multiple identity detection approach is discussed in Chapter 4, with detailed analysis of underground

[7] http://en.wikipedia.org/wiki/International_Imitation_Hemingway_Competition
[8] http://en.wikipedia.org/wiki/Faux_Faulkner_contest

forum. In Chapter 5 we present and evaluate an authorship anonymization tool Anonymouth. Chapter 6 discusses open questions in the area of authorship attribution, which is followed by conclusion in Chapter 7.

2. Authorship Attribution in the wild

In this section we will first discuss regular authorship recognition techniques and then show how these techniques are perform in recognizing authorship in adversarial situations.

2.1 Background

The basic intuition behind authorship attribution is that everybody has unique writing style Language is learned in an individual level, thus everybody learns language differently. This introduces many idiosyncrasies in the written language of an author. For example, some people use "though" whereas others use "although." There are also regional differences in the language. For example, the spelling of the word "colour" is typical in the United Kingdom and most of the British commonwealth, but very uncommon in the United State. Previous researchers denoted the individual stylistic variation as "authorial fingerprint"[19, 2] or "human stylome"[39] to refer to a specific set of measurable traits that can be used to uniquely identify an author.

Approach. Most research considers *supervised* authorship attribution problem that given a document D and a set of authors $\mathcal{A} = \{A_1, ..., A_n\}$ determines who among the authors in $\mathcal{A}$ wrote D. The authorship attribution algorithm has two steps: training and testing. During training, the algorithm trains a classifier using the extracted features, defined in feature set F, from the sample documents of the authors in $\mathcal{A}$. In the testing step, it extracts features predefined in F from D and determines the probability of each author in $\mathcal{A}$ of being the author of D. It considers an author A_{max} to be the author of D if the probability of A_{max} being the author of D, $Pr(A_{max}\ wrote\ D)$, is the highest among all $Pr(A_i\ wrote\ D), i = 1, 2, ...n$.

K-attribution is the relaxed version of authorship attribution. Regular authorship attribution outputs only one author with the highest probability as the author of a given document D. The k-attribution outputs k top authors, ranked by their corresponding probabilities, $Pr(A_i\ wrote\ D)$, where $i = 1, 2, ...k$ and $k \leq n$.

2.2 Case study 1: Stylistic Deception

***This work was completed by Michael Brennan, with support from Sadia Afroz.*

In this section, we show how regular stylometry methods work when authors change their writing style. We developed three methods of circumvention against stylometry techniques in the form of obfuscation, imitation and machine translation passages. Two of these, obfuscation and imitation, were manually written by human subjects. These passages were very effective at circumventing attempts at authorship recognition. Machine Translation passages are automated attempts at obfuscation utilizing machine translation services. These passages were not sufficient in obfuscating the identity of an author.

Obfuscation

In the obfuscation approach the author attempts to write a document in such a way that their personal writing style will not be recognized. There is no guidance for how to do this and there is no specific target for the writing sample. An ideal obfuscated document would be difficult to attribute to any author. For our study, however, we only look at whether or not it successfully deters recognition of the true author.

Imitation

The imitation approach is when an author attempts to write a document such that their writing style will be recognized as that of another specific author. The target is decided upon before a document is written and success is measured both by how successful the document is in deterring authorship recognition systems and how successful it is in imitating the target author. This could also be thought of as a “framing” attack.

Machine Translation

The machine translation approach translates an unmodified passage written in english to another language, or to two other languages, and then back to english. The hypothesis was that this would sufficiently alter the writing style and obfuscate the identity of the original author. We did not find this to be the case.

We studied this problem through a variety of translation services and languages. We measured the effectiveness of the translation as an automated method as well as the accuracy of the translation in producing a comprehensible, coherent obfuscation passages.

We performed three language experiments in addition to the English baseline. In all cases the original and final language were English. We performed single step translations from English to German and back to English as well as English to Japanese and back to English. We then performed two step translations from English to German to Japanese and then back to English. German was chosen for its linguistic similarities to English and Japanese for its differences.

The two machine translation services we compared were Google Translate[1] and Bing Translator[2]. Both services are free and based on statistical machine translation.

2.2.1 The Drexel-AMT and Brennan-Greenstadt Corpora

We have published two freely available research corpora. The first is the Brennan-Greenstadt corpus, which is based on a survey conducted through Drexel University and contains 12 authors who volunteered their time and were not compensated for their efforts. This corpus was the basis for our original work on adversarial stylometry [7]. The second is the Drexel Amazon Mechanical Turk (Drexel-AMT) corpus containing 45 authors solicited through the Amazon Mechanical Turk platform. Submissions were vetted against a series of guidelines to ensure the quality of the content, as described below.

[1] `http://translate.google.com`

[2] `http://www.microsofttranslator.com`

Brennan-Greenstadt Corpus

Participants for the Brennan-Greenstadt corpus were solicited through classes at Drexel University, colleagues, and other personal relationships. This provided us with submissions from 12 authors. The Brennan-Greenstadt corpus used an earlier version of the survey which had two relaxed requirements. Authors were only required to submit 5000 words of pre-existing writing and they were not required to fill out a demographic survey.

While this corpus was sufficient for preliminary results presented in earlier work [7], we desired a more robust corpus in order to confirm our original findings in a larger author space with a greater diversity of writers and tweaked survey requirements.

Drexel-AMT Corpus

We utilized the Amazon Mechanical Turk (AMT) platform to create a large and diverse corpus that could be used for more robust analysis.

Amazon Mechanical Turk[3] is a platform that provides access to a large and diverse population that is willing to perform human intelligence tasks. Participants choose tasks that they would like to complete in exchange for a sum of money decided by a task creator.

Submission quality is a serious consideration when using the AMT platform as the completion of a task does not necessarily indicate that the worker has followed the directions and completed it correctly. In order to ensure that the submissions were acceptable we reviewed every submission and judged their acceptability by scrutinizing them according to the guidelines and requirements listed on the submission form. We only removed authors from the data set who did not adhere to the directions of the survey. We did not remove authors because of the quality of their writing, demographic information, or anything other than their ability to follow directions.

In addition to the existing requirements we published four guidelines that submissions should adhere to:

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

1. The submitted pre-existing writing to be "scholarly" in nature (i.e: a persuasive piece, opinion paper, research paper, journals, etc.).

2. Anything that is not the writing content of the work should be removed (i.e: citations, urls, section headings, editing notes, etc.).

3. The papers/samples should have a minimal amount of dialogue/quotations.

4. Please refrain from submitting samples of less than 500 words, laboratory and other overly scientific reports, Q&A style samples such as exams, and anything written in another person's style.

As an added incentive for authors to take care with their submissions we offered a bonus payment of two dollars on top of an original payment of three dollars if their submission adhered to the quality guidelines. Of the 100 submissions we received, 45 satisfied the requirements of the survey. These 45 submissions make up the Drexel-AMT adversarial stylometry corpus and are the basis of the evaluation for this research.

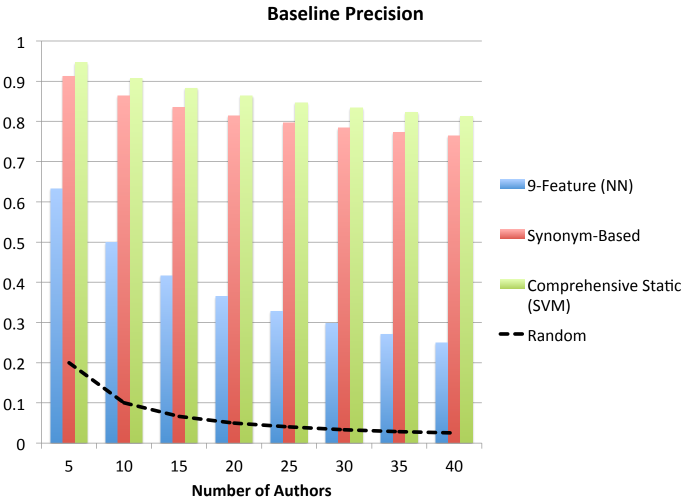


Figure 2.1: Baseline accuracy.

2.3 Case study 2: Underground forums

Our goal in this section is to see how well stylometry works in the challenging setting of underground accounts and adapt stylometric methods to improve performance.

2.3.1 Underground Forums

We analyzed four underground forums: AntiChat (AC), BlackhatWorld (BW), Carders (CC), L33tCrew (LC) (summarized in Table 2.1). For each of these four forums we have a complete SQL dump of their database that includes user registration information, along with public and private messages. Each of these SQL forum dumps has been publicly "leaked" and uploaded to public file downloading sites by unknown parties.

Forum	**Language**	**Date covered**	**Posts**	**Pvt msgs**	**Users**	**Lurkers**
Antichat (AC)	Russian	May 2002-Jun 2010	2160815	194498	41036	15165 (36.96%)
BlackHat (BW)	English	Oct 2005-Mar 2008	65572	20849	8718	4229 (48.5%)
Carders(CC)	German	Feb 2009-Dec 2010	373143	197067	8425	3097(36.76%)
L33tCrew (LC)	German	May 2007-Nov 2009	861459	501915	18834	9306 (46.41%)

Table 2.1: Summary of forums

2.3.2 Forums

This section gives an overview of the forums, in particular, it shows the relationship between a member's rank and his activities in the forum. In all forums, high-ranked members had more posts than low-ranked members. Access to special sections of these forums depends on a member's rank, having the full SQL dump gives us the advantage of seeing the whole forum, which would have been unavailable if we had crawled the forums as an outsider or as a newly joined member. In general, the high-ranked users have more reputation, a longer post history, and consequently more words for our algorithms to analyze.

Antichat

Antichat started in May 2002 and was leaked in June 2010. It is a predominantly Russian language forum with 25871 active users (users with at least one post in the forum). Antichat covers a broad array of underground cybercrime topics from password cracking, stolen online credentials, email spam, search engine optimization (SEO), and underground affiliate programs.

Anybody with a valid email address can join the forum, though access to certain sections of the forum is restricted based on a member's rank. At the time of the leak, there were 8 advanced groups and 8 user ranks in our dataset[4]. A member of level N can access groups at level ≤ N. Admins and moderators have access to the whole forum and grant access to levels 3 to 6 by invitation. At the time of the leak, there were 4 admins and 89 moderators in Antichat.

Members earn ranks based on their reputation which is given by other members of the forum for any post or activity[5]. Initially each member is a *Beginner (Новичок)* [6], a member with at least 50 reputation is *Knowledgeable (Знающий)* and 888 reputation is a *Guru (Гуру)* (all user reputation levels are shown in Table 2.2). A member can also get negative reputation points and can get banned. In our dataset there were 3033 banned members. The top reasons for banning a member are having multiple accounts and violating trade rules.

Rank	**Rep.**	**Members**	**Members with ≥4500 words**
Ламер (Lamer)	-50	646	22
Чайник (Newbie)	-3	340	4
Новичок (Beginner)	0	38279	553
Знающий (Knowledgeable)	50	595	256
Специалист (Specialist)	100	658	413
Эксперт (Expert)	350	271	177
Гуру (Guru)	888	206	153
Античатовец (Antichatian)	5555	1	1

Table 2.2: AntiChat members rank

[4] `http://forum.antichat.ru/thread17259.html`

[5] Member rules are described `https://forum.antichat.ru/thread72984.html`

[6] Translated by Google translator

Antichat has a designated a “Buy, Sell, Exchange” forum for trading. Most of the transactions are in WebMoney[7]. To minimize cheating, Antichat has paid “Guarantors” to guarantee product and service quality[8]. Sellers pay a percentage of the value of one unit of goods/services to the guarantor to verify his product quality. Members are advised not to buy non-guaranteed products. In case of a cheating, a buyer is paid off from the guarantor’s collateral value.

BlackhatWorld

BlackhatWorld is primarily an English speaking forum that focuses on blackhat SEO techniques, started in October 2005 and is still active. At the time of the leak (May 2008) Blackhat had 4489 active members.

Like Antichat, anybody can join the forum and read most public posts. At the time of the leak, a member needed to pay $25 to post in a public thread.[9] A member can have 8 ranks depending on his posting activities and different rights in the forums based on his rank. This rank can be achieved either by being active in the forum for long period or by paying fees. A new member with less than 40 posts is a *Blacknoob* and 40-100 posts is a *Peasant*, both of these ranks do not have access to the “Junior VIP” section of the forum which requires at least 100 posts[10]. The “Junior VIP” section is not indexed by any search engines or visible to any non Jr. VIP members. At the time of the leak, a member could pay $15 to the admin to access this section. A member is considered active after at least 40 posts and 21 days after joining the forum. Member ranks are shown in Table 2.3. The forum also maintains an “Executive VIP” section where membership is invited and a “Shitlist” for members with bad reputations. There were 43 banned members in our dataset. Most of the members in our BlackhatWorld dataset were Blacknoobs.

Currently, only the Junior VIP members can post in the BlackhatWorld marketplace, the “Buy, Sell, Trade” section[11]. In our dataset any member with over 40 posts was allowed to trade. Each

[7] http://www.wmtransfer.com/

[8] https://forum.antichat.ru/thread63165.html

[9] The posting cost is now $30

[10] http://www.blackhatworld.com/blackhat-seo/misc.php?do=vsarules

[11] http://www.blackhatworld.com/blackhat-seo/bhw-marketplace-rules-how-post/387929-marketplace-rules-how-post-updated-no-sales-thread-bumping.html

Rank	**Members**	**Members with ≥4500 words**
Banned Users	43	4
21 days 40 posts	7416	4
Registered Member	248	74
Exclusive V.I.Ps	7	7
Premium Members (PAID/Donated)	191	19
Admins and Moderators	8	8

Table 2.3: Blackhat members rank

post in the marketplace must be approved by an admin or moderator. In our dataset, there were 3 admins and 5 moderators. The major currency of this forum is USD. Paypal and exchange of products are also accepted.

Carders

Carders was a German language forum that specialized in stolen credit cards and other accounts. This forum was started in February 2009 and was leaked and closed in December 2010 [12].

At the time of the leak, Carders had 3 admins and 11 moderators. A regular member can have 9 ranks, but unlike other forums the rank was not dependent only on the number of posts (Table 2.4). Access to different sections of the forum was restricted based on rank. Any member with a verified email can be a *Newbie*. To be a *Full Member* a member needs at least 50 posts. A member had to be at least a *Full Member* to sell tutorials. *VIP Members* were invited by other high-ranked members. To sell products continuously a member needs a *Verified vendor* license which requires at least 50 posts in the forum and 150+ € per month. For certain products, for example, drugs and weapons, the license costs at least 200 €. Carders maintained a “Ripper” thread where any member can report a dishonest trader. A suspected ripper was assigned *Ripper-Verdacht!* title. Misbehaving members, for example, spammers, rippers or members with multiple accounts, were either banned temporarily or permanently depending on the severity of their action. In our dataset, there were 1849 banned members. The majority of the members in our Carders dataset are Newbie.

[12] Details of carders leak at `http://www.exploit-db.com/papers/15823/`

Rank	**Members**	**Members with ≥4500 words**
Nicht registriert (Not registered)	1	0
Email verification	323	1
Newbie	4899	23
Full Member	1296	431
VIP Member	7	6
Verified Vendor	16	6
Admins	14	13
Ripper-Verdacht! (Ripper suspected)	14	7
Time Banned	6	2
Perm Banned	1849	193

Table 2.4: Carders members rank

Other products traded in this forum were cardable shops (shops to monetize stolen cards), proxy servers, anonymous phone numbers, fake shipping and delivery services and drugs. The major currencies of the forum were Ukash[13], PaySafeCard (PSC)[14], and WebMoney.

L33tCrew

Like Carders, L33tCrew was a predominantly carding forum. The forum was started in May 2007 and leaked and closed in Nov 2009. We noticed many users joined Carders after L33tCrew was closed. At the time of the leak, L33tCrew had 9528 active users.

L33tCrew member rank also depended on a member's activity and number of posts. With 15 posts a member was allowed in the base account area. The forum shoutbox, which was used to report minor problems or offtopic issues, is visible to members with at least 40 posts. A member's ranking was based on his activity in the forum (Table 2.5). On top of that, a member could have 2nd and 3rd level rankings. 100–150 posts were needed to be a 2nd level member. Members could rise to 3rd level after "proving" themselves in 2nd level and proving that they had non-public tools, tricks, etc. To prove himself a 2nd level member had to send at least three non-public tools to the admin or moderators.

[13] https://www.ukash.com/

[14] https://www.paysafecard.com/

Rank	**Min. posts**	**Members**	**Members with ≥4500 words**
Newbie	0-30	715	93
Half-Operator	60	158	67
Operator	100	177	121
Higher Levels	150	412	398
Unranked Members	–	16482	679
Banned	–	847	197
Admins	–	11	11
Invited	–	33	8
Vorzeitig in der Handelszone	–	5	2

Table 2.5: L33tCrew members rank

2.3.3 Member overlap

For this experiment, we identified common active users in the forums by matching their email addresses. Here “active” means users with at least one private or public message in a forum. Among the four forums, Carders and L33tCrew had 563 common users based on email addresses, among which 443 were active in Carders and 439 were active in L33tCrew. Common users in other forums are negligible.

2.3.4 Hiding identity

In all of the forums, multiple identities were strictly prohibited. On Carders and Antichat one of main reasons for banning a member is creating multiple identities. We wanted to check whether the users were taking any measures to hide their identities. We found several users were using disposable email addresses (562 in Carders, 364 in L33tCrew) from top well-known disposable email services, e.g. trashmail.com, owlpic.com, 20minutemail.

Carders used an alternative-ego detection tool (AE detector)[15] which saves a cookie of history of ids that log into Carders. Whenever someone logs into more than one account, it sends an automated warning message to forum moderators saying that the forum has been accessed from multiple accounts. The AE detector also warns the corresponding members. We grouped these

[15] `http://www.vbulletin.org/forum/showthread.php?t=107566`

multiple account holders based on whether or not they received these warning messages from the AE detector. We found 400 multiple identity groups with total 1692 members, where group size varies from 2 to 466 accounts (shown in Figure 2.4).

We suspect that the AE detector does not reflect multiple account holders perfectly. There are possible scenarios that would trigger the AE detector, e.g. when two members use a shared device to log into Carders or use a NAT/proxy. The corresponding users in these situations were considered as doppelgängers by the AE detector, which does not reflect the ground truth. Likewise, the AE detector may not catch all the alter egos, as some users may take alternate measures to log in from different sources. These suspicions were supported by our stylometric and manual analyses of Carders posts.

2.3.5 Public and private messages

In a forum a member can send public messages to public threads and private messages to other members. In our dataset we had both the public and private messages of all the members. Public messages are used to advertise/request products or services. In general, public messages are short and often have specific formats. For example, Carders specifies a specific format for public thread titles.

Private messages are used for discussing details of the products and negotiating prices. Sometimes members use their other email, ICQ or Jabber address for finalizing trades.

2.3.6 Authorship Attribution

Our goal in this section is to see how well stylometry works in the challenging setting of underground forums and adapt stylometric methods to improve performance.

2.3.7 Feature extraction

Our feature set contains lexical, syntactic and domain specific features. The lexical features include frequency of n-grams, punctuation and special characters. The synactic features include frequency of language-specific parts-of-speech and function words. In our dataset we used English, German, and Russian parts-of-speech taggers and corresponding function words. For English and German parts-of-speech tagging we used the Stanford log-linear parts-of-speech tagger [37] and for Russian parts-of-speech tagging we used TreeTagger [35] with Russian parameters[16]. Function words or stop words are words with little lexical meaning that serve to express grammatical relationships with other words within the sentence, for example, in English function words are prepositions (to, from, for), and conjunctions (and, but, or). We used German and Russian stop words from Ranks.nl (`http://www.ranks.nl/resources/stopwords.html`) as function words. Similar feature sets have been used before in authorship analysis on English texts [1, 25, 23]. We modified the feature set for the multilingual case by adding language specific features. As the majority of the members use leetspeak in these forums, we used the percentage of leetspeak per document as a feature. Leetspeak (also known as Internet slang) uses combinations of ASCII characters to replace Latin letters, for example, leet is spelled as l33t or 1337. We defined leetspeak as a word with symbols and numbers and used regular expression to identify such words.

Feature	**Count**
Freq. of punctuation (e.g. ‘,’ ‘.’)	Dynamic
Freq of special characters (e.g., ‘@’, ‘%’	Dynamic
Freq. of character ngrams, n =1-3	150
Length of words	Dynamic
Freq. of numbers ngrams, n=1-3	110
Freq. of parts-of-speech ngrams, n=1-3	150
Freq. of word ngrams, n=1-3	150
Freq. of function words, e.g. for, to, the.	Dynamic
Percentage of leetspeak, e.g, l33t, pwn3d	-

Table 2.6: Feature set

We used the JStylo (Section 5) API for feature extraction, augmenting it with leetspeak percentage and the multilingual features for German and Russian.

[16] http://corpus.leeds.ac.uk/mocky/

2.3.8 Classification

We used a linear kernel Support Vector Machine (SVM) with Sequential Minimal Optimization (SMO) [30]. We performed 10-fold cross-validation, that is, our classifier was trained on 90% of the documents (at least 4500 words per author) and tested on the remaining 10% of the documents (at least 500 words per author. This experiment is repeated 10 times, each time randomly taking one 500-word document per author for testing and the rest for training. To evaluate our method's performance we use precision and recall. Here *true positive* for author A means number of times a document written by author A was correctly attributed to author A and *false positive* for author A means number of times a document written any other author was misclassified to author A. We calculate per author precision/recall and take the average to show overall performance.

2.3.9 Removing product data

One of the primary challenges with this dataset is the mixing of conversational discussion with product discussions, e.g., stolen credentials, account information with passwords, and exploit code. This is particularly pronounced in the most active users who represent the majority of the trading activities. As the classifier relies on writing style to determine authorship, it misclassifies when two or more members share similar kinds of product information in their messages. Removing product information from conversation improved our classifier's performance by 10-15%. Identifying product information is also useful for understanding what kind of products are being traded in the forums.

Our product detector is based on two observations: 1) product information usually has repeated patterns, 2) conversation usually has verbs, but product information does not have verbs. To detect products, we first tag all the words in a document with their corresponding parts-of-speech and find sentence structures that are repeated more than a threshold of times. We consider the repeated patterns with no verbs as products and remove these from the documents.

To find repeated patterns, we measured Jaccard distance between each pair of tagged sentences. Due to errors in parts-of-speech tagging, sometimes two similar sentences are tagged with different parts-of-speech. To account for this, we considered two tagged sentences as similar if their distance is less than a threshold. We consider a post as a product post if any pattern is repeated more than three times. Note that our product detector is unsupervised and not specific to any particular kind of product, rather it depends on the structure of product information.

To evaluate our product detector we randomly chose 10,000 public posts from Carders and manually labeled them as product or conversation. 3.12% of the posts contained products. Using a matching threshold of 0.5 and repetition threshold of 3, we can detect 81.73% of the product posts (255 out of 312) with 2.5% false positive rate.[17]

2.3.10 Results

Minimum text requirement for authorship attribution

We trained our classifier with different numbers of training documents per author to see how much text is required to identify an author with sufficient accuracy. We performed this experiment for all the forums studied. In our experiments, accuracy increased as we trained the classifier with more words-per-author. On average, the accuracy did not improve when more than 4500 words-per-author were used in training (Figure 2.5).

Attribution within forums

Many users were removed from the data set due to insufficient text, especially after products and data dumps were removed. Table 2.7 shows the number of authors remaining in each forum and our results for author attribution in each forum which are mostly the high ranked members (section 2.3.2). Results are for the public and private messages respectively. Aside from this, performance on private messages ranged from 77.2% to 84% precision. Recall results were similar, as this is a multi-class rather than a binary decision problem and precision for all authors was averaged (a false positive for one author is a false negative for another author). This is comparable to results on less challenging stylometry problems, such as English language emails and essays [1]. Performance

[17] Note that false positives are not that damaging, since they only result in additional text being removed.

on public messages, which were shorter and less conversational—more like advertising copy—was worse, ranging from 60.3% to 72%. The product detection and changes to the features set we made increased the overall accuracy by 10-15% depending on the setting.

However, it is difficult to compare the performance across different forums due to the differing number of authors in each forum. Figure 2.6 shows the results of k-attribution for $k = 1$ to $k = 10$ where the $k = 1$ case is strict authorship attribution. In this figure we can see that the differences between private and public messages persist in this case and that the accuracy is not greatly affected when the number of authors scale from 50 to the numbers in Table 2.7. Furthermore, this figure shows that the results are best for the Carders forum. The higher accuracy for Carders and L33tCrew may be due to the more focused set of topics on these forums or possibly the German language. Via manual analysis, we noted that the part-of-speech tagger we used for Russian was particularly inaccurate on the Antichat data set. A more accurate part-of-speech tagger might lead to better results on Russian language forums.

Relaxed or k-attribution is helpful in the case where stylometry is used to narrow the set of authors in manual analysis. As we allow the algorithm to return up to 10 authors, we can increase the precision of results returned to 96% in the case of private messages and 90% in the case of public messages.

Forum	**Public**		**Private**	
	Members	**Precision**	**Members**	**Precision**
AntiChat	1459	44.4%	25	84%
Blackhat	81	72%	35	80.7%
Carders	346	60.3%	210	82.8%
L33tCrew	1215	68.8%	479	77.2%

Table 2.7: Author attribution within in a forum.

2.3.11 Importance of features

[18] mfg is an abbreviation of a German greeting “Mit Freundlichen Gruessen” (English: sincerely yours).

[19] German subordinating conjunctions (e.g. weil (because), daß (that), damit (so that))

German forums	**English forums**	**Russian forums**
Char. trigram: mfg [18]	Punctuation: (’)	Char. 1-gram: (ё)
Punctuation: Comma	Punctuation: Comma	Function word: ещё (Trans.: more)
Leetspeak	Foreign words	Punctuation: Dot
Punctuation: Dot	Leetspeak	Char. 3-grams: ени
Char 3-gram:(...)	Function word: i’m	Char. bigrams: (,)
Nouns	Punctuation: Dot	Word-bigrams:что бы (that would)
Uppercase letters	POS-bigram (Noun,)	
Function word: dass (that)	Char. bigram: (,)	
Conjunctions [19]		
Char. 1-gram: ∧		

Table 2.8: Features with highest information gain ratio in different forums

To understand which features were the most important to distinguish authors, we calculated the Information Gain Ratio (IGR) [32] of each feature F_i over the entire dataset:

$$IGR(F_i) = (H(A) - H(A|F_i))/H(F_i) \tag{2.1}$$

where A is a random variable corresponding to an author and H is Shannon entropy.

In all the German, English and Russian language forums punctuation marks (comma, period, consecutive periods) were some of the most important features (shown in Table 2.8). In German and English forums leetspeak percentage was highly ranked. Interestingly, similar features are important across different forums, even though the predominant languages of the forums are different.

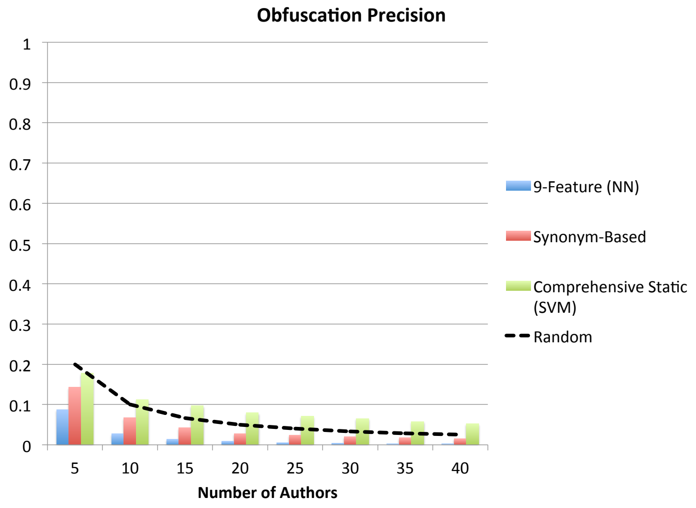


Figure 2.2: Detection of obfuscation attacks.

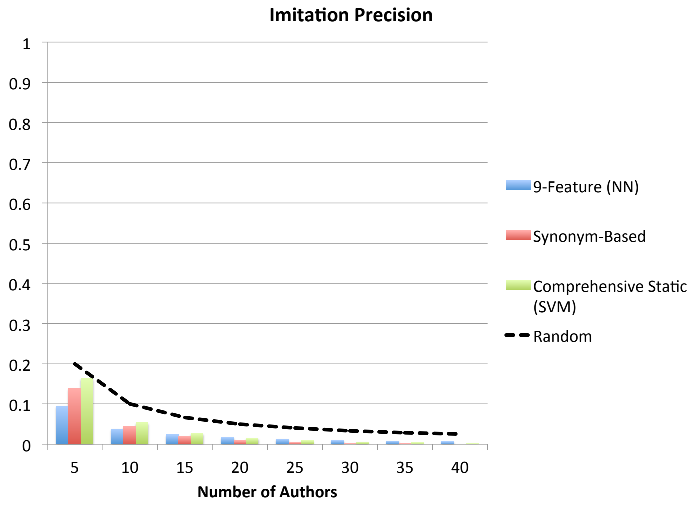


Figure 2.3: Detection of imitation attacks.

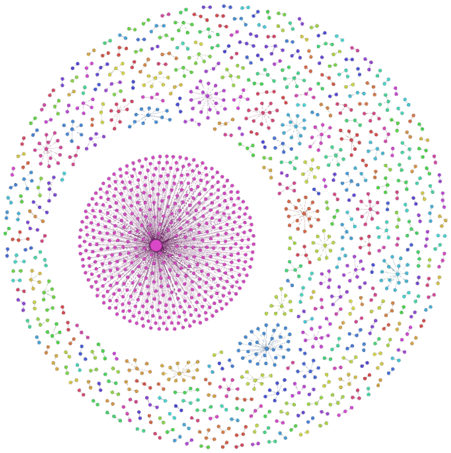


Figure 2.4: Duplicate account groups within Carders as identified by the AE detector. Each dot is one user. There is an edge between two users if AE detector considered them as duplicate user.

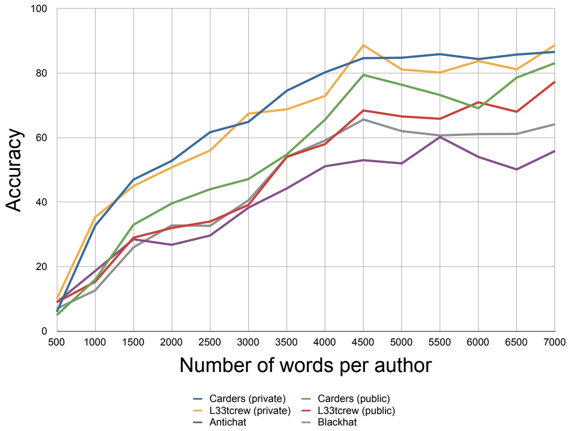


Figure 2.5: **Effect of number of words per user on accuracy**

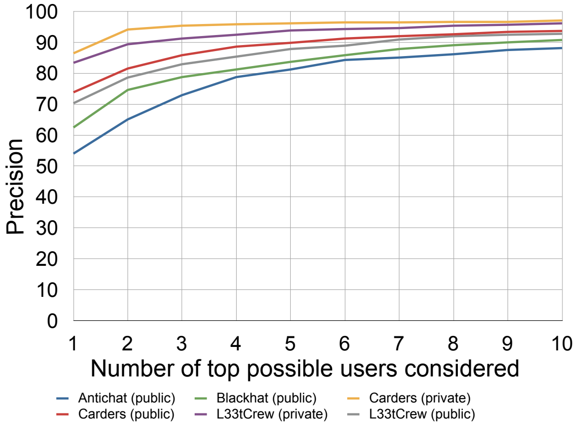


Figure 2.6: User attribution on 50 randomly chosen authors.

3. Stylistic Deception

***This work was completed with support from Michael Brennan.*

Our goal is to determine whether an author has tried to hide his writing style in a written document. In traditional authorship recognition, authorship of a document is determined using linguistic features of an author's writing style. In deceptive writing, when an author is deliberately hiding his regular writing style, authorship attribution fails because the deceptive document lacks stylistic similarity with the author's regular writing style. Though recognizing correct authorship of a deceptive document is hard, our goal is to see if it is possible to discriminate deceptive documents from regular documents.

To detect adversarial writing, we need to identify a set of discriminating features that distinguish deceptive writing from regular writing. After determining these features, supervised learning techniques can be used to train and generate classifiers to classify new writing samples.

3.1 Approach

3.1.1 Feature selection

The performance of stylometry methods depends on the combination of the selected features and analytical techniques. We explored three feature sets to identify stylistic deception.

Writeprints feature set Zheng et al. proposed the Writeprints features that can represent an author's writing style in relatively short documents, especially in online messages [41]. These "kitchen sink" features are not unique to this work, but rather represent a superset of the features used in the stylometry literature. We used a partial set of the Writeprints features, shown in Table 3.1.

Our adaptation of the Writeprints features consists of three kinds of features: lexical, syntactic, and content specific. The features are described below:

Lexical features: These features include both character-based and word-based features. These features represent an author's lexicon-related writing style: his vocabulary and character choice. The feature set includes total characters, special character usage, and several word-level features such as total words, characters per word, frequency of large words, unique words.

Syntactic features: Each author organizes sentences differently. Syntactic features represent an author's sentence-level style. These features include frequency of function words, punctuation and parts-of-speech (POS) tagging. We use the list of function words from LIWC 2007 [27].

Content Specific features: Content specific features refer to keywords for a specific topic. These have been found to improve performance of authorship recognition in a known context [2]. For example, in the spam context, spammers use words like "online banking" and "paypal;" whereas scientific articles are likely to use words related to "research" and "data."

Our corpus contains articles from a variety of contexts. It includes documents from business and academic contexts, for example school essays and reports for work. As our articles are not from a specific context, instead of using words of any particular context we use the most frequent word n-grams as content-specific features. As we are interested in content-independent analytics, we also performed experiments where these features were removed from the feature set.

Table 3.1: **Writeprints feature set**

Category	Quantity	Description
Character related	90	Total characters, percentage of digits, percentage of letters, percentage of uppercase letters, etc. and frequency of character unigram, most common bi-grams and tri-grams
Digits, special characters, punctuations	39	Frequency of digits (0-9), special characters(e.g., %,&, *) and punctuations
Word related	156	Total words, number of characters per word, frequency of large words, etc. Most frequent word uni-/bi-/ tri-grams
Function words and parts-of-speech	422	frequency of function words and parts-of-speech

Lying-detection feature set Our feature set includes features that were known to be effective in detecting lying type deception in computer mediated communications and typed documents [8, 14]. These features are:

1. Quantity (number of syllables, number of words, number of sentences),
2. Vocabulary Complexity (number of big words, number of syllables per word),
3. Grammatical Complexity (number of short sentences, number of long sentences, Flesh-Kincaid grade level, average number of words per sentence, sentence complexity, number of conjunctions),
4. Uncertainty (Number of words express certainty, number of tentative words, modal verbs)
5. Specificity and Expressiveness (rate of adjectives and adverbs, number of affective terms),
6. Verbal Non-immediacy (self-references, number of first, second and third person pronoun usage).

We use the list of certainty, tentative and affective terms from LIWC 2007 [27].

9-feature set (authorship-attribution features) This minimal feature set consists of the nine features that were used in the neural network experiments in Brennan's 2009 paper [7]. The features are: number of unique words, complexity, Gunning-Fog readability index, character count without whitespace, character count with whitespace, average syllables per word, sentence count, average sentence length, and Flesch-Kincaid readability score.

3.1.2 Classification

We represent each document as $(\vec{x}, y)$ where $\vec{x} \in \mathbb{R}^n$ is a vector of n features and $y \in \{Regular,\ Imitation,\ Obfuscation\}$ is the type of the document. In our study, $n = 9$ for 9-features, $n = 20$ for lying-detection features and $n = 707$ for the Writeprints features. For classification, we used Support Vector Machine (SVM) with Sequential Minimal Optimization (SMO) [29] implemented in the WEKA tool [12] with a polynomial kernel. We tested our dataset with other classifiers in the WEKA tool such as k-Nearest Neighbor, Naive Bayes, J48 Decision Tree, Logistic Regression and SVM with RBF kernel. We chose to focus on the SMO SVM as it outperformed other classifiers in most of the test cases. The exception to this is the Lying-detection feature set, in which a J48 Decision Tree[1] outperformed SVMs. J48 is the JAVA implementation of C4.5 algorithm for constructing decision tree [33]. It is notable that the original work using these features also used a decision tree [8].

3.2 Data Collection

We present results on three datasets. The first one is the Extended-Brennan-Greenstadt corpus which contains the Brennan-Greenstadt corpus, extended with regular, obfuscated and imitated writing samples of the AMT workers. The second dataset, which we call the Hemingway-Faulkner Imitation corpus, contains articles from the International Imitation Hemingway Competition and Faux Faulkner contest. The last dataset, Thomas-Amina Hoax corpus, contains blog posts from "A Gay Girl in Damascus" blog, posts of Thomas MacMaster in the alternate-history Yahoo! group[2] as himself and as Amina Arraf, and writing samples of Britta Froelicher, a graduate student at Center for Syrian Studies at St Andrews, who is also Thomas's wife.

3.2.1 Extended-Brennan-Greenstadt corpus

We used the Brennan-Greenstadt adversarial corpus for this study[3]. This dataset consists of two types of writing samples, regular and adversarial, from 12 participants. The regular writing contains approximately 5000 words of pre-existing writing samples per author. The regular writings are formal in nature, written for business or academic purposes. In the adversarial writing samples, participants performed two adversarial attacks: obfuscation and imitation. In the obfuscation attack, each participant tried to hide his identity while writing a 500-word article describing his neighborhood. In

[1] http://weka.sourceforge.net/doc/weka/classifiers/trees/J48.html

[2] http://groups.yahoo.com/group/alternate-history/

[3] This data set is publicly available at https://psal.cs.drexel.edu

the imitation attack, each participant tried to hide his writing style by imitating Cormac McCarthy's writing style in 'The Road' and wrote a 500-word article describing a day of their life in the third person.

We extended this corpus by collecting similar writing samples using AMT. We created a Human Intelligence Task (HIT) where participants were asked to submit the three kinds of writing sample described in the previous paragraph[4]. After collecting the data, we manually verified each submission and only accepted the ones that complied with our instructions. 56 participants' work was accepted.

Participants were also asked to provide their demographic information. According to the provided demographic information, all the participants' native language is English and all of them have some college-level degree.

A total of 68 authors' writing samples are used in this study, 12 of the authors are from the Brennan-Greenstadt corpus and others are the AMT workers.

3.2.2 Hemingway-Faulkner Imitation corpus

The Hemingway-Faulkner Imitation corpus consists of the winning articles from the Faux Faulkner Contest and International Imitation Hemingway Competition[5]. The International Imitation Hemingway Competition is an annual writing competition where participants write articles by imitating Ernest Hemingway's writing style. In the Faux Faulkner Contest participants imitate William Faulkner's artistic style of writing, his themes, his plots, or his characters. Each article is at most 500 words long. We collected all publicly available winning entries of the competitions from 2000 to 2005. The corpus contains sixteen 500-word excerpts from different books of Ernest Hemingway, sixteen 500-word excerpts from different books of William Faulkner, 18 winning articles from The International Imitation Hemingway Competition and 15 winning articles from The Faux Faulkner Contest.

In the imitation contests, participants chose different topics and imitated from different novels of the original authors. Table 3.2, 3.3, and 3.4 show imitation samples. Cormac McCarthy imitation

[4] `https://www.cs.drexel.edu/~sa499/amt/dragonauth_index.php`

[5] Available at http://web.archive.org/web/20051119135221/ http://www.hemispheresmagazine.com/fiction/2005/hemingway.htm

samples are all of same topic but the contest articles are of varied topics and most of winners were professional writers.

Table 3.2: Imitation samples from the Extended-Brennan-Greenstadt dataset.

Cormac McCarthy imitation sample: 1
Laying in the cold and dark of the morning, the man was huddled close. Close to himself in a bed of rest. Still asleep, an alarm went off. The man reached a cold and pallid arm from beneath the pitiful bedspread.
Cormac McCarthy imitation sample: 2
She woke up with a headache. It was hard to tell if what had happened yesterday was real or part of her dream because it was becoming increasingly hard to tell the two apart. The day had already started for most of the world, but she was just stumbling out of bed. Across the hall, toothbrush, shower.

Table 3.3: Imitation samples from the International Imitation Hemingway Competition.

Hemingway imitation sample: 1
At 18 you become a war hero, get drunk, and fall in love with a beautiful Red Cross nurse before breakfast. Over absinthes you decide to go on safari and on your first big hunt you bag four elephants, three lions, nine penguins, and are warned never to visit the Bronx Zoo again. Later, you talk about the war and big rivers and dysentery, and in the morning you have an urge to go behind a tree and get it all down on paper.
Hemingway imitation sample: 2
He no longer dreamed of soaring stock prices and of the thousands of employees who once worked for him. He only dreamed of money now and the lions of industry: John D. Rockefeller, Jay Gould and Cornelius Vanderbilt. They played in the darkened boardrooms, gathering money in large piles, like the young wolves he had hired.

3.2.3 Long Term Deception: Thomas-Amina Hoax corpus

In 2010, a 40-year old US citizen Thomas MacMaster opened a blog “A Gay Girl in Damacus” where he presented himself as a Syrian-American homosexual woman Amina Arraf and published blogposts about political and social issues in Syria. Before opening the blog, he started posting as Amina Arraf in the alternate-history Yahoo! group since early 2006. We collected twenty 500-word posts of Amina and Thomas from the alternate-history Yahoo! group, publicly available articles

Table 3.4: Imitation samples from the Faux Faulkner Contest.

William Faulkner imitation sample: 1
And then Varner Pshaw in the near dark not gainsaying the other but more evoking privilege come from and out of the very eponymity of the store in which they sat and the other again its true I seen it and Varner again out of the near dark more like to see a mule fly and the other himself now resigned (and more than resigned capitulate vanquished by the bovine implacable will of the other) Pshaw in final salivary resignation transfixed each and both together on a glowing box atop the counter.
William Faulkner imitation sample: 2
From a little after breakfast until almost lunch on that long tumid convectionless afternoon in a time that was unencumbered by measure (and before you knew to call it time: when it was just the great roiling expressionless moment known only elliptically and without reference to actual clocks or watches as When We Were Very Young) Piglet, his eyes neither seeing nor not-seeing, stood motionless as if riveted to the iron landscape from which he had equally motionlessly emerged until he became the apotheosis of all tiny pigs wearing scarves standing on two legs and doing absolutely nothing but blinking at what lay before them in the dust.

written by Britta Froelicher[6] who was a suspect of this hoax and 142 blog posts from “A Gay Girl in Damacus.” The blog posts were divided into 500-word chunks. In total we had 248 articles.

3.3 Evaluation and Results

3.3.1 Evaluation Methodology

To evaluate our approach, we perform a three class classification, where the three classes are Regular, Imitation, and Obfuscation, with three feature sets. We use 10-fold cross-validation with Support Vector Maching (SVM) and J48 Decision Tree classifier. In k-fold cross-validation the original dataset is randomly partitioned into k equal folds or subsets. Then the classifier is trained on k-1 subsets and tested on the remaining one subset. The cross-validation process is then repeated k times (the folds), with each of the k folds used exactly once as the validation data. The final estimation of the classifier is the average of the k results from the folds.

We also split the Writeprints feature set into three sets, lexical, syntactic, and content specific and run the three class classification using each set separately. In this experiment, we show that non-specific features are as effective as content specific features in detecting adversarial attacks.

[6] One such article: http://www.joshualandis.com/blog/?p=1831

3.3.2 Results

Can we detect stylistic deception in documents?

The results show that a classifier trained on sets of adversarial and non-adversarial documents can detect deceptive documents with 96.6% accuracy on our best feature set, as is shown in Table 3.5.

The classification was performed on the Brennan-Greenstadt dataset, Amazon Mechanical Turk dataset, and Extended-Brennan-Greenstadt dataset which combines both of the sets. On all of the datasets, the Writeprints features showed the best performance in detecting adversarial attacks. With this feature set, an SVM classifier can detect imitation attacks with 85% accuracy and obfuscation attacks with 89.5% accuracy on the Extended-Brennan-Greenstadt dataset.

Though deception in writing style is significantly different from lying, both deceptions have similar characteristics. With the Lying-detection features we can detect imitation attacks with 75.3% accuracy and obfuscation attacks with 59.9% accuracy.

The 9-feature set, which can detect authorship of regular documents with over 90% accuracy, performed poorly (less than 50%) in detecting adversarial attacks.

The type of machine learning method used in classification is another important factor in detecting deception. The SVM classifier worked best with the Writeprints features whereas the J48 decision tree performed well with the Lying-detection features.

Which linguistic features indicate stylistic deception?

To understand the effect of different features, we ranked the Writeprints features based on their Information Gain Ratio (IGR) [32]. IGR of a feature F_i is defined as,

$$IGR(F_i) = (H(D) - H(D|F_i))/H(F_i),$$

Table 3.5: The table shows performance of different feature sets in detecting regular and adversarial writing samples. The Writeprints feature set with SVM classifier provides the best performance in detecting deception.

Dataset	Feature set, Classifier	Type	Precision	Recall	F1	Avg. F1
EBG	**Writeprints, SVM**	**Regular**	**97.5%**	**98.5%**	**98%**	**96.6%**
		Imitation	**87.2%**	**82.9%**	**85%**	
		Obfuscation	**93.2%**	**86.1%**	**89.5%**	
	Lying-detection, J48	Regular	95.2%	96.2%	95.7%	92%
		Imitation	80.6%	70.7%	75.3%	
		Obfuscation	60.3%	59.5%	59.9%	
	9-feature set, J48	Regular	92.3%	96.8%	94.5%	89%
		Imitation	52.9%	43.9%	48%	
		Obfuscation	61.9%	32.9%	43%	
AMT	Writeprints, SVM	Regular	96.5%	98.6%	97.5%	95.6%
		Imitation	82.3%	72.9%	77.3%	
		Obfuscation	96.4%	79.1%	86.9%	
	Lying-detection, J48	Regular	94.2%	96.2%	95.2%	90.9%
		Imitation	71.7%	54.3%	61.8%	
		Obfuscation	58.5%	56.7%	57.6%	
	9-feature set, J48	Regular	92.5%	96.3%	94.3%	88%
		Imitation	45.5%	35.7%	40%	
		Obfuscation	45.5%	29.9%	36%	
BG	Writeprints, SVM	Regular	94%	100%	96.9%	94.7%
		Imitation	100%	83.3%	90.9%	
		Obfuscation	100%	50%	66.7%	
	Lying-detection, J48	Regular	90%	92.9%	91.4%	85.3%
		Imitation	90.9%	83.3%	87%	
		Obfuscation	11.1%	8.3%	9.5%	
	9-feature set, J48	Regular	89.4%	93.7%	91.5%	84%
		Imitation	25%	25%	25%	
		Obfuscation	83.3%	41.7%	55.6%	

where D is document class and H is entropy. We used WEKA to calculate IGR. The top features are mostly function words, as shown in Table 3.6. Other than function words, some syntactic features such as personal pronoun, adverbs, adjectives, and average word length were some of the most discriminating features.

In our dataset, the non-content-specific features performed similar to the content-specific features in detecting deception, as shown in Figure 3.1, which suggests the possibility of generalizing these features to detect multiple forms of adversarial attacks.

Table 3.6: This table shows the features that discriminate deceptive documents from regular documents. The top discriminating features according to Information Gain Ratio are mostly function words.

Top 20 features	
Imitated documents	**Obfuscated documents**
whats	alot
atop	near
lately	up
wanna	theres
underneath	thousand
anymore	ours
beside	shall
she	thats
herself	cuz
beneath	whats
like	havent
he	Frequency of comma
till	lots
her	tons
onto	anyway
soon	plus
Frequency of dot	other
Personal pronoun	maybe

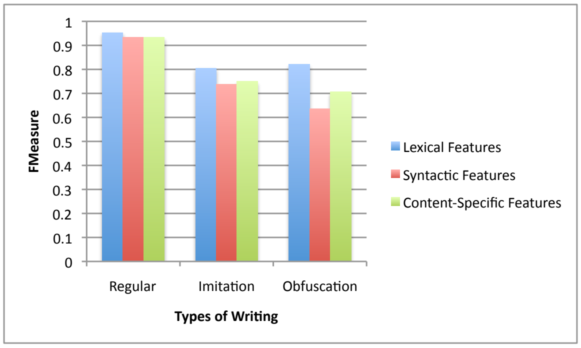


Figure 3.1: Effect of different feature types in detecting adversarial attacks. In each case, SVM is used for classification.

Which features do people generally change in adversarial attacks and which features remain unchanged?

We analyzed the Extended-Brennan-Greenstadt dataset to understand which features people change in stylistic deception. We computed change in a feature f (C_f) in regular and adversarial documents using the following formula:

$$C_f = 100 * (f_{adv} - f_{reg})/(f_{reg} + 1) \tag{3.1}$$

where, f_{adv} and f_{reg} are the average values of feature f in the adversarial documents and regular documents respectively. We added 1 with f_{reg} in the denominator to avoid divide-by-zero error, as f_{reg} can be zero for some features.

Figure 3.2 shows the mostly changed features among parts-of-speech and 9-features. In Figure 3.3, we grouped similar Writeprints features together and added their corresponding changes to show the overall change. In both graphs, the y-axis represents a list of features that have been adjusted in the passages and the x-axis represents the change in each feature.

In our experiments, participants used more existential *there*[7] and adverbs in obfuscated passages. We also noticed an increase in the usage of particles and personal pronouns and decrease in the usage

[7] Existential *there* (EX) is the unstressed *there* that triggers inversion of the inflected verb and the logical subject of a sentence [34]. For example: There/EX were lots of people in town.

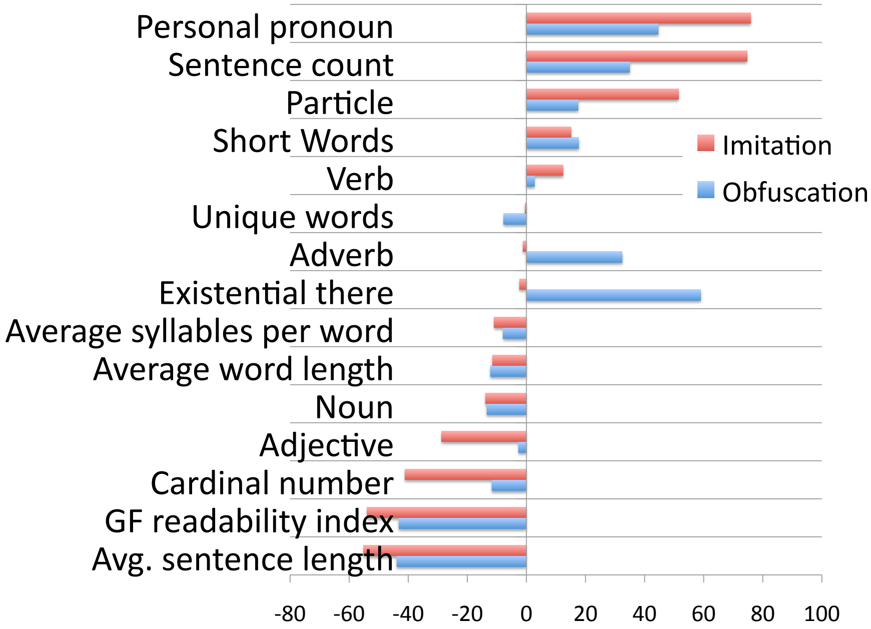


Figure 3.2: Mostly changed features among parts-of-speech and 9-features.

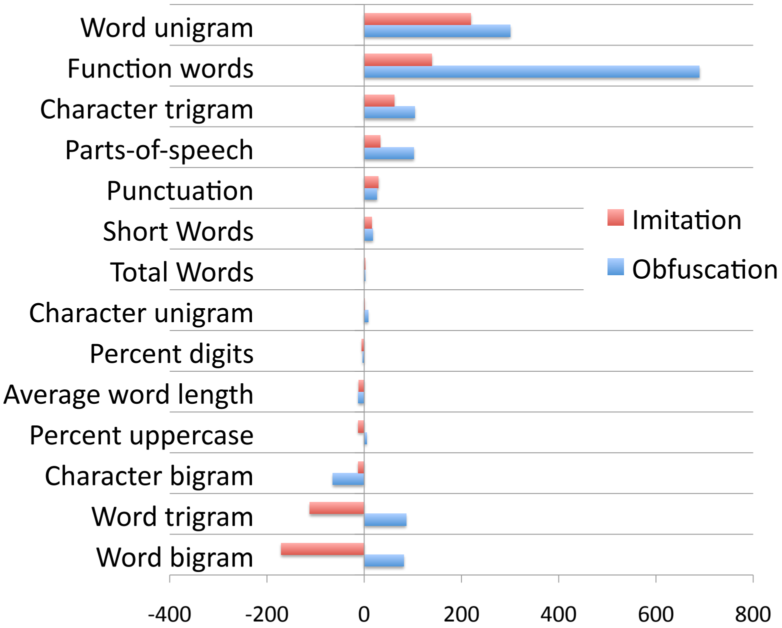


Figure 3.3: Overall changes in the Writeprints features.

of noun, and Wh-pronoun. We do see hints that authors are "dumbing down" their writing style with shorter sentences, simpler words with less syllables, lower readability scores and higher readability ease (Figure 3.2). Another interesting result is that participants used a large number of function words to obfuscated their writings (Figure 3.3). This suggests changing function words as one possible way of anonymizing a document. In our dataset, the function words that were used more in obfuscated documents than regular documents are: I, my, there, are, you. The function words that were used less are: as, his, by, her, be.

In imitated passages, more usage of personal pronouns, particles and less usage of adjectives were noticed (shown in Figure 3.2). Participants also used shorter sentences and words with less syllables which is why the imitated passages had less readability score (Figure 3.2). The personal pronouns can likely be attributed to the direction of the writing survey which asked participants to describe their day in the third person. The shorter sentences are likely a result of imitating Cormac McCarthy who often uses short sentences in his prose. Participants were found to use more function words than regular in this case also, although not as much as in the obfuscated documents.

Information Gain Ratio (IGR) and usage change of a feature are positively correlated (shown in Figure 3.5 and Figure 3.4). All of the highly changed features have high IGR and unchanged features have zero IGR. But some function words were found to have high information gain ratio in spite of being changed very little.

Does stylistic deception share similar characteristics with other deceptions?

Stylistic deception shares similar characteristics with lying. As shown in Table 3.5, features used to detect lying can detect imitation attacks with 75.3% accuracy and obfuscation attacks with 59.9% accuracy on the Extended-Brennan-Greenstadt dataset. This result indicates the similarity of linguistic changes associated with any kind of information hiding.

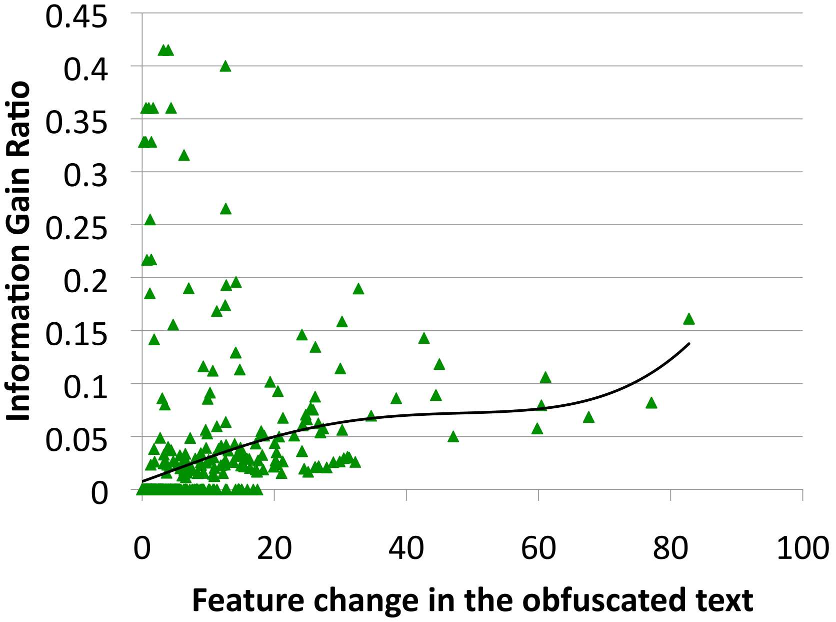


Figure 3.4: Relation between Information Gain Ratio and absolute usage change in features in the obfuscated documents. The black line represents trend.

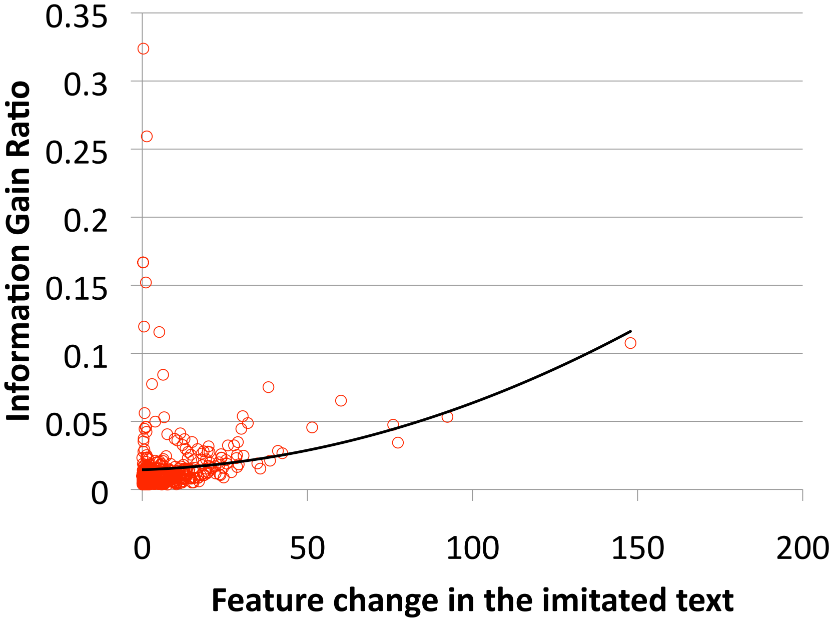


Figure 3.5: Relation between Information Gain Ratio and absolute usage change in features in the imitated documents. The black line represents trend.

Are some adversarial attacks more difficult to detect than others?

In our experiment, obfuscation was harder to detect than imitation. In imitation, participants followed one specific writing style, the writing style of Cormac McCarthy. Different people followed different linguistic aspects in imitating him, for example, some participants used short sentences, some used descriptive adjectives and some used a conversational format with dialogs. But the overall writing style was limited to the style of Cormac McCarthy. Obfuscation is different than imitation as in obfuscation an author can choose to imitate more than one authors' writing style or develop a new style different from his own. However, when we include multiple imitated authors it becomes correspondingly more difficult to detect imitation attacks.

Can we generalize deception detection?

We check whether our deception detection approach that can detect imitation and obfuscation on the Extended-Brennan-Greenstadt can detect imitation samples from the Ernest Hemingway and William Faulkner imitation contests. We performed a 10-fold cross-validation on the Hemingway-Faulkner imitation corpus. We used Writeprints and Lying-detection features with SVM and J48 classifiers respectively from the WEKA tool. Our classifier can distinguish imitated articles from the original writings of Ernest Hemingway and William Faulkner with 88.6% accuracy (Table 3.7).

Table 3.7: Imitated document prediction result: Hemingway-Faulkner imitation corpus. (P = Precision, R= Recall and F= F-measure)

Type	**Lying-detection, J48**			**Writeprints, SVM**		
	P	R	F	P	R	F
Imitation	69.7%	69.7%	69.7%	83.8%	93.9%	88.6%
Regular	61.5%	61.5%	61.5%	92.6%	80.6%	86.2%
Weighted Avg.	66.1%	66.1%	66.1%	88.1%	87.5%	87.4%

We also performed an experiment where a classifier trained on the Extended-Brennan-Greenstadt dataset was tested on Hemingway-Faulkner Imitation corpus. Only 57.1% of the imitated documents were considered as imitation in that case. The Hemingway-Faulkner Imitation corpus is different from our dataset in several ways. The participants in the training set imitated Cormac McCarthy using one pre-specified excerpt from 'The Road' in writing about their day. But in the imitation contests, participants imitated two different authors without any topic constraint. Also the contest winners were found to be more successful than the mechanical turkers in imitating, as shown in Table 3.8. To see how often a classifier can be fooled into predicting imitated document as written by the original authors, we trained an SMO SVM classifier with the Writeprints features using the original writing excerpts of Cormac McCarthy, Ernest Hemingway, William Faulkner and tested the classifier with the imitated documents. In this test, we classified imitated documents into three classes: Cormac McCarthy, Ernest Hemingway, William Faulkner. The result shows that the contest winners can imitate Ernest Hemingway in 84.27% cases, and William Faulkner in 66.67%, whereas the turkers were successful in 47.05% cases in imitating Cormac McCarthy.

Table 3.8: This table shows the success rate (Precision) of participants in imitating different authors. Imitation contest winners were more successful in imitating than the AMT participants.

Author name	Imitation success rate	Writer's skill
Cormac McCarthy	47.05%	Not professional writers
Ernest Hemingway	84.21%	Writing contest winners
William Faulkner	66.67%	Writing contest winners

3.3.3 Detecting long term deception

Detecting long term deception is similar to detecting fiction as deception. Fiction and elaborate deception have different linguistic characteristics than short-term on-the-spur deception, as in the long-term deception the author has sufficient time and topic to write descriptively and edit sufficiently to make it appear as a truthful document. This is why a different approach is required to detect long-term hoaxes and deception. Regular authorship recognition can be helpful to find inconsistencies in writing and to discover real authorship of the deceptive documents.

To test our method on long-term deception, we used the Thomas-Amina Hoax corpus. We performed an authorship attribution test on the posts he created as himself and as Amina in the alternate-history Yahoo! group. We found that he consistently maintained a separate writing style as Amina in the Yahoo! group as none of the Amina's posts were attributed to Thomas in the authorship attribution test. Thomas's writing style as Amina was different than his regular writing style. The use of upper-case letters and modal verbs [8] were noticeable in Amina's posts, whereas Thomas used longer sentences and more adjective and adverbs. Table 3.9 lists the top Writeprints features that discriminate Thomas and Amina's posts in the Yahoo! group.

Table 3.9: The top features according to Information Gain Ratio that discriminate Thomas and Amina.

Feature	Information Gain Ratio
Percent upper case	0.758
t	0.620
than	0.620
all	0.516
only	0.516
tha	0.501
though	0.501
Frequency of *	0.432
less	0.433
can	0.433
not	0.433
a	0.423
that	0.423
Modal verb	0.423
that	0.423
them	0.373
even	0.372
doing	0.361

[8] Modal verbs are verbs that do not take an -s ending in the third person singular present, i.e. can, could, may, might, ought.

Moreover, all of the posts Thomas wrote as Amina and as himself and posts of Britta were considered as regular when tested on an SVM classifier which was trained with the Extended-Brennan-Greenstadt corpus. Deception classification of the posts from "A Gay Girl in Damascus" also did not show any indication of masking. In our test, only 14% of the blog posts were considered as deceptive which is less than the error rate, suggesting a random effect. 13 blog posts were classified as obfuscated documents, 22 were classified as imitated document. Table 3.10 shows that 57.14% of the deceptive documents were attributed to Amina during authorship attribution.

Table 3.10: Blog post classification.

	Thomas	Amina
Imitation	11	11
Obfuscation	4	9

But maintaining an alternate writing style consistently for a long time is hard, which was evident in the Thomas-Amina case. When people started questioning Amina's existence, Thomas and his wife Britta were suspected as possible writers of the blog based on various pieces of evidence, for example, Thomas's address was used in Amina's account, and photos from Britta's picasa album were used in Amina's blog. In the end, Thomas admitted that he was "Amina." Regular authorship recognition also supports this fact. More than half of the blog posts (54.03%) were attributed to Thomas during authorship attribution with an SVM classifier and the Writeprints feature set. Only 10 posts were attributed to Britta and the rest were attributed to "Amina." Native language detection, age and other demographics analysis are other possible ways to detect this form of deception, which are not explored in this paper.

3.4 Discussion

In this research, we showed that two kinds of adversarial attacks—imitation and obfuscation—can be detected with high accuracy using a large feature set. However, the results of cross-validation on the Extended-Brennan-Greenstadt corpus are considerably better than the results on the imitation contest participants (96.6% vs. 87.4%).

There are several potential explanations for this discrepancy. The first is that the original dataset only handles a single imitated author, Cormac McCarthy, and the imitation classifier may be overfit to detect imitations of his style. This suggests that training on a combination of different author imitations will produce a stronger general classifier and that an imitation classifier should be trained on the target of the attack, if known.

The second potential explanation is the skill and incentive differential between the AMT participants and the contest winners. We included all reasonable attempts by the mechanical turk participants, but only the top three contestants per year were included in the contest data set. We offered our participants $5. A Faulker contest winner receives airfare for two to Memphis on United Airlines, five days free car rental, two free passes to the Faulkner Conference, and a complimentary room for five days and nights at an Oxford motel. First prize in the Heminway contest was round-trip tickets and dinner for two at Harry's in Florence, Italy.

A third potential explanation is that part of the success on the original data set is due to topic detection, not deception detection. In the Extended-Brennan-Greenstadt corpus, regular documents are from a variety of documents, obfuscation documents are a description of a neighborhood, and imitation documents provide a third-person narrative of a day.

Our analysis of the feature set shows that the non-content specific features have the same accuracy as of the content-specific features (Figure 3.1). Also, as most top ranked discriminating features are function words, even by ignoring contextual similarity of the documents, it is possible to detect adversarial documents with sufficient accuracy.

While it is true that content features may indicate authorship or deception, we do not believe this is the case here. Our non-deceptive writing samples consist of multiple documents per author, yet our authorship recognition techniques identify them properly with high levels of accuracy. The different content features there did not dissuade the standard authorship recognition techniques and we do not believe they greatly alter the outcome of the deception analysis. Furthermore, previous linguistic research has shown that the frequencies of common function words are content neutral and indicative of personal writing style [24].

What the “A Gay Girl in Damascus” results show is that obfuscation is difficult to maintain in the long term. While Tom’s posts as Amina were not found to be deceptive by our classifier, we show that traditional authorship attribution techniques work in this case.

Implications for Future Analyses The current state of the art seems to provide a perfect balance between privacy and security. Authors who are deceptive in their writing style are difficult to identify, however their deception itself is often detectable. Further, the detection approach works best in cases where the author is trying fraudulently present themselves as another author.

However, while we are currently unable to unmask the original author of short term deceptions, further analyses might be able to do so, especially once a deception classifier is able to partition the sets. On the other hand, the Extended-Brennan-Greenstadt data set used contains basic attacks by individuals relying solely on intuition (they have no formal training or background in authorship attribution) and the results on the more skilled contest winners are less extensive.

We are currently working on a software application to facilitate stylometry experiments and aid users in hiding their writing style. The software will point out features that are identifying to users and thus provide a mechanism for performing adaptive countermeasures against stylometry. This tool may be useful for those who need longer term anonymity or authors who need to maintain a consistent narrative voice. Even though Thomas MacMaster proved extremely skilled in hiding his writing style, half his posts were still identifiable as him rather than the fictional Amina.

In addition, these adaptive attacks may be able to hide the features that indicate deception, especially those in our top 20. It is also possible that attempts to change these features will result in changes that are still indicative of deception, particularly in the case of imitations. The fact is that most people do not have enough command of language to convincingly imitate the great masters of literature and the differences in style should be detectable using an appropriate feature set. A broader set of imitated authors is needed to determine which authors are easier or harder to imitate. Despite our ability to communicate, language is learned on an individual basis resulting in an individual writing style [20].

Implications for Adversarial Learning Machine learning is often used in security problems from spam detection, to intrusion detection, to malware analysis. In these situations, the adversarial nature of the problem means that the adversary can often manipulate the classifier to produce lower quality or sometimes entirely ineffective results. In the case of adversarial writing, we show that using a broader feature set causes the manipulation itself to be detectable. This approach may be useful in other areas of adversarial learning to increase accuracy by screening out adversarial inputs.

4. Detecting multiple identities

***This work was completed with support from Aylin Caliskan Islam, Ariel Stolerman and Damon McCoy.*

In a practical scenario, an analyst may want to find any probable set of duplicate identities within a large pool of authors. Having multiple identities per author is not uncommon, e.g., many people on the Internet have multiple email addresses, accounts on different sites (e.g. Facebook, Twitter, G+) and blogs. Grouping multiple identities of an author is a powerful ability as the easiest way to change identity on the Internet is to create a new account.

Grouping all the identities of an author is not possible using only the traditional supervised authorship attribution. A supervised authorship attribution algorithm, trained on a set of unique authors, can answer who, among the training set, is the author of an unknown document. If the training set contains multiple identities of an author, supervised AA will identify only one of the identities as the most probable author, without saying anything about the connection among the authors in the training set.

4.0.1 Approach

The goal of our work is to identify multiple identities of an author. We leverage supervised authorship attribution to group author identities. For each pair of authors A and B we calculate the probability of A's document attributed to B ($Pr(A \rightarrow B)$) and B's document attributed to A ($Pr(B \rightarrow A)$). We consider A and B as the same author if the combined probability is greater than a threshold. To calculate the pairwise probabilities, for each author $A_i \in \mathcal{A}$ we train a model using every other authors in $\mathcal{A}$ except A_i and test using A_i. The algorithm is described in Procedure 1. We call this method *Doppelgänger Finder*.

This method can be extended to larger groups. For example, for three authors A, B and C we compute P(A==B), P(B==C) and P(C==A). If A=B and C=B, we consider A, B and C as the three identities of one author.

Procedure 1 *Doppelgänger Finder*

```
Input: Set of authors $\mathcal{A} = A_1, ..A_n$ and associated documents, $D$, and threshold $t$
Output: Set of multiple identities per authors, $M$
  $F \Leftarrow$ Add weight k with every feature frequency (default k=10)
  $F' \Leftarrow$ Features selected using PCA on $F$
  ▷ Calculate pairwise probabilities
  for $A_i \in \mathcal{A}$ do
      $n =$ Number of documents written by $A_i$
      $C \Leftarrow$ Train on all authors except $A_i$ using $F'$
      $R \Leftarrow$ Test $C$ on $A_i$ ($R$ contains the probability scores per author.)
      for $A_j \in R$ do
          $Pr(A_i \to A_j) = \frac{\sum_{x=1}^{n} Pr(A_{jx})}{n}$
      end for
  end for
  ▷ Combine pairwise probabilities
  for $(A_i, A_j) \in \mathcal{A}$ do
      $P = Combine(Pr(A_i \to A_j), Pr(A_j \to A_i))$
      if $P > t$ then
          $M.add(A_i, A_j, P)$
      end if
  end for
  return $M$
```

4.0.2 Feature extraction

To identify similarity between two authors we use the same features as regular authorship attribution (Table 2.6), with two exceptions: 1) exclude the word n-grams, and 2) instead of limiting the number of other n-grams, we use all possible n-grams. Word n-grams made the feature extraction process much slower without any improvement in the performance. We used all possible n-grams to increase the difference between authors, e.g., if author A uses a bi-gram "ng" but author B never uses it, then "ng" is an important feature to distinguish A and B. If we include all possible n-grams instead of only the top 50, we can catch many such cases, specially the rare author-specific n-grams.

After extracting all the features, we add weight to the feature frequencies to increase distance among authors. This serves to increase the distance between present and not present features and gives better results. As our features contain all possible n-grams, the total number of feature per dataset is huge (over 100k for 100 authors). All the features are not important and they just make the classification task slower without improving the accuracy. To reduce the number of features without hurting performance, we use Principal Component Analysis (PCA) to weight and select only the features with high variance.

Principal component analysis (PCA) is a widely used mathematical tool for high dimension data analysis. It uses the dependencies between the variables to represent the data in a more tractable, lower-dimensional form. PCA finds the variances and coefficients of a feature matrix by finding the eigenvalues and eigenvectors. To perform PCA, the following steps are performed:

1. Calculate the covariance matrix of the feature matrix F. The covariance matrix measures how much the features vary from the mean with respect to each other. The covariance of two random variable X and Y is:

$$cov(X,Y) = \sum_{i=1}^{N} \frac{(x_i - \bar{x})(y_i - \bar{y})}{N} \tag{4.1}$$

   where $\bar{x} = mean(X)$, $\bar{y} = mean(Y)$ and N is the total number of documents.

2. Calculate eigenvectors and eigenvalues of the covariance matrix. The eigenvector with the highest eigenvalue is the most dominant principle component of the dataset (PC1). It expresses the most significant relationship between the data dimensions. Principal components are calculated by multiplying each row of the eigenvectors with the sorted eigenvalues.

3. One of the reasons for using PCA is to reduce the number of features by finding the principal components of input data. The best low-dimensional space is defined as having the minimal error between the input dataset and the PCA (eq. 4.2).

$$\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{N} \lambda_i} > \theta \tag{4.2}$$

   where K is the selected dimension, N is the original dimension and λ is an eigenvalue. We chose $\theta = 0.999$ so that the error between the original dataset and the projected dataset is less than 0.1%.

4.0.3 Probability score calculation

We use Logistic regression with 'L1' regularization and regularization factor $C = 1$ as a classifier in Procedure 1 to calculate pairwise probabilities. We experimented with linear kernel SVM, which was slower than Logistic regression without any performance improvement. Any machine learning method that gives probability score can be used for this. After that we need to calculate $P(A == B)$ by combining the two probabilities: $P(A \to B)$ and $P(B \to A)$. We experimented with three ways of combining the probabilities:

1. Average: Given two probabilities $Pr(A \to B)$ and $Pr(B \to A)$, combined score is $\frac{Pr(A \to B) + Pr(B \to A)}{2}$.

2. Multiplication: Given two probabilities, combined score is $Pr(A \to B) * Pr(B \to A)$. We can consider the two probabilities as independent because when $Pr(A \to B)$ was calculated A was not present in the training set. Similarly B was not present when $Pr(B \to A)$ was calculated. Also in this case if any of the one-way probabilities are 0, the combined probability would be zero.

3. Squared average: The combined score is $\frac{Pr(A \to B)^2 + Pr(B \to A)^2}{2}$.

All the three approaches give similar precision/recall. We finally used the multiplication approach as its performance is slightly higher in the high recall region.

4.0.4 Baseline

We implement two distance based methods, as suggested by previous work, to compare our performance.

1. Unsupervised: Calculate the euclidean distance between any two authors. Choose a threshold. Two authors are same if the distance between them is less than the threshold.

2. Supervised: Train a classifier using the euclidean distance between any two authors in the training set. Test it using the euclidean distance between the authors in the test set.

We use the same features and classifiers for both our method and the baseline method. Note that, we did not try different feature sets and weighting schemes to improve accuracy. The distance method might provide different results with different feature sets and classifiers.

4.0.5 Evaluation

Data

To evaluate *Doppelgänger Finder* we used a real world blog dataset used in the Internet scale authorship experiment by Narayanan el al.[25]. These blogs were collected by scanning a dataset of 3.5 million Google profile pages for users who specify multiple blogs. From this list of blog URLs, RSS feeds and individual blog posts were collected, filtered to remove HTML and any other markups and only the blogs with at least 7500 characters of text across all the posts were retained. This resulted in total 3,628 Google profiles where 1,663 listed a pair of blogs and 202 listed three to five blogs.

Out of the 1,663 pairs of blogs, many were group blogs with more than one author. We removed the group blogs from the dataset and then manually verified 200 blogs written by 100 authors. Each author in the dataset has at least 4500 words. Among the 200 blogs, we used 100 blogs as our development dataset, we call it **Blog-dev** and the other 100 as a test dataset **Blog-test**. We use the Blog-dev dataset to measure the effect of different feature sets and probability scores. The Blog-test dataset is used to verify that our method provides similar performance on different datasets. The two sets are mutually exclusive.

Methodology

To evaluate our method's performance we use precision and recall. Note that, this is a binary task as oppose to multiclass classification discussed in section 2.3.6. The precision-recall curve (PR curve) shows the precision and recall values at different probability scores. We chose the PR curve instead of ROC curve as we have more false cases (no match between two authors) than true cases, which makes the false positive rate very low even when the number of false positive is very high[1] Area under a curve (AUC) value shows area under the PR curve. Higher value of AUC denotes better performance.

Result

Figure 4.1 shows the precision-recall curve for Blog-dev using different feature sets. The algorithm performs best when all the features are used, although only one feature class (char n-grams or function words) also give high performance. All features give higher combined probability scores than one feature set (Figure 4.2). The combined probability score is much greater than zero when two authors are the same. Our method has similar performance on the Blog-test set of 100 authors with 50 pairs (Figure 4.3). On average, distances between two blogs written by the same author is 0.0001, which is lower than when the blogs are from different authors (0.0003). The distance based method performs much worse than our method on Blog-test set, specially the supervised method performs similar to a random classifier.

4.0.6 Discussion

Dataset	**Threshold**	**Precision**	**Recall**
Blog-dev	**0.004**	**0.90**	**0.94**
	0.01	0.91	0.82
	0.04	1.0	0.64
Blog-test	**0.003**	**0.90**	**0.92**
	0.004	0.95	0.88
	0.01	0.95	0.78
	0.04	1.0	0.46
L33tCrew-Carders	**0.004**	**0.85**	**0.82**
	0.01	0.87	0.71
	0.04	0.92	0.39

Table 4.1: Precision-Recall at different thresholds. Threshold in **bold** gives the best performance.

The goal of our method is to identify possible multiple identities from a dataset by ranking

[1] For example, in the case of 100 authors with 50 true pairs, number of true cases is 50 but number of false cases is 10000-50=9950. So, the false positive rate would be 1% even when number of false positives is 100.

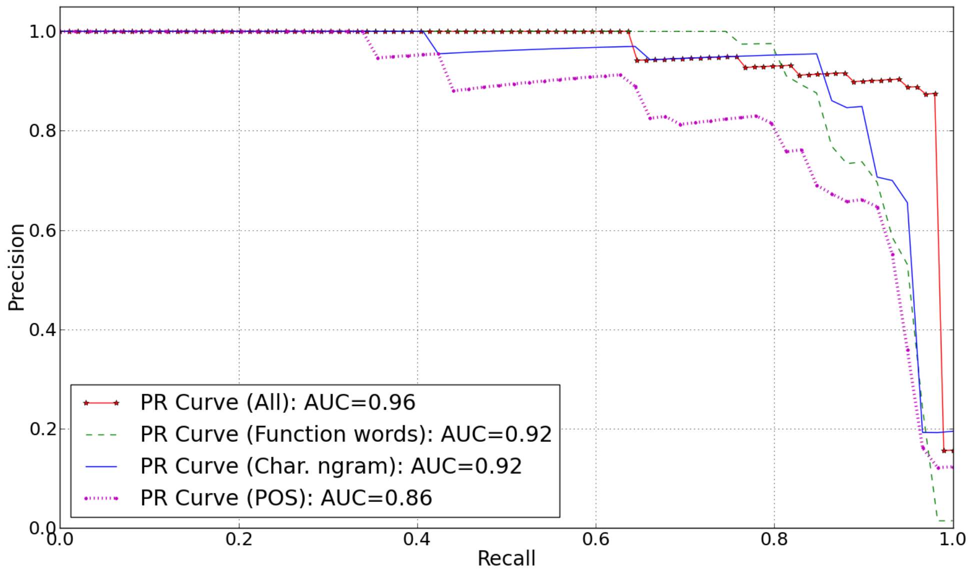


Figure 4.1: Precision/Recall curve on Blog-dev dataset.

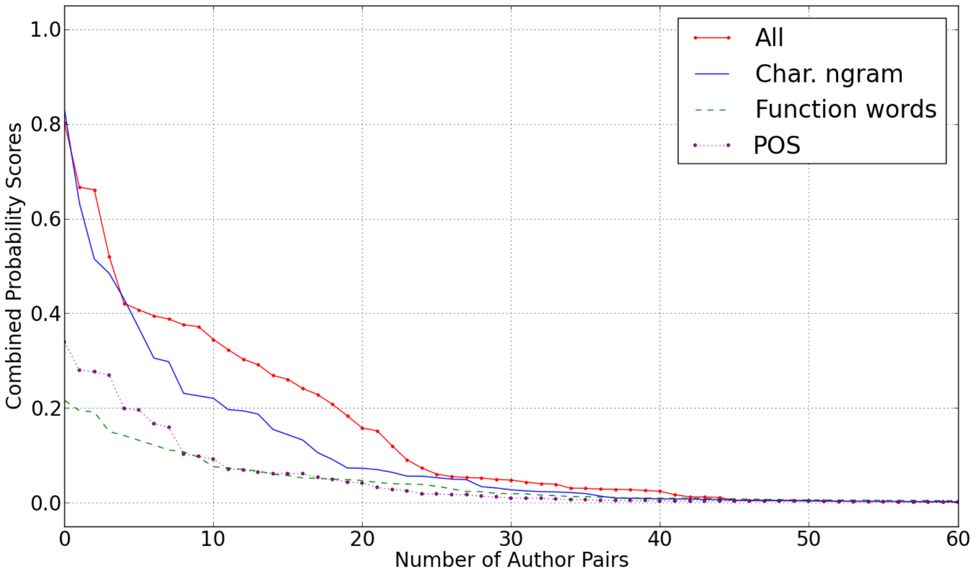


Figure 4.2: Probability scores on Blog-dev dataset.

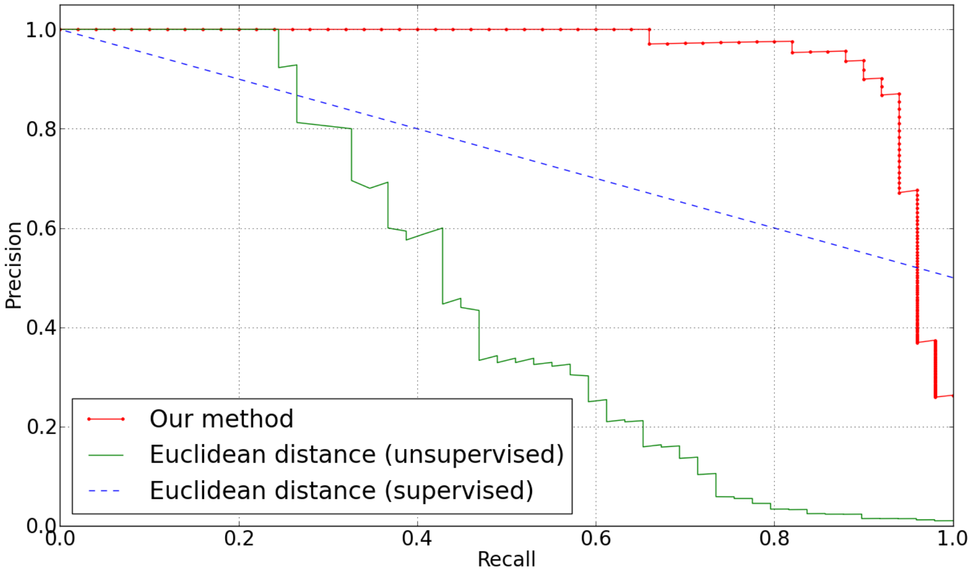


Figure 4.3: Comparing *Doppelgänger Finder* on Blog-test dataset.

the author pairs in case where any training set is unavailable. However, the actual score can vary depending of the properties of the dataset, such as size of the dataset and language of the text. For example, in the Blog-dev dataset the threshold of *0.004* gave the best performance (Table 4.2), but in the Blog-test set 0.003 provided the best recall. The recommended approach of using it for manual analysis is to plot the probability curve (as in Figure 4.2) and verify author pairs in decreasing order. We provide a detailed manual analysis of an underground forum in the following section.

We also experimented with unsupervised clustering algorithms like k-Nearest Neighbor with k=2, but it could cluster 6 out of 50 pairs of blogs.

4.1 Multiple Identities in Underground Forums

In this section we show how our method can be used to identify duplicate accounts by performing a case study on the underground forums. In the forums, many users create multiple identities to hide their original identity (reasons for doing so are discussed later) and they do so by changing the obvious identity indicators, e.g. usernames and email addresses. So we did not have any strong ground truth information for the multiple identities in a forum. We do, however, have some common users across two forums. We treat the common identities in multiple forum as one dataset and use that to evaluate *Doppelgänger Finder* in underground forum. After that we run it on a forum and manually verify our results.

4.1.1 Multiple identities across forums

We collected users with same email address from L33tCrew and Carders. We found 563 valid common email addresses between these two forums. Among them, 443 users were active (had at least one post) in Carders and 439 were active in L33tCrew. Out of these 882 users, 179 had over 4500 words of text. We performed *Doppelgänger Finder* on these 179 authors which included 28 pairs of users (the rest of the 123 accounts did not have enough text in the other forum so merely served as distractor authors for the algorithm). Our method provides 0.85 precision and 0.82 recall when the threshold is 0.004 with exactly 4 false positive cases (Table 4.1 and Figure 4.4).

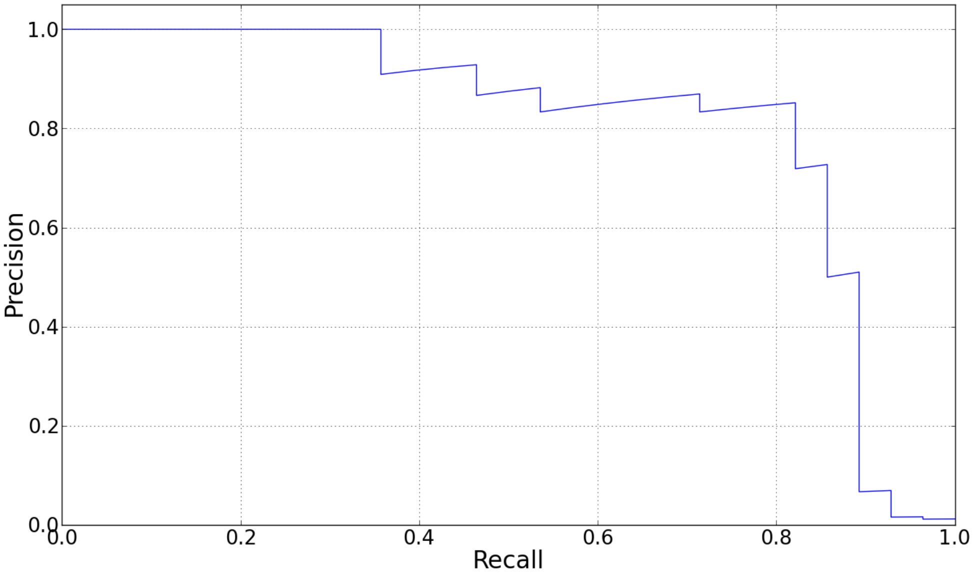


Figure 4.4: *Doppelgänger Finder*: With common users in Carders and L33tCrew: 179 users with 28 pairs. AUC is 0.82.

4.1.2 Multiple identities within forum

We used *Doppelgänger Finder* on Carders and manually analyzed the member-pairs with high scores to show that they are highly likely to be the same user. We selected all the Carders users with at least 4500 words in their private messages, which resulted total 221 users. We chose only private messages as our basic authorship attribution method was more accurate in private messages than in public messages. After that we ranked the member pairs based on the scores generated by our method. The highest combined probability score of the possible pairs is 0.806 and then it goes down to almost zero after the first 50 pairs (Figure 4.5).

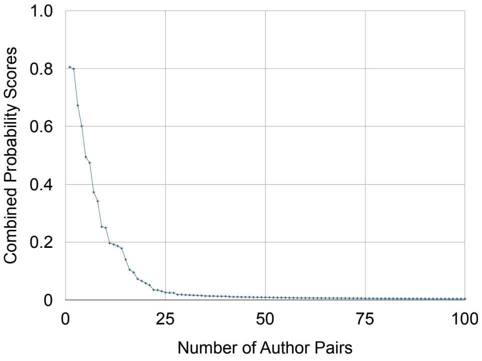


Figure 4.5: Combined probability scores of the top 100 pairs from Carders.

Methodology

Table 4.2 shows the criteria we use to validate the possible doppelgängers. We manually read their private and public messages in the forum and information used in the user accounts to extract these features. The first criterion is to see if two users have the same ICQ numbers a.k.a UINs which is used by most traders to discuss details of their transactions. ICQ's are generally exchanged in private messages. Our second criterion is to match signatures. In all the forums a user can enable or disable a default signature on their forum profiles. Signatures could be generic abbreviations of common phrases such as 'mfg,' or 'Grüße' or pseudonyms in the forum. We also investigate the

Criteria	**Description**
Username	Whether their usernames are same
ICQ	If two users have the same ICQ numbers
Signature (Sig.)	Whether they use the same signatures
Contact Information	Phone number and other contact information shared
Acc. Info	Information in the user table, e.g, their group membership, join and ban date, activity time
Topics	Their topic of discussion
OR AE	At least one of the users trigger the AE detector.
Interaction (Intr.)	Do they talk with each other?
Other	Other identity indicators, e.g., users mention their other accounts or the pair is banned for having the same IP address.

Table 4.2: Criteria for verifying multiple accounts

products traded, payment methods used, topics of messages, and user information in the user table, e.g., join date, banned date if banned, rank in the forum and groups the user joined. We check whether or not they set off the Alter-Ego detector on Carders. Lastly we check whether or not members in a pair sent private messages to each other because that would indicate that they are likely not the same person. We understand that there are many ways to verify identity but in most cases these serve as good indicators.

The *Doppelgänger Finder* algorithm considered $\binom{221}{2}$ possible pairs. We chose all the pairs with score greater than 0.05 for our manual analysis (21 pairs). We limit our analysis to limit the number of pairs to analyze as it could be quite time consuming. We also chose three pairs with low score (rank 22-24 in Table 4.3) to illustrate that higher score pairs are more likely to be true match than the lower score pairs. Note that, all of the top possible doppelgängers use completely different usernames. To protect the members' identity we only show the first three letters of their usernames in Table 4.3.

There are five possible outcomes of our manual analysis: True, Probably True, Unclear, Probably False and False. *True* indicates that we have conclusive evidence that the pair is doppelgängers, e.g., sometimes the pair themselves admit in their private/public messages about their other accounts or the pair shares same IM/payment accounts. *Probably True* indicates that the members share similar uncommon attributes but there's no conclusive evidence of them being the same. *Unclear* indicates that some criteria are similar in both and some are very different and no conclusive attributes either way. *Probably False* means there are very few to no similarity between the members but no evidence that they are not the same. *False* indicates that we found conclusive evidence that the members in a pair are not the same, e.g., the members trade with each other.

Result and Discussion

Rank	Score	Usernames	ICQ	Sig.	Contact	Acc.	Topics	AE	Intr.	Result
1	0.806	per**, Smi**	X		icq		weed	X	0	T
2	0.799	Pri**, Lou**	X					X	0	T
3	0.673	Kan**, deb**	X					X	0	T
4	0.601	Sch**, bob**	–	mfg	–		Kokain	–	0	Prob. T
5	0.495	Duk**, Mer**	X	–				–	0	T
6	0.474	Dra**, Pum**	X					X	0	T
7	0.372	p01**, tol**	–	greezz			X	–	0	Prob. T
8	0.342	Qui**, gam**	X			X		X	0	T
9	0.253	aim**, sty**	X					X	0	T
10	0.250	Un1**, Raz**	X					X	0	T
11	0.196	PUN**, soc**	–		Jabber		X	–	0	T
12	0.192	Koo**, Wic**	–	peace		X	weed	X	0	Prob. T
13	0.187	Ped**, roc**	–				X	–	0	U
14	0.178	Tzo**, Haw**	–				X	X	0	Prob. F
15	0.140	Xer**, kdk**	–			X	X	X	0	U
16	0.105	sys**, pat**	X					X	0	T
17	0.095	Xer**, pat**	–			–	X	X	0	Prob. F
18	0.072	Qui**, Sco**	–					X	0	F
19	0.066	Fru**, DaV**	–			–	–	–	0	Prob. F
20	0.058	Ber**, neo**	–						5	F
21	0.051	Mr.**, Fle**	–					X	26	F*
22	0.01	puT**, pol**	–	–	–	–	–	–	0	F
23	0.001	BuE**, Fru**	–	–	–	–	–	–	0	F
24	0.0001	Car**, Din**	–	–	–	–	–	–	0	F

Table 4.3: Manual analysis of users: X indicates same, – indicates different, empty means the result is inconclusive or complicated with many values.

We found that in Carders, as in the blog and cross-forum experiments, the accounts produced at the high end of the probability range were doppelgängers. The 12 pairs with the highest probabilities were assessed as **True** or **Probably True**. After that, there is a range where both the manual and linguistic evidence is thinner but nonetheless contains some true pairs (pairs 13-17). The manual analysis suggested that pairs below this probability threshold were likely not doppelgängers. Thus, our manual analysis overall agreed with the linguistic analysis performed by *Doppelgänger Finder*.

True (T). True cases are particularly seen when users explicitly state their identities and/or use the same ICQ numbers in two separate accounts. For example, each pair of users in **Pair 1-3, 5, 6, 8 9, 10, and 16** provides an ICQ number in their private messages that is unique to that pair. The users in **Pair-11** use the same jabber nickname. One of the users in **Pair 1** (user name **per****) was asking the admins to give his other account back and telling other members that he is *Smi***.

Other cases had just as convincing, but more subtle evidence. The accounts in **Pair-8** both use *trashmail* which provides disposable email addresses, which shows that these users are careful about hiding their identities. However, the most convincing evidence of their connection was a third doppelgänger account, which we will call user-8c, who did not have enough text to be in our initial user set, but was brought to our attention by the linguistic similarity between the accounts in Pair-8. Both users in Pair-8 share the same ICQ number with user-8c. User-8b explicitly writes two messages from User-8c's account, one in Turkish and one in English revealing his user-8b username. These users do not send private messages to each other. These findings imply that the three user accounts belong to the same person.

Probably True (Prob. T). These accounts do not have a "smoking gun" like a shared ICQ number or Jabber account, however, we are able to observe that the accounts shared have similar interests or other properties. We consider how common these similar properties are in the entire forum and assess as probably true accounts that share uncommon properties.

In the case of **Pair-4**, user-4a does not have an ICQ number, but user-4b frequently gives out an ICQ number. User-4a wants to buy new ICQ numbers. This suggests that he uses ICQ and hides his own ICQ number. They both use a similar signature: 'mfg', but this is common. They trade similar products and talk about similar topics such as Kokain and D2 numbers. Since these are not common, this suggests they might be the same user. User-4a is a newbie while user-4b is a full member. The accounts were active during the same period.

The accounts in **Pair-7** have different ICQ numbers. However, both user-7a and user-7b deal with online banking products, PS3, Apple products, Amazon accounts and cards. They both use Ukash. They both use the same signature such as 'grüße' or 'greezz'. User-7a is a full member and user-7b is permanently banned. They have both been active account holders at the same period. User-7a has a 13th level reputation and user-7b has a 11th level reputation.

Similarly, the accounts in **Pair-12** use the same, rare signature 'peace' and both are interested in weed.

Unclear (U). The accounts in **Pair-13** do not have common ICQ numbers, even though they have the same ICQ numbers with other users (suggesting they do use doppelgänger accounts with lower text, lower reputation accounts). User-13a is a full member with a reputation level of 8. User-13b is a full member with a reputation level of 15. User-13a's products are carding, ps, packstation, netbook, camcorder, and user-13b's products are carding, botnets, cc dumps, xbox, viagra, iPod.

Probably False (Prob. F). The **Pair-14** accounts have different ICQs. User-14a products are tutorials, accounts, Nike, ebay and ps. User-14b's products are cameras and cards. User-14a is a full member with reputation level of 5. User-14b is permanently banned with a reputation level of 15.

One of the users in **Pair-17**, User-17b shares two ICQ numbers with another user but not with User-17a. User-17a's products are iPhone, iPad, macbook, drops, and paypal and User-17b's products are: paypal, iPhone, D2 pins, and weed.

False (F). These users have specific and different signatures and also they use different ICQ numbers. These accounts sometimes interact, suggesting separate identities.

Pairs such as **20** send each other private messages to trade and complete a transaction, suggesting they are business partners not doppelgängers.

The accounts in **Pair-24** do not have any common UINs. They have different signatures, User-24a uses the signature 'LG Carlos' and 'Julix' interchangeably. User-24b never uses 'Carlos' or 'Julix' but he sometimes uses 'mfg' or 'DingDong' at the end of his messages. User-24a's products are iPhone, ebay, debit, iTunes cards, drop service, pack station, fake money while User-24b's products are camera, ps3, paypal, cards, keys, eplus, games, perfumes. They do not talk to each other.

Pair-21 is a special case of false labels. User-21a and user-21b belong to group accounts. User-21a tells user-21b: "*You think it is good that they think we are the same.*", because they got a warning from the admins for using the same computer. In their private messages, they state that they are meeting at each other's houses in person for business, which implies that they might be using the same accounts. They send many messages to other people mentioning each other's names to customers.

4.2 Discussion

4.2.1 Lessons learned about underground markets

Doppelgänger Finder helped us detect difficult to detect dopplegänger accounts. We performed a preliminary analysis on L33tCrew and Blackhat and found similar result as Carders. Our manual analysis of these accounts improves our understanding of why people create multiple identities in underground forums, either within or across forums.

Banning. Getting banned in a forum is one of the main reasons for creating another account within a forum. Rippers, spammers or multiple account holders get penalized or banned once the admins become aware of their actions. Users with penalties get banned once their infraction points go over a certain threshold. There are hundreds of users within forums that have been banned and they open new accounts to keep actively participating in the forums. Some of the new accounts get banned again because the moderators realize that they have multiple accounts, which is a violation of forum rules.

Sockpuppet. Some forum members create multiple accounts in order to raise demand and start a competition to increase product prices.

Accounts for sale. Some users maintain multiple accounts and try to raise their reputation levels and associate certain accounts with particular products and customers. Once a certain reputation level is reached, they offer to sell these extra accounts.

Branding. Some users appear to setup multiple accounts to sell different types of goods. One reason to do this is if one class of goods is more risky, such as selling drugs, the person can be more careful about protecting their actual identity when using this account. Another reason to do this might be to have each account establish a "brand" that builds a good reputation selling a single class of goods, such as stolen credit cards.

Cross-forum accounts. Many users have accounts in more than one forum potentially as a method to grow in their sales by reaching more people not present on the same forums and to purchase goods not offered in a single forum.

Group accounts. In some cases groups of people work together as an organization and each member is responsible for a specific operation among a variety of products that are traded across different accounts. How to adapt stylometry algorithms to deal with multi-authored documents is an open problem that is left as future work.

4.2.2 Lessons learned about Stylometry

We found that any stylometric method can be used in other language by using a high quality parts-of-speech tagger and function words of that language. We have access to one more forum called *BadhackerZ* whose primary language is transliterated Hind using English letters. We did not have a POS tagger that could handle the mixture of these two languages. We were not able to get meaningful results by applying stylometry to *BadhackerZ*, therefore we excluded this forum from stylometric analysis. Similarly, the Russian POS tagger we used produced poor results on our dataset. POS tags generally have high information gain in stylometric analysis and as a result play a crucial role in stylometry. Future work might involve experimenting with other POS taggers or improving their efficacy by producing manually annotated samples of forum text.

4.2.3 Doppelgänger detection by forum administators

One of the primary reasons for banning accounts on these underground forums is because of users creating multiple accounts. This shows that forum administators are actively looking for these types of accounts and removing them since they can be used to undermine underground forums. They use a number of methods ranging from automated tools, such as AE detector, and more manual methods, such as reports from other members. As we have seen from analysis all of these methods are error prone and result in many false positives and false negatives. Many of the false positives were probably generated by users using proxies to hide their IP and location. In addition, when static tools with defined heuristics (IPs, browser cookies, etc.) are used to detect doppelgänger accounts' users can take simple precautions to avoid detection. Many of the accounts detected by doppelgänger finder were not detected by these methods potentially because that user was actively evading known detection methods.

4.2.4 Performance

Our method needs to run N classifiers for N authors. Each classifier is independent, thus can be run in parallel. Using only 4 threads on a quad core Apple laptop the blog experiments took around 10 minutes and the underground forum experiment took around 35 minutes, which can be made faster with more threads.

4.2.5 Hybrid doppelgänger finder methods

Based on what we have learned from our manual analysis of our doppelgänger finder results on Carders, we could potentially build a hybrid method that integrates both stylometry and more underground specific features. For instance, some of the doppelgänger accounts could be identified with simple regular expressions that find and match contact information, such as ICQ numbers. In other cases manual analysis revealed more subtle features, such as two accounts selling the same uncommon product or talk about a similar set of topics can be a good indicator that they are doppelgängers.

Custom parsers and pattern matchers could be created and combined with our doppelgänger finder tool to improve its results. However, it is difficult to know a priori what patterns to look for in different domains. Thus, using doppelgänger finder and performing manual analysis would make this task of designing and adding additional custom tools easier.

4.2.6 Methods to evade doppelgänger finder

There are several limitations to using stylometry to detect doppelgängers. The most obvious limitation is that our method required a large number of words from a single account. A forum member could stop using their account and create a new one before reaching this amount of text, but as pointed out in Section 2.3.1 parts of the forum are closed off to new members, thus less activity is not beneficial to them. They are often not allowed to engage in commerce until they have payed a fee and built up a good reputation by posting.

Another way to evade our method is for the author to intentionally change their writing style to deceive stylometry algorithms. As shown in previous research this is a difficult, but possible task [?], and tools such as Anonymouth can give hints as to how to alter writing style to evade stylometry [23]. We do not currently see any evidence of this technique being used by members of underground forums, but Anonymouth could be integrated into forums.

5. Anonymouth

***This work was completed by Andrew McDonald, Ariel Stolerman and Aylin Caliskan Islam, with support from Sadia Afroz.*

This chapter presents Anonymouth, a novel framework for anonymizing writing style. Without accounting for style, anonymous authors risk identification. This framework is necessary to provide a tool for testing the consistency of anonymized writing style and a mechanism for adaptive attacks against stylometry techniques. Our framework defines the steps necessary to anonymize documents and implements them. A key contribution of this work is this framework, including novel methods for identifying which features of documents need to change and how they must be changed to accomplish document anonymization. In our experiment, 80% of the user study participants were able to anonymize their documents in terms of a fixed corpus and limited feature set used. However, modifying pre-written documents were found to be difficult and the anonymization did not hold up to more extensive feature sets. It is important to note that Anonymouth is only the first step toward a tool to acheive stylometric anonymity with respect to state-of-the-art authorship attribution techniques. The topic needs further exploration in order to accomplish significant anonymity.

5.1 Problem Statement

An author A has a document D that he wants to anonymize. The author selects a set of his own writing D_{pre} and a set B of N authors where $A \notin B$. Author A also chooses a feature set F and authorship attribution method M. The goal is to create a new document $D\prime$ from D where the feature values F are changed sufficiently so that $D\prime$ does not appear to be written by A. To evaluate the level of anonymity, D_{pre} is used. $D\prime$ is anonymized if a classifier trained on D_{pre} and documents written by B, B_{pre}, attributes authorship of $D\prime$ to A with a probability p less than random chance, i.e. $p \leq \frac{1}{N+1}$.

5.2 Approach

Our writing style anonymization framework consists of two platforms: JStylo and Anonymouth. JStylo is a standalone platform for authorship attribution. It is used as an underlying feature extraction and authorship attribution engine for the anonymization framework. Anonymouth is the writing style anonimization platform. It uses the extracted stylometric features and classification results obtained through JStylo and provides suggestions to users to anonymize their writing style.

5.2.1 JStylo: An Authorship-Attribution Platform

JStylo uses NLP techniques to extract linguistic features from documents, and supervised machine learning methods to classify those documents based on the extracted features. JStylo first “learns” the style of known candidate authors based on documents of those authors, and the style of a given set of anonymous documents. It then attributes authorship of the anonymous documents to any of the known authors. JStylo is a Java-based open-source software with a graphic user interface and an extendable API.

Structure and Usage.

The main work-flow of JStylo consists of four consecutive phases: defining a problem set, defining a feature set, selecting classifiers and running the analysis.

A problem set is defined by a training corpus, constructed of documents of all potential authors (as it is supervised learning), and a set of documents of unknown authorship whose authorship are to be determined.

A feature set is defined by a set of various stylistic features to be extracted from the text. Currently there are just above 50 different configurable features available, spanning over different levels of the text, like parts-of-speech in the syntactic level or word frequencies in the lexical level.

The current version of JStylo supports three pre-defined feature sets: Basic-9, Writeprints, and Writeprints (limited). The Basic-9 feature set consists of the nine features that were used in the neural network experiments in [7]. The Writeprints feature set consists of the features used for the Writeprints technique [2]. The Writeprints (Limited) feature set consists of the same features used for Writeprints, where feature classes with potential of exceeding 50 features (e.g. letter bigrams, with a potential of 26^2 features) are limited to the top 50 features. The documents in the training set are mined for the selected features, which are later used for training the classifier, basically profiling the stylistic characteristics of each candidate author. The same features are mined in the test set, for later classification by the trained classifiers.

Each feature is defined by 1) optional text pre-processing tools that allow various filtering methods, to be applied before the feature extraction (e.g. stripping all punctuation); 2) the "core" of the feature which is the feature extractor itself; 3) optional feature post-processing tools to be applied on the features after extraction (e.g. picking the top features frequency-wise); and 4) optional normalization baselines and factoring multipliers (e.g. normalizing over the number of words in each document). The components in 1-3 are based on the JGAAP API [17].

The classifiers available for selection are a subset of Weka [13] classifiers commonly used, such as support vector machine, Naïve Bayes, decision tree, etc. There are several analysis configurations available, the main choice being either to run a 10-fold cross validation analysis over the training corpus or to train the classifiers using a training corpus and classifying the test documents.

JStylo as a Stylometry Research Platform.

The main advantages and novelties of JStylo are 1) allowing integration of multiple features to represent various stylistic characteristics of documents, and 2) a high level of feature-set customizability, where each feature can be configured with its own text pre-processing tools, feature extractors, feature post-processing tools and normalization methods. Its user-friendly graphic interface and Java API allow a high level of usage across both linguistic researchers and computer scientists, providing a convenient platform for stylometry research.

Details of the performance and accuracy of JStylo as a stylometry research platform are discussed in section 5.3.1.

5.2.2 Anonymouth: An Authorship–Anonymization Framework

Anonymouth aims to use the tools of authorship attribution to systematically render them ineffective on a text, while preserving the message of the document in question to the highest degree possible. The task of actively changing the document is however, at this point, left to the user. For Anonymouth to be able to read in a document and output an anonymized version satisfying the constraint that the meaning be preserved, it would need a deep understanding of the structure of the English language (assuming English text), knowledge of almost all words, and a reasonable grasp of things like metaphors and idioms - which is quite a tall order.

After initialization via JStylo, Anonymouth performs an iterative two-step approach to achieve writing style anonymization. These steps are: 1) feature clustering and preferential ordering, and 2) feature modification and document reclassification.

Initialization.

Anonymouth requires[1] the user (A) to input three sets of documents: 1) a single document consisting of approximately 500 ± 25 words, the `documentToAnonymize` (D); 2) a set (at least 2, though preferably more) of sample documents written by the user, totaling 6500 ± 500 words, the `userSampleDocuments` (D_{pre}); and 3) a corpus — preferably made up of at least 3 different authors — of sample documents, *not* written by the user, containing 6500 ± 500 words per author, the `otherSampleDocuments` (B_{pre}). The `userSampleDocuments` are used to determine where the `documentToAnonymize`'s features should *not* be, while the `otherSampleDocuments` are used to determine where the `documentToAnonymize`'s features could be moved to.

After an initial classification has been produced (by JStylo), four groups of features result: 1) those extracted from D, `toAnonymizeFeatures`; 2) those extracted from D_{pre}, `userSampleFeatures`; 3) those extracted from B_{pre}, `otherSampleFeatures`; and 4) a combination of the two previous groups, `userAndOtherFeatures`. Anonymouth then runs Weka's information

[1] In its present state as a research platform rather than a software designed for an end-user, this is the case. However, these limitations are by no means absolute.

gain method on the `userAndOtherFeatures` to select the top f features according to information gain. These top f features will be used in the subsequent computations to generate suggestions for changing writing style. Among the top f features, any that *are not present* in D are excluded from the suggestions Anonymouth deliveres. Resultantly, f becomes $f\prime$. This is done to provide effective suggestions because it cannot be freely assumed that any given feature can be reasonably added to the document. This only applies to JStylo's Writeprints feature sets, where without excluding the non-existing features from suggestions (as an extreme example), a user might be asked to include the word, "Electromagnetic" — when that particular word has no business appearing in the document the user is interested in anonymizing.

Feature Clustering and Preferential Ordering.

Knowing what features to change in order to anonymize a document says nothing about how much to change them, nor does it indicate how much they can be changed and still represent a coherent document that adheres to the rules of grammar. The cause–and–effect relationship among the stylometric features is comparable to that of a field of Dominoes: altering the sentence count of a text will impact the average sentence length; which will affect the Gunning-Fog Readability Index — which happens to be inversely related to the Flesch-Kincaid Reading Ease Score; all of which will inevitably change the character count and will (probably) change the number of occurrences of the three letter word "and". Because of this, it is hard to decide exactly what changes can/should be made in an existing document. However, individually grouping the values of every feature across all B_{pre} seems to provide a fairly decent guideline. It allows Anonymouth to decide how to change each of the $f\prime$ features based upon where the 'real' document's features lie with respect to both one another as well as the user's normal distribution for each feature. The clustering of all instances of each feature assists Anonymouth in selecting physically realizable 'target' values to represent the 'suggested' final document configuration that the user should aim to achieve in order to evade authorship detection. The mechanism behind this selection process is presented through the rest of this section.

Objects containing `otherSampleFeatures` and their respective document names are then fed into a modified k-means clustering algorithm (described in Algorithm 2). The algorithm clusters the objects with respect to each Object's value with, $k = numAuthors$ (where $numAuthors$ is the total number of authors), means, using a modified k-means++ [6] initialization algorithm on a per feature basis spanning across all documents represented by `otherSampleFeatures`. The most significant change to the k-means algorithm is that if any clusters exist with less than three elements after the algorithm converges, the algorithm is re-initialized with $k = k - 1$ means. A more accurate representation might be ak-means. The reasoning behind this adjustment is: because target values for the `documentToAnonymize` are chosen as the centroids of clusters, more elements weighing in on the target value (centroid) increases the potential for anonymization — as opposed to having a single element cluster and effectively copying another's writing style[2]. It remains to be seen whether it would be beneficial to scale the minimum cluster size limit as the number of documents increases; as of now, the value remains fixed.

Implementing these changes in the k-means++ and k-means algorithms creates a safety net that allows Anonymouth to deal with many potential issues that may arise while analyzing an unknown number of documents with unknown similarities/differences. Anonymouth assumes that the documents it receives will be easily clustered. It will adapt if this is not the case, and produce the most beneficial output it can.

[2] There is no guarantee that each cluster will contain documents from more than one author. However, limiting the minimum cluster size helps increase the chances of this happening. In practice, clusters have been observed to contain documents by more than one author more often than not.

Procedure 2 The ak-means Clustering Algorithm (Done on a per-feature basis)

```
1. Initialization:
    (a) run k-means++ algorithm to initialize cluster's based on otherSampleFeatures, with the following exceptions:
        i. If 10,000 numbers have been tried before finding a new centroid, restart.
        ii. If all remaining unchosen values are the same, update the number of total centroids to number of current centroids, set $maxCentroidsFound = True$, and exit initialization; nothing else can be done.
    (b) Assign all instances of the current feature (one per document) from otherSampleFeatures to the centroid nearest to it based on one-dimensional euclidean distance. These are the initial clusters.
2. Update Centroids:
    (a) Calculate the average of the elements (features) contained within each cluster, and update that cluster's centroid with the calculated average.
3. Reorganization:
    (a) Calculate the linear distance between each element, and each existing centroid.
    (b) Assign each element to its closest centroid based on the distance calculation in (a).
    (c) If no elements moved:
        i. If $maxCentroidsFound$ is $True$, or there are at least two clusters with no less than 3 elements per cluster, algorithm has converged.
        ii. If there is only one cluster and $maxCentroidsFound$ is $False$, increment $numMeans$, and Initialize.
        iii. If there are any clusters with less than 3 elements and $maxCentroidsFound$ is $False$, decrement $numMeans$, and Initialize.
    (d) Else if elements did move:
        i. Update centroids.
```

Once the ak-means algorithm has converged, clusters are assigned a preference value based on the primary preference calculation, and placed into an $i \times j$ array after being sorted (from least to greatest).

$$p_{i,j} = numElements_{i,j} \times \mid centroid_{i,j} - userSampleMean_i \mid \tag{5.1}$$

where: $p_{i,j}$ is the primary preference of feature i's jth cluster; $numElements_{i,j}$ is the number of elements in feature i's jth cluster; $centroid_{i,j}$ is the average of feature i's jth cluster's elements; and $userSampleAvg_i$ is the average of the user's sample documents, `userSampleDocuments`, for feature i. The purpose of taking the number of elements into account rather than basing a cluster's preference value off its distance from the user's average values alone is to avoid attempting to modify a user's `documentToAnonymize` to take the form of a document who's features lie in the extremes due to specific content, while refraining from unwittingly restricting the pool of potential target values due to a single feature. Ordering each feature's clusters in such a way that the most desirable cluster has the highest value also lays the groundwork that allows cluster groups to be ordered by a secondary preference calculation.

The secondary preference calculation weights features with respect to their information gain ranking, and ensures that cluster groups that appear with high frequency take precedence over those that appear less often. Because the most desirable cluster, as determined by the primary preference calculation in Eq. (1), has the highest value, weighting the secondary preference calculation in this manner is intended to assign the greatest cluster group preference to the most common cluster group that has the most impact on the features with high information gain. The centroids of the cluster group with the highest ranking are likely to be the best target values for the `documentToAnonymize`. However, because the primary and secondary preference calculations have not been completely optimized, it is possible that the actual best target cluster will be found slightly further down the list of cluster group preferences. For this reason, as well as to help validate the approach by graphically displaying the workings of Anonymouth, the Clusters tab was created.

The "Clusters" tab, as seen in Fig. 5.1, displays the clusters formed by the Algorithm 2, represented by the empty green ellipses which contain clusters of blue dots representing the `otherSampleFeatures`. A shaded purple ellipse displays the user's confidence interval (CI) for a given feature. CI is computed using the following formula,

$$CI = D_{pre_{mean}} \pm 1.96 \times \sigma \tag{5.2}$$

where, $D_{pre_{mean}}$ = average of all `userSampleDocuments`(D_{pre}), and σ = standard deviation from the mean. The visible red dot displays the present value of the same feature in the `documentToAnonymize`, which Anonymouth tries to 'put' in the most populated location as far away from the purple shaded ellipse as possible. By selecting one of the available cluster configurations from the drop-down menu, the user may view configurations from, P_{CG_0} (which should provide the greatest potential to anonymize the document) to $P_{CG_{u-1}}$ (which should provide the least potential to anonymize the document), where u is the number of unique document cluster configurations, and P_{CG_n} is the nth document cluster group's cluster group preference. Upon choosing a configuration, one cluster per feature will be shaded green, and is the target cluster for that feature within that configuration. When a cluster configuration is selected, each target cluster's centroid — represented by the empty black circle — is set to be the target value for each feature (respectively) within the `documentToAnonymize`.

One might ask, why not simply pick the cluster farthest away from the author's average value for each feature? The danger in doing this, as has been determined experimentally, is that many features are co-dependent upon one another; so, it may be physically impossible to modify a document to be represented by a set of independently chosen features. For example, it is impossible to increase the average sentence length of a document, while increasing the number of sentences (assuming the idea is to keep the document more or less the same length). Target values for features must be selected while being mindful of other features. Why, then, not just use the document with a cluster group configuration farthest from the author's standard set of values? This is done because ideally it is more feasible to alter an existing document to look 'generic' than it is to attempt to drive it

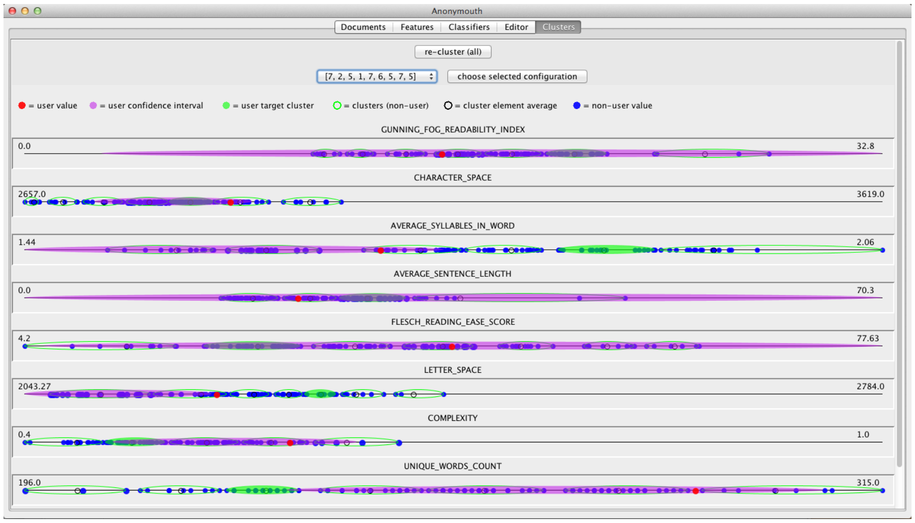
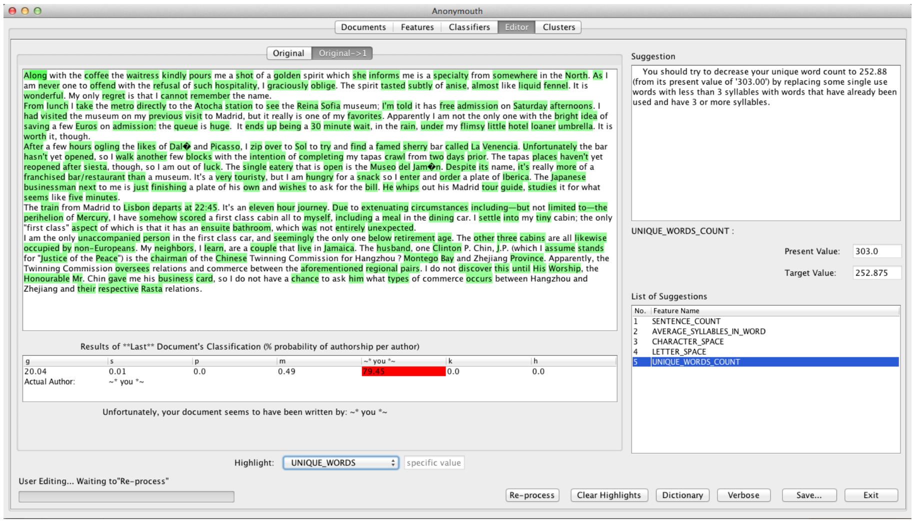


Figure 5.1: Anonymouth Clusters tab with cluster group P_{CG_0} selected, using the Basic-9 feature set, with 6 'otherSample' authors. The red circles display the present value of each feature within the user's 'documentToAnonymize', the purple ellipses depict the user's confidence interval (assuming a normal distribution) for each feature, and the shaded green ellipses show where the user's feature's will fall if all features are modified as recommended by Anonymouth.

toward an extreme, which may only be that way as a result of content (including unusual errors that one might have a hard time trying to, or would not want to, reproduce). If many documents share more or less the same configuration, there is a greater chance that any given document can also be fit to share that configuration while maintaining readability. Furthermore, changing a document to look more like many other documents should be more effective in making it anonymous than simply altering it to look as much unlike the true author's work as possible.

Feature Modification and Document Reclassification.

Once the targets are selected, the user is presented with a clickable list of features to change. When a feature is selected, a suggestion appears that aids the user in changing the present value of the feature to its target value. The suggestions for the Basic-9 feature set have been optimized to guide the user to change the elements in their document that will have the greatest overall impact on its classification. An example of this is, "[replace] some single use words with less than 3 syllables with words that have already been used and have 3 or more syllables", as seen in Fig. 5.2. Once



Along with the coffee the waitress kindly pours me a shot of a golden spirit which she informs me is a specialty from somewhere in the North. As I am never one to offend with the refusal of such hospitality, I graciously oblige. The spirit tasted subtly of anise, almost like liquid fennel. It is wonderful. My only regret is that I cannot remember the name.
From lunch I take the metro directly to the Atocha station to see the Reina Sofia museum; I'm told it has free admission on Saturday afternoons. I had visited the museum on my previous visit to Madrid, but it really is one of my favorites. Apparently I am not the only one with the bright idea of saving a few Euros on admission: the queue is huge. It ends up being a 30 minute wait, in the rain, under my flimsy little hotel loaner umbrella. It is worth it, though.
After a few hours ogling the likes of Dal� and Picasso, I zip over to Sol to try and find a famed sherry bar called La Venencia. Unfortunately the bar hasn't yet opened, so I walk another few blocks with the intention of completing my tapas crawl from two days prior. The tapas places haven't yet reopened after siesta, though, so I am out of luck. The single eatery that is open is the Museo del Jam�n. Despite its name, it's really more of a franchised bar/restaurant than a museum. It's a very touristy, but I am hungry for a snack so I enter and order a plate of Iberica. The Japanese businessman next to me is just finishing a plate of his own and wishes to ask for the bill. He whips out his Madrid tour guide, studies it for what seems like five minutes.
The train from Madrid to Lisbon departs at 22:45. It's an eleven hour journey. Due to extenuating circumstances including—but not limited to—the perihelion of Mercury, I have somehow scored a first class cabin all to myself, including a meal in the dining car. I settle into my tiny cabin; the only "first class" aspect of which is that it has an ensuite bathroom, which was not entirely unexpected.
I am the only unaccompanied person in the first class car, and seemingly the only one below retirement age. The other three cabins are all likewise occupied by non-Europeans. My neighbors, I learn, are a couple that live in Jamaica. The husband, one Clinton P. Chin, J.P. (which I assume stands for "Justice of the Peace") is the chairman of the Chinese Twinning Commission for Hangzhou ? Montego Bay and Zhejiang Province. Apparently, the Twinning Commission oversees relations and commerce between the aforementioned regional pairs. I do not discover this until His Worship, the Honourable Mr. Chin gave me his business card, so I do not have a chance to ask him what types of commerce occurs between Hangzhou and Zhejiang and their respective Rasta relations.

Results of **Last** Document's Classification (% probability of authorship per author)

g	s	p	m	~* you *~	k	h
20.04	0.01	0.0	0.49	79.45	0.0	0.0
Actual Author:	~* you *~					

Unfortunately, your document seems to have been written by: ~* you *~

Suggestion

You should try to decrease your unique word count to 252.88 (from its present value of '303.00') by replacing some single use words with less than 3 syllables with words that have already been used and have 3 or more syllables.

UNIQUE_WORDS_COUNT :

Present Value: 303.0

Target Value: 252.875

List of Suggestions

No.	Feature Name
1	SENTENCE_COUNT
2	AVERAGE_SYLLABLES_IN_WORD
3	CHARACTER_SPACE
4	LETTER_SPACE
5	UNIQUE_WORDS_COUNT

Highlight: UNIQUE_WORDS specific value

User Editing... Waiting to"Re-process"

Figure 5.2: Anonymouth Editor tab showing the 'Unique Words Count' suggestion selected, with unique words (only used once) highlighted, and an initial classification attributing authorship of the displayed text to the user of the Anonymouth with 79.5% probability.

the document has been changed so that its present values reflect the target values, the document is reclassified. If the document has reached a sufficiently low classification[3] the document is considered anonymized. Until that point, the process loops back to 'feature clustering and preferential ordering.' Every time the features are clustered, slightly different clusters may result; which leads to changing target values. We found that in some cases (especially for the Writeprints features) clustering the features only once is a better alternative to continually re-clustering the features upon every classification.

The Editor tab contains a 'Dictionary' which brings up an interface to Princeton's WordNet 3.0, allowing a user to search for synonyms and words containing various continuous character strings (e.g. 'ie'). A 'verbose' button will bring up a window that prints Anonymouth's standard output and error streams in real time as well. Finally, should the user want to revert back to a previous copy of the `documentToAnonymize`, tabs that display where each copy of the document originated from permit the user to trace back through processed changes, while viewing each document's classification results.

[3] In this case, a sufficiently low classification means at or below random chance, which is $1/(numAuthors)$, where $numAuthors$ is the total number of authors.

Table 5.1: Authorship attribution results using Writeprints, Synonym-based and JStylo.

Method	*Accuracy*
Writeprints	73.33%
Synonym-based	89.61%
JStylo with Basic-9	53.98%
JStylo with Writeprints (Limited)	92.03%

5.3 Evaluation and Results

This section discusses the results of the Anonymouth user study. The following subsections explain the effectiveness of JStylo in attributing authorship, effectiveness of Anonymouth in anonymizing a document, the effect of the choice of background corpus and feature set on anonymity, which features were changed by the users to achieve anonymity, and the user satisfaction survey.

5.3.1 Effectiveness of JStylo

To evaluate the effectiveness of JStylo as a sufficiently accurate authorship attribution engine for Anonymouth and as an authorship attribution research platform in general, we conducted experiments using the Brennan-Greenstadt Adversarial Stylometry Corpus, which includes 13 authors with 5000-word documents each. We then compared the results with those of two other state-of-the-art authorship attribution methods in the literature: the Writeprints method and the synonym-based approach [9]. The experiments with JStylo were conducted using a SVM classifier, over two feature sets: the Basic-9 and the Writeprints (Limited). All experiments were evaluated using 10-folds cross-validation. The results are summarized in table 5.1.

Although the Basic-9 feature set did not produce as high results as the other methods, it is still much higher than random chance (7.69%), and is used only as baseline for authorship attribution features in JStylo, or anonymization features baseline in Anonymouth. It is notable that using the Writeprints (Limited) feature set with JStylo produced the highest results across all four experiments.

5.3.2 Effectiveness of Anonymouth

Figure 5.3 shows authorship attribution accuracy of the modified and unmodified documents. Using the Basic-9 features 80% participants were able to anonymize their documents in terms of the corpus used. The first participant's (s1) original document was not attributed to him as an author. The second participant (s2) made no changes to his document. All other participants were able to anonymize their documents.

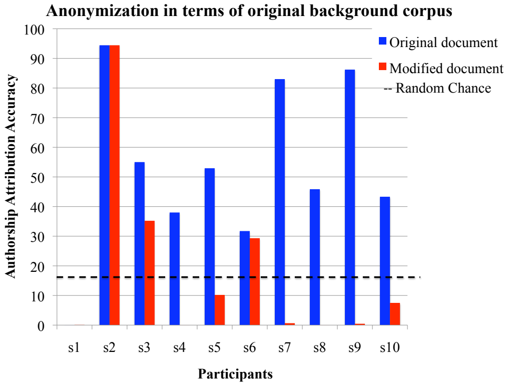
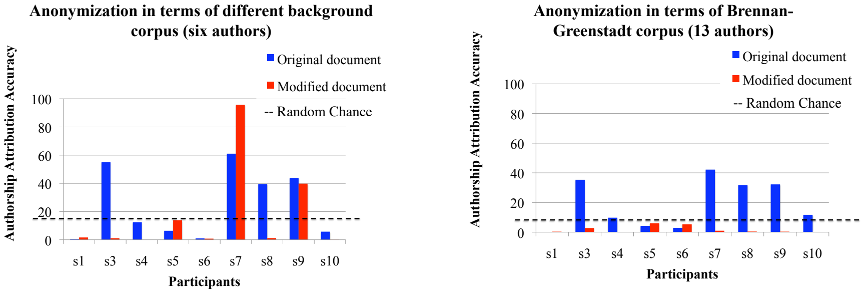


Figure 5.3: Authorship attribution accuracy of modified and original documents using the original background corpus. The Basic-9 feature set and SMO SVM classifer were used. All subjects who made changes were able to anonymize their documents (8/10).

5.3.3 Effect of the Background Corpus on Anonymity

The background corpus, or set of reference authors and documents, is important for document anonymization with Anonymouth as the tool calculates the average value of each feature based on the background corpus and suggests changes to users based on the average feature values.

We tested if documents anonymized in terms of one background corpus are also anonymized against a different background corpus. To test this, we used a different six author subset from the Brennan-Greenstadt adversarial corpus. We also tested the results using the whole 13-author corpus. Results are shown in Figure 5.4(a) and Figure 5.4(b). The effectiveness of the anonymization changes if the background corpus is changed. Unfortunately, the basic 9-Feature set is not very effective at stylometry. Where possible, we pre-selected documents that were correctly classified for the anonymization with respect to the original background corpus. However, when we switched to the new background corpus, only four of these were correctly classified. Of these four, 50% (2) of the authors' documents were still anonymized even in terms of a different corpus of six authors and the others remained anonymized (as shown in Figure 5.4(a)). For the corpus of 13 authors, 5 subjects' original documents were classified correctly and all modified documents were classified incorrectly (as shown in Figure 5.4 (b)).



(a) Six different authors samples as background corpus

Figure 5.4: Authorship attribution accuracy of modified and original documents using six different author samples as background corpus 5.4(a) and 13 authors as background corpus 5.4 (b). The Basic-9 feature set and SMO SVM classifier were used.

5.3.4 Effect of Feature Set on Anonymity

We wanted to see if documents anonymized with one authorship attribution approach are detectable by another approach. Unfortunately in every case documents anonymized with Basic-9 features were attributed to the real author when Writeprints (Limited) feature were used. The Writeprints feature set is much larger than Basic-9, contains around 700 linguistic, content specific and structural features. Most of these features are very low level features, for example, frequencies of character uni-/bi-/tri-grams. Providing effective suggestions for such low level features is challenging. Changing existing documents by following those suggestions to hide author specific features is also very difficult. For this reason, none of the participants in our study were able to anonymize themselves using the Writeprints (Limited) features.

We wanted to evaluate functionality of Anonymouth using the Writeprints (Limited) features to find out the minimum number of features that need to be changed to anonymize a document. To do this, we first ranked the features based on information gain ratio [32]. Then we calculated clusters of feature values using Anonymouth. We chose the top K features based on information gain ratio and changed their values with those of the first cluster, where K= 25, 50, 75, ..., 300. Result of the experiment is shown in Figure 5.5.

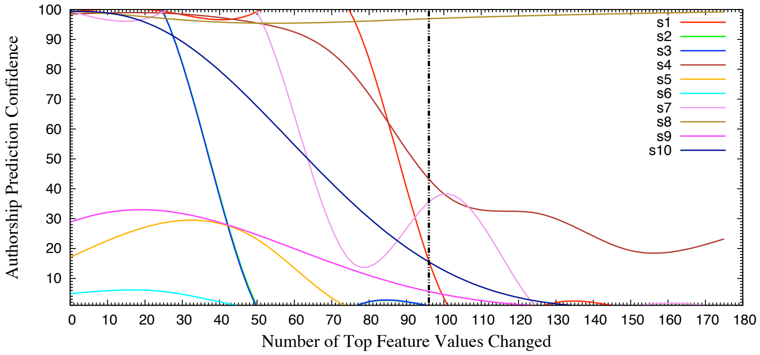


Figure 5.5: Number of Writeprints (Limited) features needs to be changed to achieve anonymity. Authorship prediction accuracy decreases as the top feature values are replaced with those predicted by Anonymouth. Sufficient anonymity can be achieved after changing 13.79% (96 out of 696) of the features.

The result shows that authorship prediction accuracy decreases as the top feature values are replaced with the values predicted by Anonymouth. After changing 13.79% of the features, 90% of the documents were anonymized. This experiment shows that the core approach of Anonymouth works successfully to anonymize a document even against a robust feature set like Writeprints.

5.3.5 Change in features

We compare the frequencies of different features to understand which ones people change to anonymize their writing style. The changes made to features are shown in Figure 5.6. We only used samples of the participants who were successful in anonymizing their documents. This graph illustrates the changes in frequencies for each feature. The y-axis contains a list of features that have been adjusted in the passages and x-axis of the graph denotes the change in each feature. We compute the change in feature using the following formula:

Change in Feature f, $C_f = (f_{mod} - f_{ori})/(f_{ori})$

where,

f_{mod} = Value of feature f in the modified document.

f_{ori} = Values of feature f in the original document.

The amount to the right of the y-axis represents the increases in a feature and the amount to the left represents the decreases. 87.5% of the successful participants (7/8) increased average sentence length and decreased sentence count. Average syllable count was increased in 75% of the cases. Increase in complexity was also noticed in every anonymized document. This indicates that most participants made their language complicated to anonymize their documents, which is also evident by the increase of the Gunning-Fog (GF) readability index.

This result differs from the feature changes in the Brennan-Greenstadt adversarial documents where participants used simpler language to obfuscated their document [3]. Ideally a document can be anonymized by using a language that is either more complex or less complex than the original writing style of an author. As seen in [3], people usually use less complex language while obfuscating their writing style, which is easily distinguishable from regular writings. Anonymouth allows user to choose his own background corpus and provide suggestions to change his writing style. Thus by choosing a diverse background corpus an author can hide both his writing style and the indication

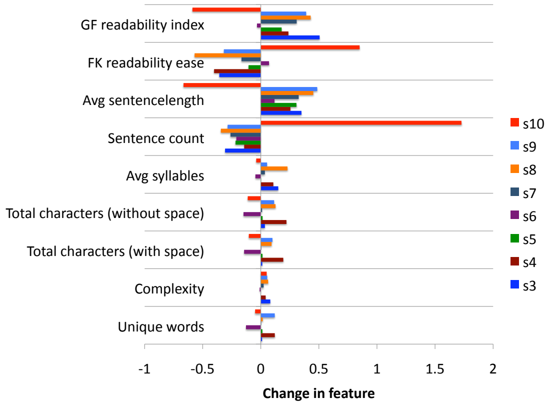


Figure 5.6: Feature changes to achieve anonymity

of changing style.

5.4 Discussion

Although it appears to be quite challenging for a user to implement the changes that Anonymouth asks for, even when only using Brennan and Greenstadt's 'Basic-9' feature set, preliminary results suggest that when users are able to do what is asked, they *can* successfully anonymize their documents — with respect to that feature set. As shown in Fig. 5.3, 80% of participants were able to reduce the accuracy of the SVM classifier used with respect to the original background corpus used. Furthermore, 60% of participants succeeded in achieving a final classification probability below random chance, which for a total of 7 authors is just under 14.3%.

Initial user tests using the 'Writeprints (Limited)' feature set implemented by JStylo suggested less usability than existed when using the Basic-9 feature set in terms of users being able to perform the actions requested by Anonymouth. Due to the complex nature of the Writeprints (Limited) feature set, the user is asked to do things like add more of the letter 'i' to his/her document, or to decrease the number of occurrences of a part of speech n-gram. While no one was able to anonymize their document with respect to the Writeprints (Limited) feature set, it has been shown that in general, if approximately 15% of possible features are changed to the values determined by Anonymouth, a document initially classified as having been written by its actual author with 98% probability, will — about 80% of the time — end up being classified as having been written by another author with over 95% probability.

This suggests that the core of Anonymouth — the methods used to determine what and how much should be changed within a document — have some merit. That is not to say that Anonymouth's core has either been optimally adjusted or is in fact the best way to decide how to anonymize a document. There is a clear separation between knowing the degree to which certain things need to be changed, and being able to execute those changes. Resulting from finding that Anonymouth's suggestions to the user regarding how to make these changes need re-working, it is quite possible that Anonymouth's algorithms will need to be re-worked as well.

5.4.1 Future Work

In general, it seems as though the information presented to the user should be of a higher level, such as, "re-write this sentence using the third person and in the past tense". Of course, doing just this is not the solution. In attempting to anonymize a document via a set of naïve algorithms, there appears to be a trade-off between anonymity and affect. Assuming that the author of a document has written that document in a style that he usually writes in, it is very difficult for that author to go back to another document and modify it to then appear in a different style, while retaining the document's meaning (it is assumed that in order to retain meaning, the imagery and tone would have to create the same end result). Simply stripping descriptive words, modifying tense, and altering the point of view (e.g. from third to first person) would certainly increase anonymity; though clearly at the expense of the documents impact on the audience (affect). While this is one approach that may be taken, it seems far from ideal, and as though it ought to be considered as a last resort.

To achieve its goal, Anonymouth must be able to understand what a sentence/passage means to the extent necessary to enable it to produce an output passage expressed using language constructs foreign to the original author's work that can *at least* capture the main idea and tone of the original passage. While a perfect system would be quite challenging to implement, constructing a system that offers a list of potentially reasonable alternatives to a given passage seems to be a realizable goal.

Adding these features to Anonymouth would resolve the current usability problem that limits the application of Anonymouth.

6. Open Questions

6.1 Verifying Authorship

In supervised authorship attribution, the algorithm would choose any author in the training set, even when the original author is not in the training set. Controversial, pseudonymous documents that are published on the Internet often have an unbounded suspect list. Even if the list is known with certainty, training data may not exist for all suspects. Nonetheless, classical stylometry requires a fixed list and training data for each suspect, and an author is always selected from this list. This is problematic both for forensics analysts, as they have no way of knowing when widening their suspect pool is required, and for internet activists as well, who may appear in these suspect lists and be falsely accused of writing certain documents.

6.2 Identifying documents across domains

People's writing style change depending on domains, context, and topics. For example, writing style on twitter is different from writing style in blogs or in academic articles. In case of underground forums, we noticed the style of writing is different in different forums. How the features vary when domain and context change is still an open question.

6.3 Identifying Robust Features

When people change their writing style consciously or unconsciously some linguistic features, e.g., content words, change and some features remain unchanged. In our work and previous research showed that for most users function words are more robust and are unlikely to change due to topic and context change. But function words alone are not enough to provide high accuracy in a dataset. Also, all function words do not have same importance in detecting an author. Authorship attribution accuracy in adversarial situation can be improved by identifying author-specific robust features.

7. Conclusions

Stylometry is necessary to determine authenticity of a document to prevent deception, hoaxes and frauds. In this work, we showed that manual countermeasures against stylometry can be detected using second-order effects. That is, while it may be impossible to detect the author of a document whose authorship has been obfuscated, the obfuscation itself is detectable using a large feature set that is content-independent. Using Information Gain Ratio, we showed that the most effective features for detecting deceptive writing are function words. We analyzed a long-term deception and showed that regular authorship recognition is more effective than deception detection to find indication of stylistic deception in this case.

Doppelgänger Finder enables easy analysis of a forum for high-value multiple identities. Our analysis of Carders has already produced insights into the use of multiple identities within these forums. We have confidence it can be applied to other forums, given the promising results on blogs and cross-forum accounts. This technique can also be used to detect multiple identities on non-malicious platforms.

This work also motivates the need for improved privacy enhancing technologies such as Anonymouth (Section 5) for authors who wish to not have their pseudonymous writings linked.

Bibliography

[1] A. Abbasi and H. Chen. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems*, 26(2):7, 2008.

[2] Ahmed Abbasi and Hsinchun Chen. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Trans. Inf. Syst.*, 26(2):1–29, 2008.

[3] S. Afroz, M. Brennan, and R. Greenstadt. Detecting hoaxes, frauds, and deception in writing style online. In *Proceedings of the 33rd conference on IEEE Symposium on Security and Privacy*. IEEE, 2012.

[4] Mishari Almishari, Paolo Gasti, Gene Tsudik, and Ekin Oguz. Privacy-preserving matching of community-contributed content. In *Computer Security–ESORICS 2013*, pages 443–462. Springer, 2013.

[5] Lorenzo Alvisi, Allen Clement, Alessandro Epasto, U Sapienza, Silvio Lattanzi, and Alessandro Panconesi. Sok: The evolution of sybil defense via social networks. In *IEEE security and privacy*, 2013.

[6] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007.

[7] M. Brennan and R. Greenstadt. Practical attacks against authorship recognition techniques. In *Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference*, 2009.

[8] J. Burgoon, J. Blair, T. Qin, and J. Nunamaker. Detecting deception through linguistic analysis. *Intelligence and Security Informatics*, pages 958–958, 2010.

[9] Jonathan H. Clark and Charles J. Hannon. A classifier system for author recognition using synonym-based features. In *Lecture Notes in Computer Science*, volume 4827, pages 839–849. Springer, 2007.

[10] George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*, 2009.

[11] David Mandell Freeman. Using naive bayes to detect spammy names in social networks. In *Proceedings of the 2013 ACM workshop on Artificial intelligence and security*, pages 3–12. ACM, 2013.

[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[14] J.T. Hancock, L.E. Curry, S. Goorha, and M. Woodworth. On lying and being lied to: A linguistic analysis of deception in computer-mediated communication. *Discourse Processes*, 45(1):1–23, 2008.

[15] D.I. Holmes and R.S. Forsyth. The federalist revisited: New directions in authorship attribution. *Literary and Linguistic Computing*, 10:111–127, 1995.

[16] Moshe Koppel Jonathan and Jonathan Schler. Exploiting stylistic idiosyncrasies for authorship attribution. In *In IJCAIÕ03 Workshop on Computational Approaches to Style Analysis and Synthesis*, pages 69–72, 2003.

[17] P. Juola. Jgaap, a java-based, modular, program for textual analysis, text categorization, and authorship attribution.

[18] P. Juola and D. Vescovi. Empirical evaluation of authorship obfuscation using JGAAP. In *Proceedings of the 3rd ACM workshop on Artificial Intelligence and Security*, pages 14–18. ACM, 2010.

[19] Patrick Juola. Authorship attribution. *Foundations and Trends in information Retrieval*, 1(3):233–334, 2006.

[20] Patrick Juola. Authorship attribution. *Foundations and Trends in information Retrieval*, 1(3):233–334, 2008.

[21] Gary Kacmarcik and Michael Gamon. Obfuscating document stylometry to preserve author anonymity. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 444–451, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[22] Moshe Koppel and Yaron Winter. Determining if two documents are by the same author. *Journal of the American Society for Information Science and Technology*, 2013.

[23] Andrew W.E. McDonald, Sadia Afroz, Aylin Caliskan, Ariel Stolerman, and Rachel Greenstadt. Use fewer instances of the letter i: Toward writing style anonymization. In *Privacy Enhancing Technologies*, pages 299–318. 2012.

[24] F. Mosteller and D. Wallace. Inference and disputed authorship: The federalist. 1964.

[25] A. Narayanan, H. Paskov, N. Gong, J. Bethencourt, E. Stefanov, R. Shin, and D. Song. On the feasibility of internet-scale author identification. In *Proceedings of the 33rd conference on IEEE Symposium on Security and Privacy*. IEEE, 2012.

[26] Michael P. Oakes. Ant colony optimisation for stylometry: The federalist papers. *Proceedings of the 5th International Conference on Recent Advances in Soft Computing*, pages 86–91, 2004.

[27] J.W. Pennebaker, R.J. Booth, and M.E. Francis. Linguistic inquiry and word count (LIWC2007). *Austin, TX: LIWC (www. liwc. net)*, 2007.

[28] Daniele Perito, Claude Castelluccia, Mohamed Ali Kaafar, and Pere Manils. How unique and traceable are usernames? In *Privacy Enhancing Technologies*, pages 1–17. Springer, 2011.

[29] J.C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. *Advances in Kernel MethodsSupport Vector Learning*, 208(MSR-TR-98-14):1–21, 1998.

[30] John C Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. *Advances in Kernel Methods Support Vector Learning*, 208(MSR-TR-98-14):1–21, 1998.

[31] Tieyun Qian and Bing Liu. Identifying multiple userids of the same author. In *EMNLP 2013*, 2013.

[32] J.R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[33] J.R. Quinlan. *C4. 5: programs for machine learning.* Morgan kaufmann, 1993.

[34] B. Santorini. Part-of-speech tagging guidelines for the Penn Treebank Project (3rd revision). 1990.

[35] Helmut Schmid. Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*. Citeseer, 1995.

[36] Harold Somers and Fiona Tweedie. Authorship attribution and pastiche. *Computers and the Humanities*, 37:407–429, 2003.

[37] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[38] Fiona J. Tweedie, S. Singh, and D.I. Holmes. Neural network applications in stylometry: The federalist papers. *Computers and the Humanities*, 30(1):1–10, 1996.

[39] Hans Van Halteren, Harald Baayen, Fiona Tweedie, Marco Haverkort, and Anneke Neijt. New machine learning methods demonstrate the existence of a human stylome. *Journal of Quantitative Linguistics*, 12(1):65–77, 2005.

[40] James Wayman, Nicholas Orlans, Qian Hu, Fred Goodman, Azar Ulrich, and Valorie Valencia. Technology assessment for the state of the art biometrics excellence roadmap. http://www.biometriccoe.gov/SABER/index.htm, March 2009.

[41] R. Zheng, J. Li, H. Chen, and Z. Huang. A framework of authorship identification for online messages: Writing style features and classification techniques. *Journal American Society for Information Science and Technology*, 57(3):378–393, 2006.