NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

BLOG FINGERPRINTING: IDENTIFYING ANONYMOUS POSTS WRITTEN BY AN AUTHOR OF INTEREST USING WORD AND CHARACTER FREQUENCY ANALYSIS

by

David J. Dreier

September 2009

Thesis Advisor:	Craig H. Martell
Second Reader:	Andrew I. Schein

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 2009	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Blog Fingerprinting: Identifying Anonymous Posts Written by an Author of Interest Using Word and Character Frequency Analysis	5. FUNDING NUMBERS
6. AUTHOR(S) David J. Dreier	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Internet blogs are an easily accessible means of global communications. Monitoring blogs for criminal and terrorist activity is a serious challenge, due to blogs' anonymous nature and the sheer volume of data. The intelligence community is often faced with more information than it can process. The need exists to develop methods for processing the massive amounts of data this media presents, without a significant increase in manpower. An automated tool capable of indentifying posts written by an individual, given a sample of his writing, would allow law enforcement and intelligence agencies to gather evidence that would otherwise be overlooked due to manpower and time constraints.

This research focuses on identifying blog posts written by a particular author, when we do not have a model of every potential author. Previous research either builds a distinct model for every possible author, or limits itself to large documents. Neither approach is appropriate for processing blog posts. Blog posts tend to be short documents, and building a distinct model of each author is unreasonable if you are looking for one author among millions. We address this problem by combining sample posts by other authors to create a model of an "average author."

14. SUBJECT TERMS Author Attribution, Authorship Attribution, Authorship Verification, Natural Language Processing, Machine Learning, Blogs, Bayes, Bayesian, Support Vector Machine, Internet Communication	15. NUMBER OF PAGES 93
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

BLOG FINGERPRINTING: IDENTIFYING ANONYMOUS POSTS WRITTEN BY AN AUTHOR OF INTEREST USING WORD AND CHARACTER FREQUENCY ANALYSIS

David J. Dreier
Captain, United States Marine Corps
B.A., University of San Diego, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2009

Author: David J. Dreier

Approved by: Craig H. Martell
Thesis Advisor

Andrew I. Schein
Second Reader

Peter J. Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Internet blogs are an easily accessible means of global communications. Monitoring blogs for criminal and terrorist activity is a serious challenge, due to blogs' anonymous nature and the sheer volume of data. The intelligence community is often faced with more information than it can process. The need exists to develop methods for processing the massive amounts of data this media presents, without a significant increase in manpower. An automated tool capable of indentifying posts written by an individual, given a sample of his writing, would allow law enforcement and intelligence agencies to gather evidence that would otherwise be overlooked due to manpower and time constraints.

This research focuses on identifying blog posts written by a particular author, when we do not have a model of every potential author. Previous research either builds a distinct model for every possible author, or limits itself to large documents. Neither approach is appropriate for processing blog posts. Blog posts tend to be short documents, and building a distinct model of each author is unreasonable if you are looking for one author among millions. We address this problem by combining sample posts by other authors to create a model of an "average author."

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
A. MOTIVATION 1
B. ORGANIZATION OF THESIS 3

II. TOPIC BACKGROUND 5
A. HISTORY OF AUTHORSHIP ATTRIBUTION 5
B. STYLOMETRIC FEATURES 7
1. Lexical Features 7
a. Bag of Words 7
b. N-grams 8
c. Term Frequency-Inverse Document Frequency 8
2. Character Features 9
3. Syntactic Features 9
4. Semantic Features 10
5. Features Specific to Blogs 10
C. METHODS 11
1. Naïve Bayes 11
2. Probability of a Term Given an Author 12
3. Smoothing 13
4. Support Vector Machines (SVM) 15
5. Evaluation Criteria 18
a. Precision, Recall, F-score 18
b. Accuracy 19
D. APPLICABILITY TO OTHER LANGUAGES 20
E. RECENT WORK IN AUTHOR ATTRIBUTION 21
1. Author Attribution on Highly Imbalanced Classes 22
2. Author Attribution on Thousands of Candidate Authors 23
3. One Author vs. Many—Long Documents 24
F. ONE AUTHOR VS. MANY—SHORT DOCUMENTS 27

III. EXPERIMENTAL DESIGN AND METHODOLOGY 29
A. SOURCE OF DATA 29
1. The Blog Authorship Corpus 29
2. Noise in the Data: Multiple Authors 29
3. Indications of Multiple Authors 30
a. Author Signatures 30
b. High Post Frequency 32
4. Data Selection 33
a. Data Remaining after Removing Posts with Suspected Multiple Authors 33
b. Authors Chosen for Data Sets 33
B. FEATURE SELECTION 34
1. Tokenizing Words 35

2. Tokenizing Characters 35
3. Test Data Selection 35
C. NAÏVE BAYES 35
1. Bag of N-grams and Smoothing 35
2. Modeling the Other Authors 36
D. SUPPORT VECTOR MACHINE 36
1. SVM Toolset 36
2. Building the Vector Model 37
3. Modeling the Other Authors 37
E. EVALUATION CRITERIA AND BASELINE 37
1. Evaluation Criteria 37
2. Baseline 38
IV. RESULTS AND ANALYSIS 39
A. RESULTS 39
1. Summary Results 40
2. Detailed Results 42
B. ANALYSIS 45
1. Effective Features 45
a. Word Unigrams 45
b. Character Trigrams 45
2. Effectiveness of the Classifiers 45
a. Naïve Bayes 45
b. SVM 46
3. Effect of Class Imbalance 47
4. Distinctive Authors 49
a. Distinctive Misspelling 51
b. Foreign Language Characters 51
c. Single Topic 52
d. No Discernable Pattern 52
e. Multiple Authors 52
5. Effect of Quantity of Training Data 53
V. CONCLUSION AND RECOMMENDATIONS 57
A. SUMMARY 57
B. FUTURE WORK 58
1. The Class Imbalance Problem 58
2. The Effect of Topic on Author Verification 58
3. Applicability of Character N-grams to Foreign Languages 59
4. Refinements to the Classifiers 60
5. Additional Noise in the Data 60
6. Application of Koppel's Unmasking to blogs 60
APPENDIX A: SVM: CALCULATING THE HYPERPLANE 61
APPENDIX B: EXAMINATION OF 20 DISTINCTIVE AUTHORS 65

APPENDIX C: AUTHORS IN MULTIPLE DATA SETS 67
LIST OF REFERENCES..... 71
INITIAL DISTRIBUTION LIST 75

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. Linear Classification [From 31]. 16
Figure 2. Linear Separating Hyperplanes [From 29]. 16
Figure 3. Naïve Bayes: Data Set 1 (10 Authors): F-scores on Character Bigrams. 42
Figure 4. Naïve Bayes: Data Set 2 (10 Authors): F-scores on Character Trigrams. 42
Figure 5. Naïve Bayes: Data Set 3 (100 Authors): F-scores on Character Trigrams. 43
Figure 6. Naïve Bayes: Data Set 4 (1000 Authors): F-scores on Character Trigrams. 43
Figure 7. SVM: Data Set 3 (100 Authors): F-scores on Character Trigrams. 44
Figure 8. SVM: Data Set 3 (100 Authors): F-scores on Character 4-grams. 44

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Blog Files Confirmed to Have Multiple Authors 31
Table 2. Blog Files Confirmed to Have a Single Author 32
Table 3. Blog Post Frequency Statistics 33
Table 4. Authors Chosen for Data Sets 1-4 34
Table 5. Naïve Bayes: Result Averages 40
Table 6. Naïve Bayes: Data Set 3 Result Averages 41
Table 7. SVM: Data Set 3 Result Averages 41
Table 8. Proportion of Training Data per Author 47
Table 9. Naïve Bayes: Data Set 4, Distribution of Zero F-scores 48
Table 10. Example of a Distinctive Author: F-scores when Identifying the Posts written by *r2117806.male.24.Student.Aries.xml* (NB: character 3-grams, SVM: character 4-grams) 49
Table 11. Distinctive Author Characteristics 50
Table 12. Distinctive Author Characteristics by Category 50
Table 13. Effect of Class Imbalance on Authors of Data Set 1 and 2. 54
Table 14. Distinctive Authors with Less than 50 Posts 65
Table 15. Distinctive Authors with 50 to 100 Posts 66
Table 16. Distinctive Authors with More than 100 Posts 66
Table 17. Naïve Bayes: Data Set 1 (10 authors) F-scores 67
Table 18. Naïve Bayes: Subset of Data Set 3 (100 Authors) F-scores 67
Table 19. Naïve Bayes: Subset of Data Set 4 (1000 Authors) F-scores 68
Table 20. Naïve Bayes: Data Set 2 (10 Authors) F-scores 68
Table 21. Naïve Bayes: Subset of Data Set 3 (100 Authors) F-scores 68
Table 22. Naïve Bayes: Subset of Data Set 4 (1000 Authors) F-scores 69

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Many people contributed to my success in this research and I owe them my gratitude.

Dr. Craig Martell, your passion for research in this field has been an inspiration. Thank you for your guidance and encouragement, for the freedom to explore a topic that fascinated me, and for helping me to develop achievable research goals. Thank you for your encouragement when I was on-track to complete this research on time, and your admonishment when I was not. One gave the confidence I needed, the other helped me to work faster. Finally, thank you for putting up with all of my questions, and answering them, most of the time….

To my lab partners, Jon Durham, Jenny Tam, Brian Hawkins, and Johnnie Caver, thank you for keeping me sane during the many hours of study in a windowless basement. Jenny, you are the hardest worker I know. Jon, thanks for the humor and sarcasm. Brian, competing with you helped me to do my best. Johnnie, I thoroughly enjoyed our many conversations. Thank you all for the assistance you provided, in this research, and in our classes. It's been "shiny."

Laura, my wife and best friend, this thesis would not have been possible without your tremendous support. Thank you for taking care of our family during the many long hours of study and research. Your efforts kept our household running smoothly, and freed me to concentrate on this research. Your constant encouragement in the pursuit of my goals and dreams has helped me to achieve so much. I could not have done this without your support. Thank you for all that you do.

Elizabeth, my bright, beautiful girl. Your smiles and laughter warm my heart, and your excitement to see me after a long day of study always helped to revive my spirits. Thank you for enduring many weeks without seeing much of your father.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

Internet blogs are easily accessible, free, and often anonymous means of global communications. In the context of international terrorism, blogs are a potential way for terrorists to spread extremist ideology, find like-minded individuals, recruit new members, and coordinate future activities. In blogs, it would be easy for such a terrorist to compartmentalize his activities, using different blogs and different screen-names for recruiting, political discussions, and planning future attacks. Even communications regarding a single activity could be spread across a number of blog forums. In such a case, law enforcement agencies would have to find and correlate numerous disparate blogs in order to have sufficient evidence to obtain a warrant or prevent a pending attack.

Monitoring blogs and other Internet communications for evidence of terrorist or criminal activity is a serious challenge due to the anonymous nature of blogs and the sheer volume of data. The number of analysts is limited. The law enforcement and intelligence communities are often faced with much more information than they can process with the personnel available. Even once a person of interest is identified, it can be nearly impossible to find other communications written by that individual. This may make the difference between identifying an active terrorist, and letting a suspect slip away due to insufficient evidence. The need exists to develop methods for processing the massive amounts of data this media presents without a significant increase in manpower. An automated tool capable of finding posts written by an individual, given a sample of his writing, would allow law enforcement and intelligence agencies to gather evidence that would otherwise be overlooked due to manpower and time constraints.

In the absence of meta-data identifying the author of a blog post, analysis of the post's body may be our only indication of the author. When the number of potential authors is large, this is not a trivial task. When there is meta-data identifying the author of a blog, authorship attribution techniques are still useful, as they allow us to identify when an individual uses multiple aliases.

The application of machine learning techniques allows us to reduce the number of documents a human analyst must evaluate. Ideally, an automated classifier would find all the documents written by a suspect without returning any documents written by a different author; however, this is not a realistic expectation. The purpose of such an automated system would be to augment the efforts of human analysts by filtering large quantities of data to reduce the number of posts an analyst must process in order to find posts written by the target author. Analysts do not have time to read every blog post. By reducing the number of posts an analyst must read, it allows the analyst to find a greater number of the posts written by the target. Even a system that fails to identify a significant number of the target author's posts would be useful if it can eliminate enough of the posts written by other authors, so that a human analyst is able to find more posts written by the target author, in less time. Without such filtering, it is feasible that an analyst would have to process thousands of posts to find a single post written by the target author. Reducing this ratio to some reasonable threshold, say one target post for every 20 processed, would be very useful. Such a system still relies on human experts to make the final determination regarding which posts were in fact written by the target author, but it enables them to be more efficient by eliminating many of the posts written by other authors.

Traditional efforts in blog authorship identification rely on having sample works of all possible authors. These methods build a model of each possible author and, for a given document, label the most likely author, or rank order all the authors in order of probability. This is unrealistic when applied to the problem of finding posts on the Internet written by a particular individual. We do not have

a model of all the authors and, in most cases, if the suspect did not author a post, we do not care which of the other authors wrote it. This is the problem of author verification; we possess examples of the writing of a single author and we desire to determine if a text of unknown authorship was written by the same author. Limited research has been done on authorship verification. The methods developed to date are restricted to verifying the authorship of lengthy documents.

This thesis addresses the problem of author verification in short documents. We focus on identifying blog posts written by a particular author when we do not have a model of every potential author. Our approach is to combine the posts of all other authors to approximate the writing style of an "average" author. This thesis limits itself to English language blogs, however many applications require the ability to process blogs in a variety of languages. Once effective techniques are developed, future research is needed in order to test their applicability across languages. To our knowledge, this thesis is the first research to address the problem of author verification on short documents.

B. ORGANIZATION OF THESIS

This thesis is organized as follows:

- Chapter I discusses the motivation for an automated system capable of detecting short documents written by a particular author, given samples of that author's work. The differences between this and prior research is also discussed.
- Chapter II contains background information about authorship attribution research including the use of stylometric features, common experimental methods, application of these techniques to other languages and recent work addressing the more challenging problem of author verification.
- Chapter III explains the experimental design and methodology, including pre-processing of the data corpus, feature selection, predictive models, setup of the experiments and evaluation criteria.
- Chapter IV presents the results of the experiments and the analysis of the results.
- Chapter V contains concluding remarks and recommendations for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. TOPIC BACKGROUND

In this chapter, we discuss the background of authorship attribution. First, we survey the history of authorship attribution. Next, we discuss the features used to quantify an author's writing style. We then review some of the techniques that have been used for authorship attribution. Finally, the focus of this research is authorship verification; the problem of determining whether a given text was written by a particular author.

A. HISTORY OF AUTHORSHIP ATTRIBUTION

The goal of authorship attribution, sometimes known as author identification, is to use the textual features within a document to distinguish between texts written by different authors. Early attempts involved quantifying writing styles as the discriminating feature. These efforts date back to at least the 19th century, with Mendenhall's 1887 efforts of to analyze the plays of Shakespeare [1]. Studies in the 20th century used a variety of statistical models to determine the most probable author. In 1964, Mosteller and Wallace used Bayesian statistical analysis to examine the 12 "Federalist Papers," of which both Hamilton and Madison claimed to be the author. The textual features Mosteller and Wallace used were a small set of common words with little or no topical meaning. Such words are often called function words. They produced significant discriminative results between the candidate authors and concluded that Madison had written all 12 of the disputed documents [2].

Prior to the work of Mosteller and Wallace, the dominant technique in author attribution was the use of human literary experts. Since Mosteller and Wallace, research in authorship attribution has focused on 'stylometry', defining features for quantifying an author's writing style, such as, sentence length, word length, word frequencies, character frequencies, and vocabulary richness. As of 1998, nearly 1,000 different measures had been identified as features useful for defining an author's style [3].

Most efforts in authorship attribution prior to the late 1990s lacked an objective means of evaluation of the methods and techniques used. Authorship attribution was generally used to examine literary works of disputed authorship, such as “The Federalist Papers [1].” The early efforts in authorship attribution were also hindered by the lack of data in an accessible digital format. Calculating the chosen features by hand, or the need to first transcribe the documents into an electronic format, limited the scale of these experiments.

The past decade has seen an explosion in the availability of electronic documents where the author is known (e-mail, blogs, electronic books, etc.). This provides fertile ground for experiments in authorship attribution. It is now possible to run extensive experiments over large volumes of data, without the need to manually calculate the discriminative features or transcribe the documents. Since the true author of these documents is known, these experiments have an objective means of evaluation.

In the typical authorship attribution problem, we have sample works of undisputed authorship from a known set of authors and a text of unknown authorship. We wish to assign the text of unknown authorship to one of the known authors. Other authorship analysis tasks, as cited by Stamatatos in [1], include:

- Author verification (i.e., to decide whether a given text was written by a certain author or not).
- Plagiarism detection (i.e., finding similarities between two texts).
- Author profiling or characterization (i.e., extracting information about the age, education, sex, etc. of the author of a given text).
- Detection of stylistic inconsistencies (as may happen in collaborative writing).

This thesis focuses on the author verification problem applied to Internet blogs.

B. STYLOMETRIC FEATURES

Stylometric features are metrics used to measure an author's style and include lexical, character, syntactic and semantic features. Lexical and character features require minimal processing to compute; the text is simply processed as a sequence of word or character tokens. Syntactic and semantic features require linguistic analysis and advanced tools to process [1].

1. Lexical Features

Lexical features include word-length, sentence length, vocabulary richness, word frequencies, n-grams, and frequency of spelling or formatting errors [1]. Recent research has used various lexical features including sentence and word length [4]; vocabulary richness [5]; word frequencies [6], [7], [8], [9], [10]; and spelling/formatting errors [11]. Word frequencies are sometimes limited to the most frequent words to reduce the dimensionality of the model. Early research generally limited the model to no more than the most frequent 100 words, while more recent research has included every word occurring more than once in the training data [1].

a. Bag of Words

The bag of words, or unigram model, is the simplest form of measuring word frequency [1]. The frequency of each word is calculated with no regard for context or word order. They are often converted to lowercase, so two tokens differing only in capitalization contribute to increasing the count of the same word type. The number of types is the number of distinct words, while the number of tokens represents the number of word occurrences. In the sentence: "the dog bit the cat," there are five tokens (word occurrences), but only four types (distinct words): the, dog, bit, cat. Punctuation can be problematic; a period following a word at the end of a sentence is not part of the word, but a period as part of an abbreviation is. In Internet blogs, punctuation is frequently used in non-traditional ways, such as when used as emoticons.

b. N-grams

An n-gram is a continuous sequence of n words. N-grams can also be thought of as sliding windows of n consecutive words. The first four tri-grams of this paragraph are: <*An, n-gram, is*>; <*n-gram, is, a*>; <*is, a, continuous*>; and <*a, continuous, sequence*>. This captures some of the local context of the words. This is generally considered advantageous, as it captures not just the individual words, but how the author uses them. However, Stamatatos and others have cautioned that when using word n-grams, one is more likely to capture content-specific information, rather than attributes characteristic an author's style [1], [2]. The other hazard of higher order n-grams is their tendency to result in a very sparse representation of the data, since most combinations of words are rarely seen.

c. Term Frequency-Inverse Document Frequency

Term Frequency-Inverse Document Frequency (tf-idf) is a common form of a word frequency measure [12], [13]. The reasoning is that a term that occurs frequently in a document is characteristic of that document unless it occurs frequently in all documents. Thus, this technique provides a measure of the frequency of a term relative to that term's frequency in all documents.

tf-idf = tf x idf

tf = term frequency = $\frac{n_{i,j}}{\sum_k n_{k,j}}$

Where $n_{i,j}$ is the number of occurrences of term i in document j and the denominator is the sum of the number of terms in document j [12].

idf = inverse document frequency = $\log \frac{|D|}{|\{d : t_i \in d\}|}$

Where D is the total number of documents and the denominator is the number of documents in which term i appears [12].

2. Character Features

When using character features, tokenization is easy, since the text is simply tokenized at each character. Character features include character types, “alphabetic character count, digit characters count, upper and lower case character counts, letter frequencies, punctuation mark counts [1],” and character n-grams. Some researchers consider character features a type of lexical feature [14], while others, such as Stamatatos, put them in their own category. Character features have been shown to be an effective approach to authorship attribution. In particular, character n-grams work well in texts with noisy texts containing frequent grammatical errors or unique use of punctuation, such as in blogs and other Internet communications [1]. Examples of research using character features include [15], [16], [17], [18] and [19].

3. Syntactic Features

Syntactic features include part-of-speech, sentence and phrase structure, and frequency of syntactic errors. Syntactic features are more reliable than lexical features as an indication of authorship, however, syntactic feature extraction “requires robust and accurate NLP [Natural Language Processing] tools to perform syntactic analysis of the text [1].” In many cases sentence splitting, part-of-speech tagging, text chunking, and partial parsing can be done accurately; however, the effectiveness of these tools, and the accuracy of their results, varies from language to language and domain to domain. For example, effective part of speech taggers have yet to be developed for Chinese [20]. In addition, these tools often require annotated data in a specific domain to be effective. A tool trained in one domain, such as Wall Street Journal articles, will lose much of its effectiveness if it is applied directly to another domain, such as Internet chat, without training on annotated data from the new domain [21]. Annotating data is a labor intensive and time-consuming task.

4. Semantic Features

Semantic features include synonyms, semantic dependencies and function words [1]. The simplest semantic feature is the use of function words. Function words are common words with no contextual meaning, such as, “and”, “to”, etc. These are often hand-selected using arbitrary criteria based on language-dependant expertise [1]. There have been few attempts to use higher-level semantic features because the more complex forms of semantic analysis, “such as full syntactic parsing, semantic analysis, or pragmatic analysis, cannot yet be handled adequately by current NLP technology for unrestricted text [1].”

5. Features Specific to Blogs

Blogs, short for web-logs, are online forums where individuals post free-form messages. Typical message lengths range from a few words to several pages. Some blogs restrict the posts to a single author, some allow others to comment on the primary authors posts, and some allow multiple users to post to the same blog. Some blogs focus on a particular topic or discussion while others serve as a personal, although public, diary.

Blogs and other online communications tend to be less formal than other forms of writing, resulting in more misspellings, abbreviations, unorthodox structures, and creative use of punctuation, i.e., emoticons. Consequently, blog data tends to be extremely noisy [14]. This makes the task of parsing and computing some types of features more challenging, but has the potential to increase the accuracy of authorship attribution techniques, since authors works are not constrained by a rigid style or modified by an editor.

Online communications often have additional features that can be useful for author attribution. [14] suggests the use of “unique structural characteristics” found in online communications, such as greetings, signatures, quotes, and contact information as well as technical features, such as fonts, embedded images, and hyper-links. However, these features can be problematic because they are not available in all cases and may change frequently if the author is

trying to mask his identity. Greetings, signatures and quotes are particularly difficult to identify automatically because of the inconsistent ways in which people use them. For example, a post ending in **_Lincoln_** could indicate a signature, but it might indicate the end of a quotation. Quotations are difficult to detect because quotation marks are often omitted. Signatures are sometimes preceded by special characters, but sometimes not. Signatures can consist of a single word, a few words or contain an entire phrase. Even authors who are not attempting to mask their identity often vary their signature. For example, in our blog data, we observed one author who signed her posts "Kathy," "Kath" or "Kat"; another author signed her posts "Sabrina_C," "Sabrina," "Sabrina See" or "S."

C. METHODS

Although a large variety of machine learning methods have been applied to authorship attribution, Naïve Bayes and linear Support Vector Machines (SVM) have both shown themselves to work well for classifying text documents into distinct classes [22]. According to [1], SVMs are considered "one of the best solutions of current technology."

1. Naïve Bayes

Naïve Bayes classifiers use Bayes rule to estimate the probability of a class, given some set of features. The features, F, are often conceptualized as a vector of counts. In the case of authorship attribution, the classes are authors ($a_i \in A$). Bayes Rule is as follows:

$$P(a_i \mid F) = \frac{P(F \mid a_i)P(a_i)}{P(F)}$$

Given a set of potential authors A, the most likely author, a^*, is the one with the highest probability:

$$a^* = \arg\max_{a_i \in A} \left[\frac{P(F \mid a_i)P(a_i)}{P(F)} \right]$$

The unconditional probability of the feature vector, $P(F)$, is constant from one author to the next given the feature vector F. Therefore, this term can be omitted, without changing the rank ordering of the authors.

$$a^* = \arg\max_{a_i \in A} \left[P(F \mid a_i) P(a_i) \right]$$

Naïve Bayes classifiers make the assumption that each element of the feature vector is independent of every other element, which allows us to compute the probability $P(F \mid a_i)$ by taking the product of each term, f_j, given author a_i. The conditional probability $P(f_j \mid a_i)$ is estimated from the author's training data. Thus,

$$P(F \mid a_i) = \prod_{f_j \in F} P(f_j \mid a_i)$$

and

$$a^* = \arg\max_{a_i \in A} \left[P(a_i) \prod_{f_j \in F} P(f_j \mid a_i) \right]$$

However, these probabilities quickly become too small to represent accurately in a computer. By taking the log of the probabilities, we maintain the rank order of the authors and avoid the problem of underflow.

$$a^* = \arg\max_{a_i \in A} \left[\log P(a_i) + \sum_{f_j \in F} \log P(f_j \mid a_i) \right]$$

2. Probability of a Term Given an Author

For unigrams, the probability of a term, w_j, given an author a_i is:

$$P(w_j \mid a_i) = \frac{C(w_j)}{N}$$

where $C(w_j)$ is the count of word w_j and N is the total number of words (tokens) seen in the training data for author a_i.

For n-grams, the probability of a word is conditional on the $n-1$ word(s) preceding it.

$$P(w_j \mid a_i, w_{j-n+1}...w_{j-1}) = \frac{C(w_{j-n+1}...w_j)}{C(w_{j-n+1}...w_{j-1})}$$

Where the numerator is the number of times we have seen this n-gram in the training data for author a_i and the denominator is the number of times we have seen the preceding $n-1$ words in this author's training data.

3. Smoothing

The shortfall of Naïve Bayes classifiers is that if a term has not been seen in the training data for an author, the probability of that term, given that author, is zero. This makes the probability of the entire document zero. This is not realistic; it is likely that we did not obtain the author's full vocabulary in our training data. One approach to solving this problem is smoothing; taking some small probability mass from the terms we have seen, and distributing it to the terms we have not seen.

The simplest form of smoothing is Laplace, or add-one, smoothing. Laplace smoothing adds one to every count. Any term not seen, is treated as having a count of one. All terms that were seen are treated as having a count one higher than they did. Unfortunately, Laplace smoothing moves too much probability mass to the zero count events, and does not perform as well as other methods of smoothing [12].

Witten-Bell smoothing, generally referring to "method C" in [23], outperforms Laplace smoothing and remains relatively simple to implement [24]. The formula for Witten-Bell smoothing appears in various forms in [23], [24], [25], [26] and [27]. Witten-Bell smoothing estimates the probability of an unseen word based on the frequency that we have seen new words in the past [24]. The unigram formula, from [23], is:

$$P_{WB}(w_i) = \frac{C(w_i)}{N+T} \qquad \text{if } C(w_i) > 0$$

$$P_{WB}(w_i) = \frac{T}{N+T} \times \frac{1}{Z} \qquad \text{if } C(w_i) = 0$$

$C(w_i)$ = the count of word w_i (number of tokens)

T = the number of distinct words (types).

N = the total number of word tokens seen

Z = the estimate number of unseen words.

Without the Z term, the second formula is the total probability mass assigned to unseen words. The Z term is used to determine how much probability each occurrence of a new word is assigned. All the above counts refer to what has been seen in the training data for this author.

The formula for bigrams, from [26], is:

$$P_{WB}(w_i \mid w_{i-1}) = \frac{C(w_{i-1}w_i)}{N(w_{i-1}) + T(w_{i-1})} \qquad \text{if } C(w_{i-1}, w_i) > 0$$

$$P_{WB}(w_i \mid w_{i-1}) = \frac{T(w_{i-1})}{N(w_{i-1}) + T(w_{i-1})} \times \frac{1}{Z(w_{i-1})} \qquad \text{if } C(w_{i-1}, w_i) = 0$$

$C(w_{i-1}, w_i)$ = count of bigrams consisting of word w_{i-1} followed by word w_i.

$T(w_{i-1})$ = Number of distinct words (types) seen to the right of word w_{i-1}.

$N(w_{i-1})$ = Total number of words (tokens) seen to the right of word w_{i-1}.

$Z(w_{i-1})$ = Estimated zero counts; the number bigrams starting with w_{i-1} that do not occur in the training set. If V is the number of words (unigram types) in the vocabulary, then $Z(w_{i-1}) = V - T(w_{i-1})$.

The bigram formula can easily be extended to arbitrary length n-grams, by replacing (w_{i-1}) with $(w_{i-n+1} \ldots w_{i-1})$.

The disadvantage of the bigram and n-gram versions of Witten-Bell smoothing is that, if the preceding words $(w_{i-n+1} \ldots w_{i-1})$ do not occur in the training data, the smoothed probability is zero [26]. In other words, if we have never seen the preceding terms, the number of words (tokens) and distinct words (types) seen following those terms is zero (N = T = 0). This problem can be solved with back-off, such as in the formulas described in [24] and [25], however this adds to the complexity of the implementation. Other smoothing techniques using back-off, such as Katz and Kneser-Ney, outperform Witten-Bell [24].

4. Support Vector Machines (SVM)

The following section is derived from the discussion on Support Vector Machines in [28], [29] and [30]. An SVM attempts to find a line or hyperplane separating two classes of data. We use an SVM to separate the posts written by the target author from the posts written by all other authors. When the classes are not linearly separable, i.e., when there is no line or hyperplane capable of separating them, a kernel function is often used to transform the data. Although there are different types of SVM kernels, this research used a linear kernel, which does not transform the data. All authors other than the target were considered members of the same class. The posts of each author were represented by an n-dimensional vector, where n is the number of terms found in the training data at least twice (any terms found only once in the training data were discarded). The SVM generates a hyperplane separating the vectors of one class from the vectors of the other class in n-dimensional space.

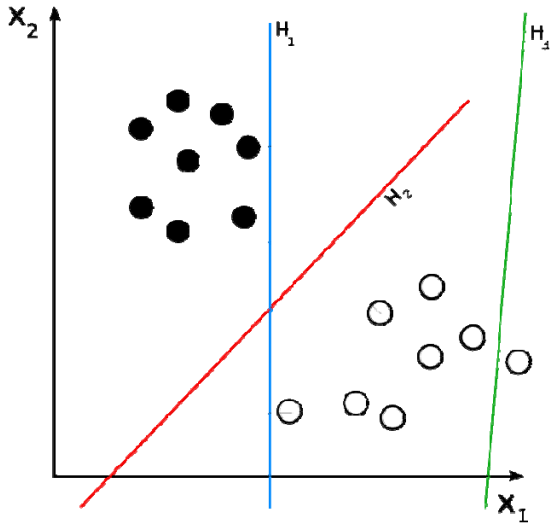

Figure 1. Linear Classification [From 31].

Based on the training data, a SVM will attempt to find the hyperplane that separates the classes with the largest distance (margin) between the hyperplane and the closest data point. This is called the maximum margin hyperplane. Thus, SVMs are known as maximum margin classifiers. In Figure 1, both lines H1 and H2 separate the two classes, but line H2 separates the classes with the maximum margin. If the classes are not linearly separable, the maximum margin hyperplane does not exist. The data points on the margin are called support vectors. Figure 2 is an example of a hyperplane that creates the maximum margin between classes. The support vectors are circled.

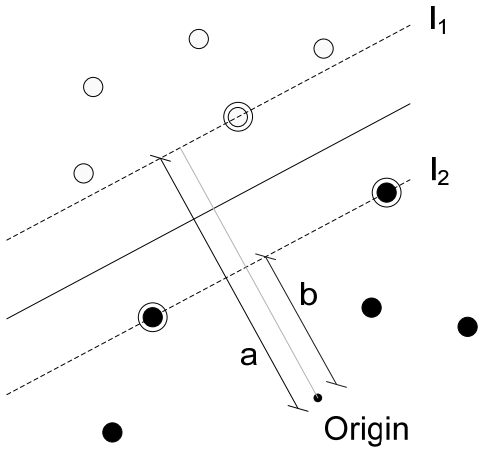

Figure 2. Linear Separating Hyperplanes [From 29].

The equation for a general hyperplane is $\vec{w}^T\vec{x}+b=0$ [28], where $\vec{w}$ is a vector of weights representing the significance of the terms, $\vec{x}$ is a data point (also a vector) and b is a constant. For the hyperplane to separate the data into distinct classes, its equation should be $\vec{w}^T\vec{x}_j+b>0$ for all the $\vec{x}_j$ of one class and $\vec{w}^T\vec{x}_k+b<0$ for all the $\vec{x}_k$ of the other class [28]. Let the training points be labeled as $y_i\in\{-1,1\}$, with $+1$ being a positive example (e.g., target author) and -1 being a negative example (e.g., not the target author), so the hyperplane can be defined as

$y_i(\vec{w}^T\vec{x}_i+b)>0$ for all data points i.

Note that $\frac{b}{\|\vec{w}\|}$ determines the offset of the hyperplane from the origin, along the vector $\vec{w}$. Thus $\vec{w}$ and b can be scaled without changing the hyperplane; we chose to scale them such that:

$y_i(\vec{w}^T\vec{x}_i+b)\geq 1\ \ \forall i$.

Next, we define an expression to describe the border of the margins; these can be thought of as additional "supporting hyperplanes," parallel to the separating hyperplane, as depicted by l_1 and l_2 in Figure 2. These supporting hyperplanes will pass through those data points closest to the separating hyperplane. Such data points are known as support vectors. The formula for the supporting hyperplanes are $y_j(\vec{w}^T\vec{x}_j+b)=1$ and $y_k(\vec{w}^T\vec{x}_k+b)=1$ for some points j,k, where $j\in\{\text{positive training examples}\}$, $k\in\{\text{negative training examples}\}$ and $j,k\in\{\text{support vectors}\}$ [28]. Recall that $y_i\in\{-1,1\}$ is the label for the classes, with $+1$ indicating a positive example, and -1 indicating a negative example. Therefore, $y_j=+1$, and $y_k=-1$. There may be multiple support vectors along each of these hyperplanes. We will use these supporting hyperplanes to determine an expression for the width of the margin, that is, the distance between the separating hyperplane and the closest data points. We choose $\vec{w}$

and b to maximize this distance [28]. The calculations for determining $\vec{w}$ and b, and thus defining both the hyperplane and the width of the margin, are presented in Appendix A.

In many cases, the two classes are not linearly separable; it is not possible to separate them with a hyperplane. In such cases we desire to find a hyperplane that separates the classes as well as possible with the fewest errors. This is done by defining "slack variables," s_k, to represent the allowable deviation from the margin, thus relaxing

$$y_k(\vec{w}^T\vec{x}_k + b) \geq 1 \quad \text{to} \quad y_k(\vec{w}^T\vec{x}_k + b) \geq 1 - s_k .$$

Thus, allowing points to be s_k distance on the wrong side of the hyperplane. To prevent large slack variables from allowing any line to partition the data, we add another term to the Lagrangian to penalize large slacks [28], [30]. The Lagrangian equation is used in Appendix A to calculate the hyperplane with the maximum margin. The Lagrangian equation without slack variables is:

$$L_p = \frac{1}{2}\vec{w}^T\vec{w} - \sum_k \lambda_k[y_k(\vec{w}^T\vec{x}_k + b) - 1]$$

Adding the slack variable and an additional term to penalize large slacks, the formula becomes:

$$L_p = \frac{1}{2}\vec{w}^T\vec{w} - \sum_k \lambda_k[y_k(\vec{w}^T\vec{x}_k + b) + s_k - 1] + \alpha\sum_k s_k .$$

5. Evaluation Criteria

a. Precision, Recall, F-score

Precision is the proportion of selected items that were correct (i.e., of all the posts whose classifier labeled was written by the target author, the percent that actually were written by the target). Recall is the proportion of target items the system selected (i.e., of all the posts actually written by the target

author, the percent the classifier correctly labeled as written by the target). The F-score is a means to combine precision and recall [25].

F-score is the harmonic mean of precision and recall. It is more heavily weighted toward the smaller of the two scores; thus penalizing a classifier that boosts one score at the expense of the other.

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$f - score = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

TP = true positives: written by the target & labeled “target”

FP = false positives: not written by the target & labeled “target”

FN = false negatives: written by the target & labeled “not target”

b. Accuracy

Accuracy is the percentage of the test documents that were correctly labeled. Accuracy is useful when there are more than two classes, since the notion of false positives does not apply in such situations. The formula for accuracy in a two-class problem is

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

TP = true positives: written by the target & labeled “target”

FP = false positives: not written by the target & labeled “target”

FN = false negatives: written by the target & labeled “not target”

TN = true negatives: not written by the target & labeled “not target”

However, accuracy is not an informative measure when the classes are highly imbalanced [25]. If the target author wrote 100 of the documents in a 100,000 document corpus, a classifier that correctly labeled 99,000 of the documents not written by the target and one document written by the target would have

TP = 1

FP = 900

FN = 99

TN = 99,000

This results in an accuracy of 0.990010 but an F-score of 0.001998. The accuracy measure gave it an outstanding score, even though it missed nearly all the documents we were interested in, because the number of documents not written by the target author dwarfed the number of documents written by the target. In fact, in the above scenario, a classifier that labeled all documents as not written by the target would have an accuracy of 0.999, but an F-score of 0.000.

D. APPLICABILITY TO OTHER LANGUAGES

Taking authorship attribution techniques developed on one language and applying them to another presents additional challenges. Techniques that work well in one language do not always transfer well to another. [14] asserts that Arabic, as a Semitic language, possesses characteristics that can make authorship attribution more difficult. Due to the orthographical and morphological properties of Arabic, many typical lexical features become more sparse as each word can take on numerous forms, reducing the effectiveness of these features [14]. In many online forums, Arabic writings are missing the diacritics, the markings above or below the letters [14]. Without diacritics, it becomes impossible to distinguish between some words [14]. This degrades the effectiveness of features such as function words, which can no longer be

distinguished without understanding the semantic context of the sentence. Semantic tagging of the data would then be necessary to distinguish these features. Arabic words are shorter than English words, but are sometimes elongated for stylistic purposes, making word length features difficult to apply effectively [14]. In [14], Abbasi and Chen used a complex set of 305 features for their English data, including 87 lexical, 158 syntactical, 45 structural, and 11 content-specific features. They used 422 features for their Arabic data, including 79 lexical, 262 syntactic, 62 structural, and 15 content-specific. The technical features of font color, font size, embedded images, and hyperlinks, were used for both languages.

In some cases, a simple set of features transfers well from one language to another. In [8], Koppel, Schler and Bonchek-Dokow demonstrated that techniques developed on English literature worked equally well when applied to 19th century and late 20th century rabbinical letters written in Hebrew-Aramaic, which is also a Semitic language. Koppel et al. used a much simpler feature set than Abbasi and Chen, the word counts of the 250 most frequent words in the document.

E. RECENT WORK IN AUTHOR ATTRIBUTION

Challenges to author attribution, when applied to Internet blogs, include short messages length, a practically unlimited number of potential authors, and highly imbalanced classes between prolific and non-prolific authors. Ample evidence exists that we can overcome the challenge of performing author attribution on short documents. [2], [10], [13] and [14] performed successful experiments on blogs and Internet forums, which tend to consist of short posts.

Gehrke [2], [10], had some success addressing the issue of highly imbalanced classes, as did [13] in addressing the issue of a large set of potential authors. These papers are discussed in more detail below.

In the “one vs. many” problem of author verification, we have example writings of a single author and we are tasked with determining if this author is responsible for a document of unknown authorship. This is especially challenging because we lack a representative sample of all works not written by the author. The research on this problem is limited, but Koppel and Schler developed a technique that produced impressive results on long documents [8], [22]. This problem is even harder when applied to small documents, and has yet to be solved by the learning community.

1. Author Attribution on Highly Imbalanced Classes

When some authors are significantly more prolific than others, the performance of most classifiers is significantly degraded. In probabilistic classifiers, such as Naïve Bayes, the prior probability of a prolific author is large enough that even very distinctive posts by other authors are unlikely to overcome the prior probability. Similarly, in instance-based classifiers, such as SVMs, the large number of examples from a prolific author makes it less likely the classifier will be able to cleanly separate this class from authors with relatively few examples. When the data is imbalanced in this way, almost all documents are assigned to these few prolific authors, and few or no documents are assigned to the less prolific authors.

In [2], [10], Gehrke introduces a post-classification corrective scaling technique to compensate for the over-classification of documents to the most prolific authors. This successfully mitigated this problem. Gehrke’s experiments used a Naïve Bayes classifier, on word bigrams, to identify the most likely author of a blog post from a set of 2000 blog authors. Gehrke’s classifier assigned probabilities to each author for a given test post and then rank ordered the authors by their probability. Some of the authors were significantly more prolific and, before the corrective scaling, the prolific authors were returned as the most likely authors, regardless of the characteristics of the test posts. The prior

probability of the prolific authors was too large for characteristics of the individual posts to overcome. The corrective scaling improved the accuracy of the classifier from 11% to 74% (on blogs with 1000 bigrams).

Gehrke's method defined success to be when the actual author was ranked within the top $n\%$, thus requiring that "the author be in some small subset of the original search space, rather than requiring that he or she be the single most probable author [2]." With this approach, he was able to achieve high accuracy while significantly reducing a search space of 2000 authors. When insisting the true author be ranked first or second the accuracy increased from 74% to 81%. When relaxing the constraint to the top 1%, he achieved an accuracy of 91.1% and reduced a search space of 2000 authors down to 20 authors using blogs containing 1000 bigrams. Gehrke also demonstrated the effect of classifying smaller posts. When only 500 bigrams were present, the accuracy of reducing the search space from 2000 authors to 20 authors was 80.4%.

2. Author Attribution on Thousands of Candidate Authors

One of the limitations in automated authorship attribution is that, as the number of potential authors increases, it becomes computationally prohibitive to construct a language model for each of the authors. The expansion in electronic media has provided ample data for experiments in authorship attribution, but it has also pushes the limits of computational feasibility. In blogs, it is possible to have data sets composed of tens or hundreds of thousands of authors.

In [13], Koppel, Schler, Argamon, and Messeri demonstrated that information retrieval techniques can be used to successfully discriminate among a set of 10,000 authors. From each of the 10,000 blogs in their test data, they removed the last 500 words, which they refer to as "snippets." They then tried to determine to which of the 10,000 authors each snippet belonged.

Koppel et al., used three feature sets to represent the data: content tf-idf (tf-idf restricted to content words), content idf (binary idf restricted to content words), and style tf-idf (function words and strings of non-alphanumeric characters). For each feature set, they used a cosine measure to quantify similarity between a particular snippet and a candidate author. They then ranked all the authors by similarity within each feature set. Koppel et al. used an SVM to evaluate 18 meta-features, in order to determine when the top ranked author was likely to be the true author. The meta-features included "the absolute similarity of the snippet to the top-ranked author, the gap in degree of similarity between the top-ranked author and the k-ranked author, the rank of the top-ranked author and the k-ranked author, the rank of the top-ranked author using the other two representation methods and so forth [13]." When the SVM indicated the top ranked author was likely to be the true author, they labeled it a "successful pair." The SVM was trained on an additional 8000 blogs held out for this purpose. If none of the feature sets returned a successful pair, or if two of the feature sets were deemed successful by the SVM but had conflicting top ranked authors, their classifier returned "Don't know." Otherwise, it returned the top ranked author. Their classifier returned an author 31.3% of the time. Of these, it was correct on 88.2% of them.

3. One Author vs. Many—Long Documents

In the traditional authorship attribution problem, we have sample works from all possible authors, and we attempt to determine which of these known authors is responsible for an anonymous text. In author verification, we have the sample works of a single author, and attempt to determine if texts of unknown authorship were written by this author. Without a closed set of alternatives, we do not have a clear way to model all the other authors' works. "As a categorization problem, [author] verification is significantly more difficult than attribution and little, if any, work has been performed on it in the learning community [22]." Koppel and Schler address the problem of author verification in long documents with impressive results, achieving an accuracy of 99.0% [22].

According to Koppel and Schler, this is essentially a one-class problem with two important distinctions. First, in author verification, we are not lacking negative examples. Quite the opposite, almost nothing was written by this author. On the other hand, the negative examples are generally not representative of all documents not written by the target author. The second distinction they make is to restrict themselves to long documents, which they divide into sub-documents, to have multiple examples that are either all written by the author, or all not written by the author. This is a significant difference from this thesis, where we place no such restrictions on the data. Koppel and Schler then ask, "whether these sets were generated by a single generating process (author) or by two different processes [22]."

In [22], Koppel and Schler introduced a technique they call "unmasking," which tests the ability of a linear classifier to distinguish between a known author's works and an anonymous document while iteratively removing the most discriminating features. Documents written by the same author quickly become indistinguishable after a few iterations. Documents written by a different author remain distinguishable much longer.

In [8], Koppel, Schler and Bonchek-Dokow extend these results to show this method remains effective when the works of the author are of varied topics. Their methods correctly classified a single author writing on multiple topics (labeled "same-author"), and multiple authors writing on a single topic (labeled "different-author").

Koppel and Schler restricted the data to long documents of 19,000 words or more (estimated from the number of 500-word chunks reported in [22]). They used a collection of 21 electronic books written by ten authors, resulting in 20 same-author pairs, and 189 different-author pairs. They subdivided each document into chunks of 500 words or more, without breaking up paragraphs. Doing so gave them "multiple examples which are known to be either all written by the author, or all not written by the author [22]." They chose to use an SVM with a linear kernel on the 250 most frequent words for each author and the test

book (weighted equally). For each pair, they trained the classifier over all the known works of the author (minus the book being tested if it was written by this author) and the unknown book, and used ten-fold cross validation to determine an accuracy score for that author-book pair. They then iteratively removed the three most strongly weighted positive features and the three most strongly weighted negative features, re-ran the SVM classifier and determined a new accuracy score. Nineteen of the same author pairs were correctly classified, as were 181 of 189 of the different author pairs, an accuracy of 95.7% (F-score of 0.809), using only the sample works of one author and a document to be tested, that is, without example works of other authors.

Koppel and Schler mention another possible approach; combine the works of a number of other authors and use them to learn a model of author A vs. not author A. They label this approach as problematic, for the following reason: if most of the examples from a text are assigned by the model to not-A, it is reasonable to conclude the text is probably not from that author. However, even if all the examples from a text are assigned by the model to A, it is not safe to conclude that A is the author. They assert that it is often the case where the text in question was written by another author possessing a similar style. Thus, this approach is reasonably accurate when it indicates the text was not written by this author, but not reliable when it indicates the text was written by this author.

Koppel uses this approach to augment his unmasking classifier by allowing the negative evidence to overrule the unmasking classifier when the negative examples indicate the text was not written by the author. When the unmasking classifier already labeled the text as not written by the author, the classifier using the negative examples is ignored. This resulted in correct classification of 18 of the 20 same-author pairs, and all 189 of the different-author pairs. The augmented classifier generated one additional false negative, but, eliminated all eight false positives, resulting in an accuracy of 99.0% (F-score of 0.947).

F. ONE AUTHOR VS. MANY—SHORT DOCUMENTS

As evidenced by [8] and [22], detecting the works of a single author can be done very effectively, even without the use of negative examples. This task is much more difficult when applied to short documents. The approach in [8] and [22] relied on the data consisting of lengthy documents. Their shortest document contained 38 500-word chunks (at least 19,000 words). They sub-divided the unknown document to produce a set of text chunks that are either all from the known author, or all not from the known author. They randomly discarded text chunks from the larger of the two classes (known author or unknown document) until they were left with the same number of text chunks in each of the two classes. They were able to divide the unknown document into enough 500-word chunks that they were able to address the problem as if it were a balanced two-class problem. This is not possible with shorter documents. This thesis does not restrict the size of the documents. The data we are working with consists of blog posts, some of which contain fewer than 20 words. The majority of the posts contain fewer than 2000 words.

Koppel and Schler indicate that the approach of combining the works of other authors in order to build a model of author A vs. not-A is unreliable when it labels a document as written by the author, but generally reliable when it labels a document as not from the author. This is useful for this thesis, as the primary application is to effectively eliminate as many documents not written by the author as possible, in order to reduce the number of documents a human analyst must process in order to find documents written by the target author.

The other significant difference between the work in [22] and [8], and this thesis, is that the set of books used by Koppel and Schler for their experiments resulted in only slightly imbalanced classes (5 to 1 in the worst case), which they balance by randomly removing data from the larger class [8]. The author verification problem, when applied to blogs, becomes extremely imbalanced. When using 1000 sample authors, the average class imbalance is 1000 to 1. The class imbalance for a particular author varies, depending on the prolificacy of

the author. In general, the number of “other” blog authors is unbounded. To our knowledge, this thesis is the first research to address the problem of author verification on short documents.

III. EXPERIMENTAL DESIGN AND METHODOLOGY

A. SOURCE OF DATA

1. The Blog Authorship Corpus

Schler, Koppel, Argamon and Pennebaker developed the Blog Authorship Corpus by collecting the blogs of more than 19,000 authors [32]. Schler et al. collected the posts from blogger.com in August 2004. Each blog is stored as a separate file, the name of which indicates the user's numeric blogger ID, self reported gender, age, industry and astrological sign. Each blog contains at least 200 occurrences of common English words. All formatting was removed, except for date tags indicating the date of each post. Hyperlinks within the body of the post were replaced with the label "urllink". The above information was obtained from [33], which also includes a link to download the corpus. The copy of the corpus, used by this thesis, was downloaded by Gehrke for his work in [2]. Gehrke reformatted the date tags, changing the format from alphanumeric, "31,May,2004," to purely numeric, "20040531." To reflect the reformatted tags, an 'r' was added to the beginning of each file name.

2. Noise in the Data: Multiple Authors

The majority of the files in the corpus contain posts by a single author. However, during our research, we discovered a blog containing posts by multiple authors. This file contained posts where the authors regularly signed their posts with distinct names, including names of both genders. Using the Google search engine, we were able to find a copy of some of these posts. The posts were no longer on www.bogger.com, but the Google search engine had cached copies of at least 15 of these posts [34], [35]. In the cached HTML pages, each post was followed by a "posted by" tag, containing the author's user-name. The user-names were linked to current profiles of the authors. The profiles were dissimilar, indicating different genders, occupations, interests, and often including a picture of the author. The "posted by" tags confirmed that seven of the 15 posts were

written by distinct authors. The “posted by” tag is also present in most single-author blogs currently posted on www.blogger.com [36]. The Blog Authorship Corpus did not retain this tag for any of the blogs.

3. Indications of Multiple Authors

a. Author Signatures

We observed that, when authors signed their posts, many preceded their signatures with at least one exclamation mark, tilde, asterisk, or dash [!~*-]. We chose to define a signature as one or more of these characters, followed by at least one alphanumeric character. We wrote a Java program using the regular expression *“.*\\s[!~*-]+\\s*\\w+\\s*”* to identify posts ending with a signature. We categorized the blogs of the corpus into one of four categories: “signed,” “conflicted,” “some signed,” or “unsigned.” Blog categorized as “signed” indicate every post ended with a signature and all signatures were identical. Conflicted blogs are those with at least two posts containing non-identical signatures. In blogs categorized as “some signed,” not all posts contained signatures, but all signatures within the blog were identical. The “unsigned” category applied to blogs where no posts ended in a signature.

Only 33 blogs were categorized as signed. These blogs did not have enough posts to be useful for this research. Only one of these had more than 13 posts; most had fewer than five posts per author. We examined 22 of the blogs categorized as conflicted, some signed, or unsigned. The blogs we examined and found multiple authors are listed in Table 1. The blogs we examined that appeared to be written by a single author are listed in Table 2.

Verified Multi-author Files	**Posts**	**Days**	**Avg Posts/ day**	**Max Posts/ day**	**Notes**
Conflicted Signatures					
r1019224.female.27.RealEstate.Libra	125	38	3.29	14	Numerous conflicting signatures
r2032593.male.24.Arts.Libra	125	38	3.29	14	Identical to r1019224
r1713845.male.23.Student.taurus	127	40	3.18	14	Identical to r1019224 but with 2 extra posts)
r3639430.female.14.indUnk.Capricorn	127	40	3.18	14	Identical to r1713845
r1119650.female.23.Student.Cancer	733	382	1.92	14	125 posts identical to r1019224, then individual: Gwen/the diva
r1417798.female.35.indUnk.Scorpio	1047	268	3.91	51	Signatures: Ellen and Melissa
r1432406.female.16.indUnk.Gemini	1045	316	3.31	18	Signatures: Kelly and Rachael
r1786023.female.16.indUnk.Libra	177	104	1.70	25	Signatures: Kath, Sandra, Kat, Kira, Ben
Some Signed					
r3868272.female.17.Arts.Sagittarius	174	47	3.70	31	Signatures: Desteny, Annie, Rachel, Ian Signatures not in a format that was easy to automatically detect. Found this blog due to # of posts/day.

Table 1. Blog Files Confirmed to Have Multiple Authors

We examined 13 “conflicting signature” blogs, and found that eight of the 13 were clearly written by multiple authors. The other five blogs appeared to be written by a single author, but they matched our pattern for conflicting signatures. In some cases, the authors used different variations of the same signature. In other cases, they ended their posts with a quote; the citation following the quote often matched our pattern for a signature.

Verified Single Author Files	**Posts**	**Days**	**Avg Posts/ day**	**Max Posts/ day**	**Notes**
Conflicted Signatures					
r1011311.female.17.indUnk.Scorpio	294	254	1.16	5	
r1015556.male.34.Technology.Virgo	386	201	1.92	7	
r1026443.female.15.Student.Scorpio	2	2	1.00	1	
r1028027.female.16.indUnk.Libra	63	50	1.26	6	
r1031806.male.17.Technology.Sagittarius	408	276	1.48	8	
Some Signed					
r1008329.female.16.student.Pisces	116	65	1.78	6	
r1015252.female.23.indUnk.Pisces	123	62	1.98	7	
r1040084.male.17.indUnk.Taurus	34	28	1.21	4	
r576311.female.34.indUnk.Capricorn	1327	100	13.27	1190	There was an error in the date stamps
r942828.female.34.indUnk.Cancer	2068	575	3.60	18	
None Signed					
r1000331.female.37.indUnk.Leo	13	10	1.30	4	
r1000866.female.17.Student.Libra	771	348	2.22	19	
r1013637.male.17.RealEstate.Virgo	512	372	1.38	7	

Table 2. Blog Files Confirmed to Have a Single Author

Of the six blogs we examined in the “some signatures” category, only one contained multiple authors. This file had four distinct signatures, but was not detected, because the style of the signatures differed from the pattern we used to define a signature. We examined three blogs with no detected signatures, and none had indications of multiple authors.

b. High Post Frequency

We discovered a trend in the blogs we examined. Blogs written by multiple authors tended to have a higher post frequency, both in terms of average posts per day, and in terms of maximum posts in a single day. We used this as additional criteria to eliminate blogs likely to contain multiple authors. The post frequency of the blogs we examined are listed in Tables 1 and 2.

4. Data Selection

a. Data Remaining after Removing Posts with Suspected Multiple Authors

We chose to eliminate any blog containing conflicting signatures, containing an average post frequency greater than two posts per day, or containing more than 11 posts in a single day. We did not use the blogs categorized as "signed," because they did not contain a sufficient number of posts per author. Table 3 lists the total number of blogs and posts in each category. Table 3 also lists the number of blogs, posts, and average posts per blog after removing blogs exceeding the post frequency thresholds. We noted that more than 40% of blogs containing conflicting signatures also exceeded the post frequency thresholds.

Category	Total Words	Total Posts	Total Blogs	Blogs Over Threshold*	Percent of Blogs Over Threshold	Remaining Blogs	Remaining Posts	Average Posts/Blog (rem. blogs)
Signed	34,399	128	33	2	6.1%	31	117	3.77
None Signed	100,462,320	456,332	17,500	3,337	19.1%	14,163	283,164	19.99
Some Signed	17,857,344	100,369	1,228	364	29.6%	864	42,072	48.69
Conflicted	18,471,658	124,447	559	244	43.6%	315	32,870	104.35
Total:	**136,825,721**	**681,276**	**19,320**	**3,947**	**20.4%**	**15,373**	**358,223**	**176.81**

*Threshold: average post freqency > 2.0 posts per day or > 11 posts in one day indicates possible multiple authors.

Table 3. Blog Post Frequency Statistics

b. Authors Chosen for Data Sets

We used four subsets of the Blog Authorship Corpus for our experiments. We chose these data sets to test the effect of various levels of class imbalance. All four of these subsets were chosen from the blogs categorized as "some signatures" or "unsigned." Two of the subsets, Data Set 1 and Data Set 2, consisted of 10 authors each, where each author wrote roughly the same number of blogs. Data Set 3 was a set of 100 authors with a slightly larger variation in the number of documents written per author. Data Set 4 consisted of 1000 authors with a wide variety in their prolificacy. Data Set 1 and 2 had a class imbalance of roughly 10 to one for each target author. Data Set 3

had a class imbalance of approximately 100 to one. Data Set 4 had an extreme class imbalance; on average, it was 1000 to one. Within each data set, when the less prolific authors were the target, there was an even larger class imbalance. The posts in Data Set 1 and 2 are disjoint from each other, but the posts of both are included in Data Set 3 and Data Set 4. The data sets, and their characteristics, are listed in Table 4. We ran the Naïve Bayes classifier on all four sets of data. We only ran the SVM on Data Set 3, the subset of 100 authors, due to time constraints.

	# of Authors	Posts per Author	Selected from Category
Data Set 1	10	107 to 120	"some signatures"
Data Set 2	10	403 to 507	"some signatures"
Data Set 3	100	107 to 773	100 most prolific authors from "some signatures"
Data Set 4	1,000	20 to 1,337	434 most prolific authors from "some signatures" 566 most prolific authors from "unsigned"

Table 4. Authors Chosen for Data Sets 1-4

B. FEATURE SELECTION

We used the following features:

- Word Features
  - Unigrams
  - Bigrams
  - Trigrams
- Character Features
  - Bigrams
  - Trigrams
  - 4-grams

Each of our experiments used one of the above features.

We converted all text to lowercase before tokenizing into word or character n-grams. When using n-grams of size 2 or larger, a new token, <post>, was added to indicate the start of a post. Similarly, the token </post> was added to indicate the end of a post.

1. Tokenizing Words

When processing the data into word grams we removed all punctuation. This was accomplished by replacing everything other than alphanumeric and whitespace characters with the empty string. The words were then tokenized on whitespace, discarding the whitespace.

2. Tokenizing Characters

When processing the data into character grams, all data was retained, to include whitespace, line feeds and carriage returns.

3. Test Data Selection

As we processed each post, we randomly (10% of the time) set aside the post for test data. This was done using the Java library class *java.util.Random* with a seed of 1. While this did not result in exactly 10% of every author's data being set aside for test data, it provided a close approximation. The remaining 90% of the posts were designated as training data. For Data Set 4, the ratio of posts set aside for test data was increased to 20% because the random selection resulted in some of the less prolific authors having none of their posts reserved for test data. We also used the 20% split for the SVM classifier.

C. NAÏVE BAYES

1. Bag of N-grams and Smoothing

We used the unigram version of Witten-Bell smoothing. To make this work with higher order n-grams, we used a bag-of-n-grams model, treating the n-grams as independent of one another. Thus, we capture more context than pure unigrams, but less context than the traditional n-gram language model. The probability formulas for a particular n-gram term are thus identical to the unigram formulas for a particular word.

$$P_{WB}(w_{i-n+1} \ldots w_i) = \frac{C(w_{i-n+1} \ldots w_i)}{N+T} \qquad \text{if } C(w_{i-n+1} \ldots w_i) > 0$$

$$P_{WB}(w_{i-n+1}w_i) = \frac{T}{N+T} \times \frac{1}{Z} \qquad \text{if } C(w_{i-n+1} \ldots w_i) = 0$$

$C(w_{i-n+1} \ldots w_i)$ = the count of the n-gram token: $(w_{i-n+1} \ldots w_i)$

T = the number of distinct n-grams (types).

N = the total number of n-gram tokens seen

Z = the estimate number of unseen words. We used the Google estimate for unigrams: 13,588,391 [37].

The motivation for using Witten-Bell smoothing (the unigram version) was that it is simple to implement and it obtains reasonably good results.

2. Modeling the Other Authors

We designated one author as the target author. The n-gram counts of all other authors were combined to approximate the characteristics of an "average" author. In each experiment, we iterated though all the authors in the data set, each one being designated as the target author in turn. Thus for each experiment we obtained as many F-scores as there were authors. In the results section we present the average of these scores as a means to evaluate the overall effectiveness of each feature type.

D. SUPPORT VECTOR MACHINE

1. SVM Toolset

We used the LIBLINEAR SVM library from [38]; the software tool set is available for download at [39]. The library uses a linear kernel and allows the user to provide a slack variable. We used powers of 2 for our slack variables, ranging from 2^{-17} to 2^{14}. The input to the SVM is a set of training vectors and a set of test vectors. Each vector represents a count of n-gram frequencies for a single blog post. We created a vector representation of every post. Each vector was assigned a class label: +1 for the target author, -1 for all other authors. The SVM is trained on the training posts vectors and produces a model file,

containing the weight vector $\vec{w}$ and constant b, defining a hyperplane. The model vector and test vectors are then used to produce a predicted class label for each of the test vectors.

2. Building the Vector Model

Prior to creating the vector models of individual posts, we calculated the number of occurrences of all n-grams in the training posts. We discard all n-gram terms that occurred only once in the training data. The remaining terms were used as a dictionary of significant terms. When we built the vectors for the training and test posts, any term not found in this dictionary was discarded.

3. Modeling the Other Authors

In each run of the experiment, we created a set of count vectors for all of the test and training posts, and labeled each vector as one of two classes. We designated one author as the target author and label all of their post vectors $+1$. All other authors were grouped into a single class, and their post vectors were labeled -1. This second class was our attempt to model an "average" author.

As in the Naïve Bayes experiments, we iterated through all authors in the data set, each one being designated as the target author in turn. The only change to the post vectors was the label; all the counts remained the same. Thus, we were able to accomplish this by making one copy of the vector files for each author, changing the labels to reflect the new target author. This produced as many F-scores as there were authors in the data set. We present the average of these scores in the results section.

E. EVALUATION CRITERIA AND BASELINE

1. Evaluation Criteria

The problem we are addressing is highly imbalanced, thus accuracy is an ineffective measure of effectiveness. Therefore, we chose to use precision, recall, and F-score as our evaluation criteria.

2. Baseline

We considered three possible baselines, detailed below.

The first baseline we considered labeled every test post as the most likely class. However, the most likely class is not the target, and thus results in an F-score of 0; over which almost any result would be an improvement.

The second baseline we considered labeled $n\%$ of the test posts of both classes as "written by the target" and the remaining posts as "not written by the target"; where $n\%$ is the percentage of the training documents written by the target. For example, if the target wrote 100 training posts, and these made up 1% of the training documents, the baseline would be:

- Precision = 0.01, Recall = 0.01, F-score = 0.0100.

The third baseline—the baseline we used—labeled all test posts as posts written by the target author. Thus, this baseline has perfect recall, but poor precision and F-score. If the target wrote 100 training posts and these made up 1% of the training documents, this baseline would be:

- Precision = 0.01, Recall = 1.00, F-score = 0.0198.

We chose to use the third baseline, as it is the most challenging to improve upon.

IV. RESULTS AND ANALYSIS

A. RESULTS

Both the Naïve Bayes and SVM models demonstrate the ability to identify documents written by a particular author. Character n-grams of size 3 or 4 proved to be the most discriminating feature. Word unigrams also performed fairly well. In addition to the results presented here, we ran Naïve Bayes experiments using word 4-grams and character n-grams of size 5 to 8. We found these higher order n-grams to be less discriminative in their ability to identify the works of a particular author. The performance of the classifier degraded as the size of the n-gram increased. In the Naïve Bayes classifier, character trigrams were the most discriminative feature, except in Data Set 1, where character bigrams attained the highest score. In the SVM classifier, character trigrams and 4-grams attained similar results and were superior to the other features.

We ran the Naïve Bayes classifier on Data Set 1 through 4. We ran the SVM on Data Set 3 (100 authors). On Data Set 3, the SVM performed significantly better than the Naïve Bayes classifier.

Each experiment produced a large number of scores. For example, Data Set 4 produced 6000 F-scores. After calculating the F-score for each target author, we calculated the mean F-scores for each experiment. In the SVM model, the scores resulting from the slack variable generating the highest F-score for each author, were used to calculate the mean precision, recall, and F-score for each experiment. The mean F-scores are presented in Tables 5–7 in Section 1. Figures 3–8 in Section 2 present the distribution of F-scores across authors, as well as F-scores as a function of percentage of training data written by the target author. The detailed results figures in Section 2 show the results for the best feature from each experiment. Because the results from character trigrams and 4-grams were so similar in the SVM, Figures 7–8 present the detailed results for both of these features.

1. Summary Results

a)

Naïve Bayes: Data Set 1 (10 Authors)	Average Precision	Average Recall	Average F-score
Average Baseline Scores	**0.1000**	**1.0000**	**0.1805**
Character Gram Size			
2	0.6170	0.5702	0.5880
3	0.7478	0.3926	0.4966
4	0.6333	0.2572	0.3540
Word Gram Size			
1	0.6639	0.3196	0.4125
2	0.5833	0.2043	0.2847
3	0.4167	0.1631	0.2203

107 to 120 posts per author. Used a 90/10 training/test split.

b)

Naïve Bayes: Data Set 2 (10 Authors)	Average Precision	Average Recall	Average F-score
Average Baseline Scores	**0.1000**	**1.0000**	**0.1813**
Character Gram Size			
2	0.5370	0.6789	0.5926
3	0.8031	0.6193	0.6749
4	0.8607	0.5280	0.6169
Word Gram Size			
1	0.8029	0.5059	0.6015
2	0.8068	0.2915	0.4051
3	0.7739	0.3061	0.4230

403 to 507 posts per author. Used a 90/10 training/test split.

c)

Naïve Bayes: Data Set 3 (100 Authors)	Average Precision	Average Recall	Average F-score
Average Baseline Scores	**0.01000**	**1.0000**	**0.0197**
Character Gram Size			
2	0.1912	0.5411	0.2684
3	0.3635	0.3998	0.3573
4	0.3748	0.2514	0.2792
Word Gram Size			
1	0.5860	0.2128	0.2849
2	0.2963	0.0920	0.1304
3	0.3576	0.0766	0.1169

107 to 773 posts per author. Used 90/10 training split.

d)

Naïve Bayes: Data Set 4 (1000 Authors)	Average Precision	Average Recall	Average F-score
Average Baseline Scores	**0.0010**	**1.0000**	**0.0020**
Character Gram Size			
2	0.0599	0.4026	0.0913
3	0.1502	0.2558	0.1655
4	0.1221	0.1494	0.1247
Word Gram Size			
1	0.2179	0.1305	0.1476
2	0.0486	0.0602	0.0482
3	0.0548	0.0228	0.0278

20 to 1337 posts per author. Used 80/20 training split.

Table 5. Naïve Bayes: Result Averages

The average F-scores for Data Set 3 are presented in Table 6 (Naïve Bayes) and Table 7 (SVM). The information in Table 6 is the same as Table 5.c, but is reprinted here for ease of comparison to the SVM results.

Naïve Bayes: Data Set 3 (100 Authors)	Average Precision	Average Recall	Average F-score
Average Baseline Scores	**0.01000**	**1.0000**	**0.0197**
Character Gram Size			
2	0.1912	0.5411	0.2684
3	0.3635	0.3998	0.3573
4	0.3748	0.2514	0.2792
Word Gram Size			
1	0.5860	0.2128	0.2849
2	0.2963	0.0920	0.1304
3	0.3576	0.0766	0.1169
107 to 773 posts per author. Used 90/10 training split.			

Table 6. Naïve Bayes: Data Set 3 Result Averages

SVM: Data Set 3 (100 Authors)	Average Precision	Average Recall	Average F-score
Average Baseline Scores	**0.0100**	**1.0000**	**0.0197**
Character Gram Size			
2	0.5985	0.4289	0.4815
3	0.7179	0.4648	0.5463
4	0.7526	0.4546	0.5453
Word Gram Size			
1	0.6896	0.4121	0.4994
2	0.6882	0.2452	0.3311
3	0.5367	0.1165	0.1723
107 to 773 posts per author. Used 80/20 training split.			

Table 7. SVM: Data Set 3 Result Averages

2. Detailed Results

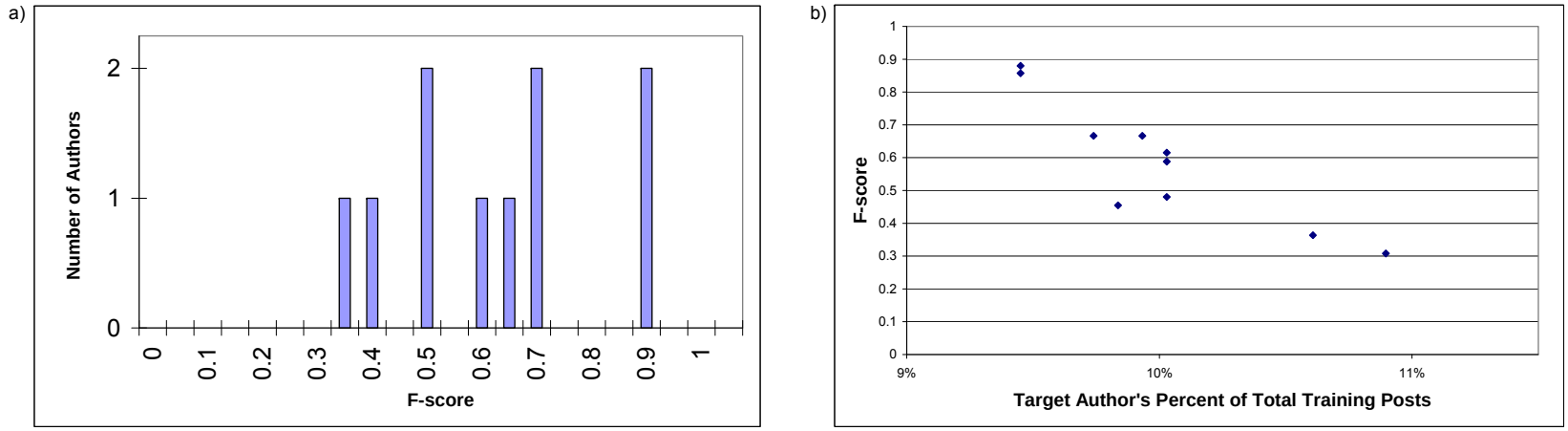

Figure 3. Naïve Bayes: Data Set 1 (10 Authors): F-scores on Character Bigrams.

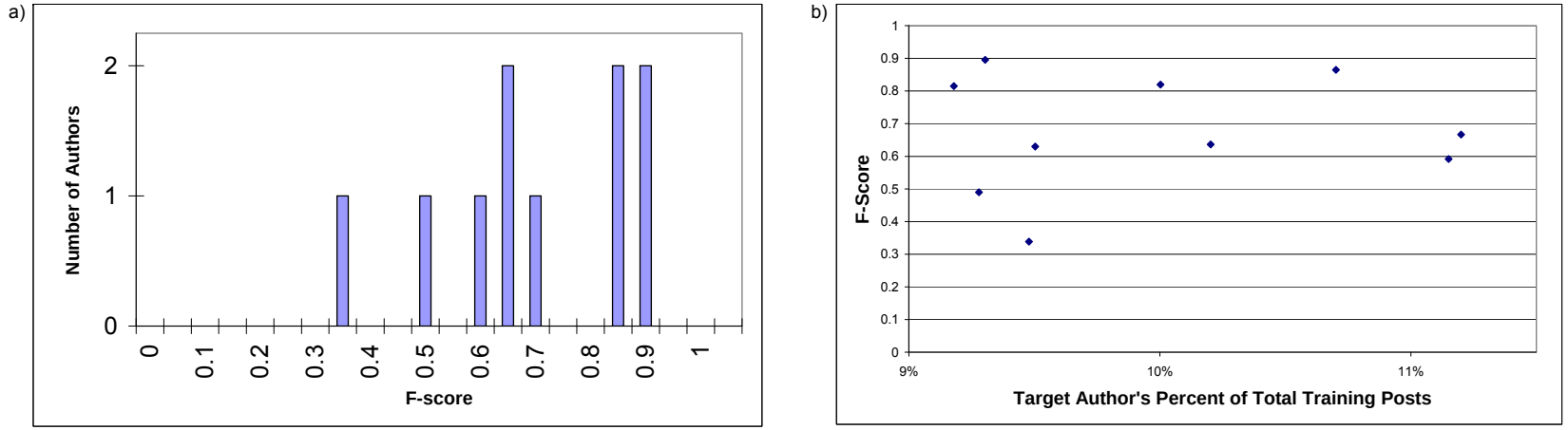

Figure 4. Naïve Bayes: Data Set 2 (10 Authors): F-scores on Character Trigrams.

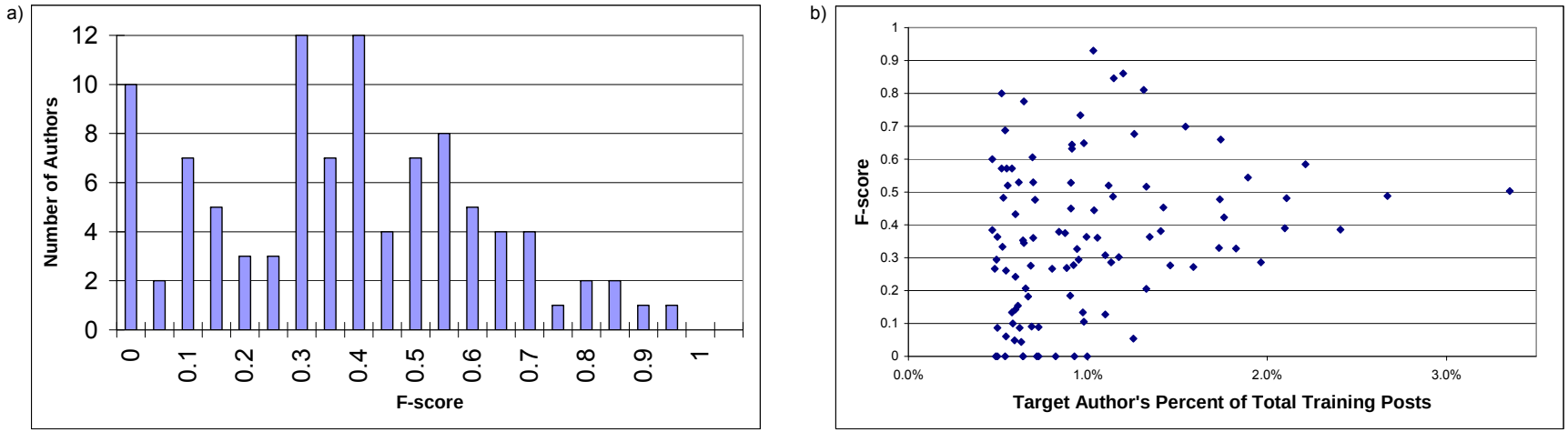

Figure 5. Naïve Bayes: Data Set 3 (100 Authors): F-scores on Character Trigrams.

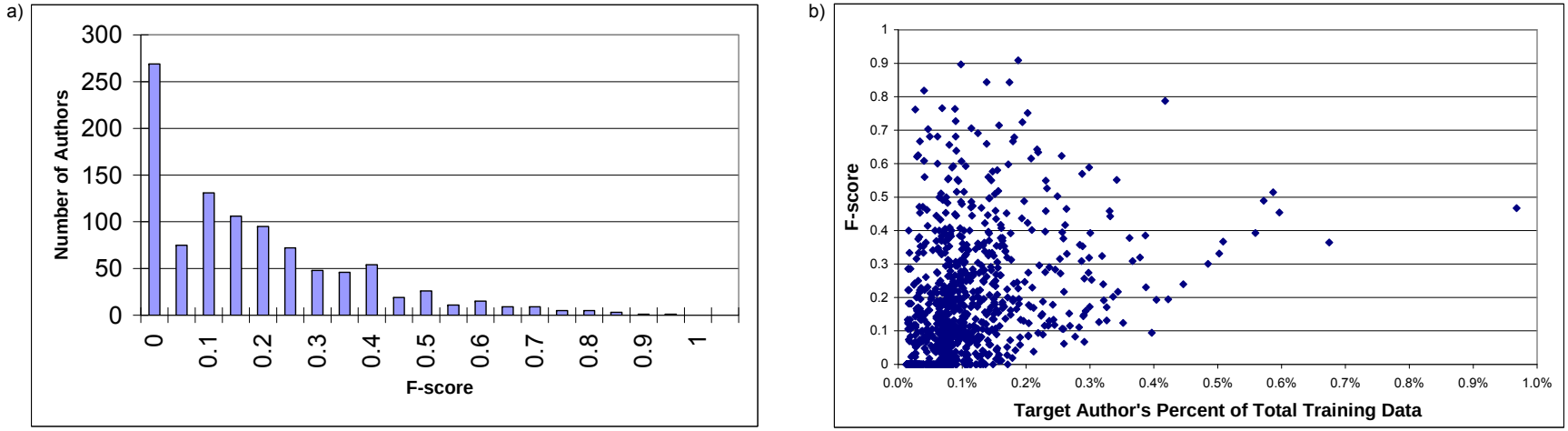

Figure 6. Naïve Bayes: Data Set 4 (1000 Authors): F-scores on Character Trigrams.

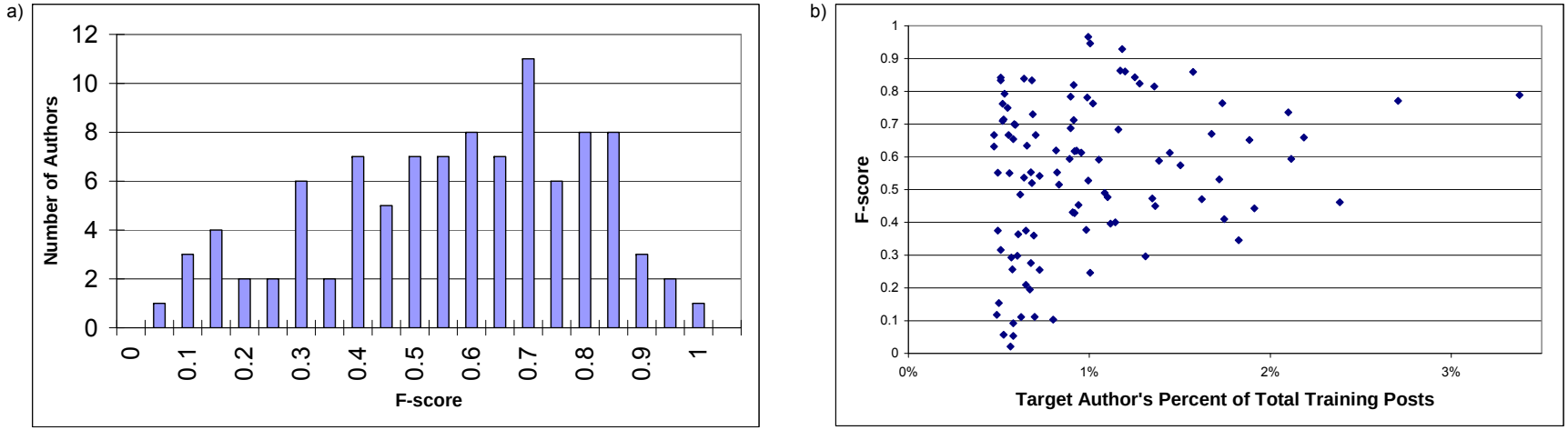

Figure 7. SVM: Data Set 3 (100 Authors): F-scores on Character Trigrams.

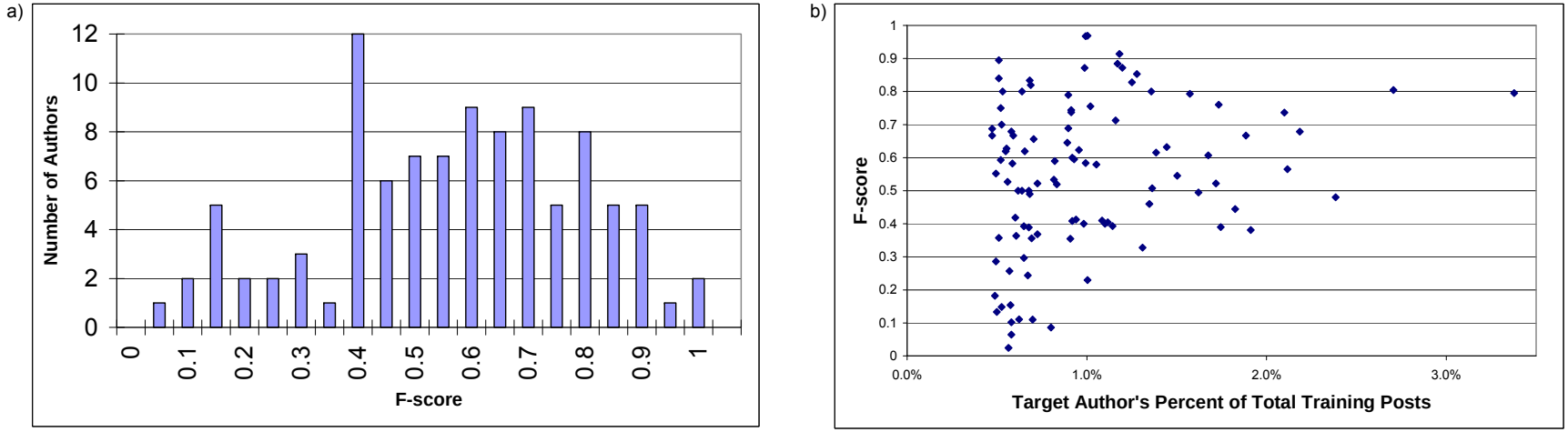

Figure 8. SVM: Data Set 3 (100 Authors): F-scores on Character 4-grams.

B. ANALYSIS

1. Effective Features

a. Word Unigrams

Word unigrams were more discriminative than higher order word n-grams. One possible reason is that higher order word n-grams could result in the feature vectors becoming too sparse. It is also possible that the higher order word n-grams did not do as well because they were capturing context specific to the topic instead of the style of the author. In the Naïve Bayes classifier, we may have assigned too much probability to each occurrence of the zero count events in the higher order n-grams. Higher order n-grams have more possible types, thus, they are likely to have an increased number of unseen types in the test data. We used a constant to estimate how much of the probability mass reserved for zero count events to give to each occurrence. Therefore, in higher order n-grams, we are likely to assign more probability mass to zero count events.

b. Character Trigrams

Character Trigrams were one of the most discriminative features across all of the data sets. Data Set 1 was an exception, where character bigrams performed better, but since this data set only had 10 authors, this is potentially anomalous. In the SVM classifier, character trigrams and character 4-grams both performed particularly well.

2. Effectiveness of the Classifiers

a. Naïve Bayes

On average, the Naïve Bayes classifiers did reasonably well. The average scores achieved on Data Set 3 (100 authors) using character trigrams were:

- Precision = 0.3635, Recall = 0.3998, F-score = 0.3573.

In application, this would allow an automated tool to reduce the workload of the human analyst, from manually finding one document for every 100 examined, to finding one document written by the target for every three examined, and recovering more than one third of the documents written by the target.

The average scores achieved on Data Set 4 (1000 authors) using character trigrams were:

- Precision = 0.1502, Recall = 0.2558, F-score = 0.1655.

Even in the high class imbalance of Data Set 4, our results would significantly reduce the workload of the human analyst. The scores achieved, on average, would reduce the workload, from only finding one document for every 1000 examined, to finding one document for every seven examined, and recovering 25% of the documents written by the target. However, the results of each experiment contained significant variance. Some authors were particularly distinctive, resulting in exceptionally high F-scores in all data sets. With other authors, in data sets with a large class imbalance, the classifier failed to identify a single post from the target author.

b. SVM

The SVM outperformed the Naïve Bayes classifier, on the set of 100 authors. We did not run the SVM on the other data sets. The average scores achieved, with character 4-grams, were:

- Precision = 0.7526, Recall = 0.4546, F-score = 0.5453.

In application, this would allow an automated tool to reduce the analytical workload from manually finding one document for every 100 examined, to finding seven or eight documents written by the target author for every 10 examined, and recovering almost half of all the documents written by the target author. Eight of the authors were particularly distinctive, with F-scores above 0.85. Three of the authors were particularly difficult to detect, with F-scores

below 0.10. Even when the F-scores are as low as 0.08, the classifier has some practical value. For example, the 3rd least distinctive author had scores:

- Precision = 0.0510, Recall = 0.2703, F-score of 0.0858.

This author wrote 149 of the 18,624 training posts, thus an analyst could be expected to find only one document by this author for every 125 examined. The classifier is able to reduce this workload down to one target document for every 20 documents examined, and recover more than 25% of the documents written by this author. 97% of the authors had precision and F-score greater than this author. 91% of the authors had a recall greater than 0.24.

Additional experiments would have to be performed to determine to what extent the SVM's discriminative ability is degraded as the classes become more imbalanced.

3. Effect of Class Imbalance

As expected, the performance of the Naïve Bayes classifier declined as the classes became increasingly imbalanced. In Data Sets 1 and 2, each author wrote roughly an equal number of posts, approximately 10% of the training data. Data Set 3 was slightly imbalanced; when the least prolific author was the designated target, the class representing the "other authors" wrote 214 times more posts than the target author. Data Set 4 was very imbalanced; when the least prolific author was the target, the class imbalance was 7,794 to one. Table 8 gives the proportion of training data per author for each of the data sets.

	Min Training Posts/Author	Max Training Posts/Author	Total Training Posts	# Authors	% of Training Data per Author (min% - max%)
Data Set 1	98	113	1,037	10	9.45% - 10.90%
Data Set 2	368	449	4,009	10	9.18% - 11.20%
Data Set 3	98	703	20,969	100	0.47% - 3.35%
Data Set 4	14	1,056	109,110	1,000	0.01% - 0.97%

Table 8. Proportion of Training Data per Author

As seen in Figure 6 in Section IV.A.2, Data Set 4 had a large number of authors with zero F-scores. Table 9 shows the distribution of zero F-scores. As might be expected, the zero F-scores were overwhelmingly concentrated among the least prolific authors. What is not expected is the significant number of non-prolific authors with high F-scores. As the number of potential authors increases, the number of posts representing the works not written by the target author grows and the classes become significantly more imbalanced. We expect to see the ability to detect the works of any particular author decline, with the most rapid degradation among the least prolific of the authors. In general, this is what we observed. The accuracy of the classifier did decline, and the classifier failed to identify any posts from a significant number of the less prolific authors. However, some of the best F-scores remained among the least prolific of the authors. For 79% of the authors who wrote 30 or fewer training posts, the classifier was unable to detect a single post by the target author, but author *r1237310.male.38.Arts.Taurus.xml*, with the ninth highest F-score (0.7619), only wrote 29 posts in the training data. For that author, the class imbalance was 3,762 to one.

Distribution of authors with F-score = 0

Training Posts	# Authors	# F-scores = 0	% of 0 F-scores
≤ 20	82	66	25%
≤ 30	174	137	51%
≤ 70	353	203	75%
≤ 100	631	255	95%

269 authors had an F-score = 0
Data Set 4: 1000 total authors in this data set

Table 9. Naïve Bayes: Data Set 4, Distribution of Zero F-scores

Some of the authors were hardly affected by an increase in class imbalance. The author in blog file *r1970293.female.24.Technology.Aries.xml* wrote 0.5198% of the training data in Data Set 3 (a class imbalance of 192 to one) and achieved an F-score of 0.80. This author wrote only 0.0889% of the

data in Data Set 4 (a class imbalance of 1,125 to one), but the F-score dropped only slightly, to 0.7636; still one of the best F-scores of that experiment. Only seven of the 1000 authors had better results.

4. Distinctive Authors

Some authors were particularly distinctive. These authors had high F-scores in both the Naïve Bayes classifier and the SVM. Even in Data Set 4, where there was an extreme class imbalance, the F-scores of these authors only declined slightly. Table 10 shows the results of one such author. All of the classifiers were able to identify the works of this author with high F-scores, the best of which was the SVM, which had zero false positives and correctly identified more than 93% of this author's posts. Even in Data Set 4, where the class imbalance on this author was 574 to one, the effectiveness of the Naïve Bayes classifier was not significantly degraded.

	Baseline Precision	Baseline Recall	Baseline F-score	Precision	Recall	F-score
NB: Data Set 3	0.0085	1.0000	0.0168	0.8696	1.0000	0.9302
SVM: Data Set 3	0.0104	1.0000	0.0206	1.0000	0.9388	0.9684
NB: Data Set 4	0.0017	1.0000	0.0033	0.7679	0.9348	0.8431

Table 10. Example of a Distinctive Author: F-scores when Identifying the Posts written by *r2117806.male.24.Student.Aries.xml* (NB: character 3-grams, SVM: character 4-grams)

In an effort to determine why some authors, in particular those authors with little training data, were easily distinguishable, we examined the posts of 20 of the distinctive authors in Data Set 4. We examined posts of authors in the following categories:

- Less than 50 training posts (examined the 10 best F-scores)
- 50 to 100 training posts (examined the 5 best F-scores)
- More than 100 training posts (examined the 5 best F-scores)

We examined more of the authors with "fewer than 50 posts," because these are the authors with the most surprising results; these authors attained

high F-scores, in an extremely imbalanced class, with very little training data. The complete list of the authors we examined, their F-scores, and the distinctive traits we discovered is provided in Appendix B.

We looked for authors that used a limited vocabulary or that wrote on a single topic. We also looked for authors writing unusually long or short posts. The post length was not a feature used by our classifier, but could still affect our results, as this would increase, or decrease, the number of n-grams in the training data for the target author. As shown in Table 11, we discovered possible explanations for eight of the 20 authors we inspected. Table 12 presents a more detailed version of these results by category.

20 Distinctive Authors
12 - varied topic, varied post length, no discernable pattern.
2 - distinctive and consistent misspelling of numerous words
6 - single topic

Table 11. Distinctive Author Characteristics

Authors with < 50 Training Posts (10 authors)

# of Authors	Characteristics
4	varied topic, varied post length, no discernable pattern
2	distinctive and consistent misspelling of numerous words
2	single topic, varied post length, varied vocabulary
1	single topic, varied post length, limited vocabulary
1	38% of posts were varied topic, varied post length, varied vocabulary 62% of posts were single topic, short posts (~75 words), limited vocabulary

Authors with 50 to 100 Training Posts (5 authors)

# of Authors	Characteristics
5	varied topic, varied post length, no discernable pattern

Authors with > 100 Training Posts (5 authors)

# of Authors	Characteristics
3	varied topic, varied post length, no discernable pattern
2	single topic, short posts, limited vocabulary ~5 words/post in one blog ~75 words/post in the other blog

Table 12. Distinctive Author Characteristics by Category

a. Distinctive Misspelling

Two of the authors we examined, consistently misspelled numerous words in distinctive ways. For example, author *r3428854.male.17.indUnk.Cancer.xml*, omits the letter the 'h' from the word "that," spelling it "tat," in 91 out of 101 uses. This author also includes Chinese characters in some of his posts. We believe that misspellings may be a good indication of a particular author, as a particular individual will often misspell certain words in a consistent manner. When using character n-grams, such characteristic misspellings are captured automatically. They change the distribution of n-gram frequencies. In the above example, the character trigrams <*[space], t, h*>; <*t, h, a*>; and <*h, a, t*> are less frequent, and the trigrams <*[space], t, a*>; <*t, a, t*> are more frequent, than if the author had not misspelled the word "that."

b. Foreign Language Characters

Two of the authors we examined used foreign language characters in some of their posts. Author, *r3428854.male.17.indUnk.Cancer.xml* included short phrases of Chinese characters in five of his 51 posts (training and test). The Chinese phrases are embedded in the middle of posts written in English. Three of the 23 posts (training and test) written by author *r3521040.male.33.Manufacturing.Cancer.xml* contained sentences in Arabic script. Two of these were entirely in Arabic, and the other was written half in Arabic and half in English. The other 20 posts this author wrote were written in English. The presence of these foreign language characters may be characteristic to some authors, however in the two we found, the foreign language characters only occurred in a small number of the authors' posts (less than 15%). The specific Arabic and Chinese characters used by these two authors are likely to be too sparse for the classifiers to take advantage of this unique characteristic. It is unclear if this increased or decreased the accuracy of the classifiers. The presence or absence of foreign language characters could

potentially be used as an additional feature in future research. Additional study would have to be done to determine the effect of intermixed foreign language characters on author verification.

c. Single Topic

Five of the authors we examined only write about a single of topic. In one such case, *r1237310.male.38.Arts.Taurus.xml*, every post consisted of one or more movie reviews. Because of the specific topic of this author, his vocabulary was limited; many of the posts contained the same words and phrases. Three of this author's posts were unusually long (1500–4900 words), but most were typical length (less than 200 words). This author's limited vocabulary make his posts easy to distinguish from the posts of other authors, but it would also make it more difficult to detect this author if he were to write on another topic. Four of the authors writing on a single topic used a noticeably limited vocabulary. The other two used a much wider vocabulary. There was also one author, r3093523.female.25.Marketing.Sagittarius.xml, who wrote on multiple topics in 38% of her posts, but the other 62% were on a single topic, cigar reviews. Her single topic posts were short and contained limited vocabulary. Many of these posts shared the same phrases. Additional study is needed to explore the effect of topic on the effectiveness of author detection.

d. No Discernable Pattern

The remaining 12 authors wrote about multiple topics and used varied vocabulary from one post to the next. Further study would be needed to identify what set these authors apart from the authors with low F-scores.

e. Multiple Authors

One of the blogs we examined, *r4283298.male.27.Arts.Taurus.xml*, contained multiple authors. Every post started with an author signature and the authors seemed to be responding to each other's posts. Surprisingly, this blog attained the second highest F-score in Data Set 4 (F-score of 0.897). One

author wrote 58% of the posts. Most of posts by this author contained more than 200 words. The remaining 42% of the posts was divided among several authors. Most of these posts contained fewer than 50 words. Additional research would be needed to determine what made this blog distinctive: the writing style of the dominant author, the combined writing style of a number of the authors, or some other factor. Recommended future work includes determining how much of the vocabulary in the blog was shared by all authors and how much of the vocabulary in the blog was not used by the dominant author.

5. Effect of Quantity of Training Data

More than 25% of the authors in Data Set 4 had an F-score of zero. 95% of the authors with an F-score of zero had fewer than 100 posts in the training data. We believe part of the explanation for the large number of authors with an F-score of zero in Data Set 4 is that many of these authors had insufficient training data for the classifier to be able to distinguish their style from that of an "average author." We suggest that the reason some of the non-prolific authors did well, despite having such little training data, is that some authors are particularly distinctive and their documents can be identified given only a small sample of their writing. Other, less distinctive, authors require a much larger sample to distinguish their work from the model of the "average author."

As a class becomes more imbalanced, more training data is required to effectively discriminate the works of a particular author. However, particularly distinctive authors mitigate the negative effects of the class imbalance.

Authors with more training data tend to have higher F-scores, and they are more resistant to the effects of class imbalance. The authors of Data Set 1 had approximately 100 training posts each, while those of Data Set 2 had roughly 400 training posts each. The posts in Data Set 1 and 2 (10 authors each) are included in Data Set 3 (100 authors) and Data Set 4 (1000 authors). As the number of authors increases, so does the class imbalance for the target author. Table 13 shows the effects of increasing class imbalance on the authors of Data

Set 1 and 2. The complete list of F-scores for these authors across all data sets is included in Appendix C. When the classes were highly imbalanced (in Data Set 4), three of the 10 authors from Data Set 1 had an F-score of zero; half of them had an F-score less than 0.1. Even with the high class imbalance of Data Set 4, the classifier has significantly more success distinguishing the authors of Data Set 2. None of the authors had an F-score of zero, and nine out of 10 had F-scores greater than 0.10. The data set with more training data, Data Set 4, did not suffer as much degradation as the class imbalance increased.

Class Imbalance	F-score Threshold	# Authors from Data Set 1 over F-score Threshold (~100 posts/author)	# Authors from Data Set 2 over F-score Threshold (~400 posts/author)
10 to 1	> 0.30	8	10
	> 0.20	9	10
	> 0.10	9	10
	> 0.00	9	10
100 to 1 (Data Set 3)	> 0.30	4	9
	> 0.20	6	10
	> 0.10	6	10
	> 0.00	7	10
1,000 to 1 (Data Set 4)	> 0.30	1	3
	> 0.20	4	4
	> 0.10	5	9
	> 0.00	7	10
Data Set 1 contains exactly 10 authors. These authors are also in Data Set 3 and 4. Data Set 2 contains exactly 10 authors. These authors are also in Data Set 3 and 4. Data Set 3 contains 100 authors. Data Set 4 contains 1000 authors.			

Table 13. Effect of Class Imbalance on Authors of Data Set 1 and 2.

Thus, poor performance of the classifier on many of the authors in Data Set 4 is possibly a combination of insufficient training data and class imbalance, both of which have limited effect on distinctive authors. In general, the larger the class imbalance, the more training data is needed to overcome the negative effects of the class imbalance.

In [2], [10], Gehrke et al., demonstrated that corrective scaling could be used to mitigate the class imbalance problem; however, his technique cannot be applied to the authorship verification problem, because we do not have distinct models for each possible author, as he did.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION AND RECOMMENDATIONS

A. SUMMARY

Our research addresses the problem of author verification in short documents; given examples of the writing of a single author, determining whether a text of unknown authorship was written by the same author. We tested the effectiveness of combining the works of other authors, to model the characteristics of an "average author." We tested this approach using a Naïve Bayes Classifier and a Support Vector Machine. In the Naïve Bayes Classifier, we tested the effects of various levels of class imbalance. We experimented with word and character n-grams of various sizes. Among the word n-grams, unigrams had the best results. Overall, character trigrams were the most discriminating feature. In the SVM, character trigrams and character 4-grams had similar results, but character 4-grams had slightly higher precision.

The SVM outperformed the Naïve Bayes classifier on a set of 100 authors, achieving an average F-score of 0.54 with a precision of 0.75. Even an F-score as low as 0.08, has the potential to be useful. We achieved an F-score of 0.08 or greater on 98% of the authors in Data Set 3.

We found that there is a minimum amount of training data required for the classifier to be effective. The classifiers were effective on most authors with at least 100 training posts. Increasing the number of posts used to model the "average author" relative to the number of training posts for the target author increases the class imbalance and degrades the effectiveness of the classifier. As the class imbalance increases, more training data is required to maintain the same effectiveness.

Some authors are particularly distinctive. These authors are not affected as much by class imbalance and require a much smaller set of training examples. In the small set of blogs we examined, we discovered that several

wrote about a single of topic, or had noticeable spelling idiosyncrasies. However, for more than half of the blogs we examined, it was not apparent what made their posts easier to identify than the posts other authors.

B. FUTURE WORK

1. The Class Imbalance Problem

Some of the authors were included across multiple data sets. Authors with high F-scores in one data set had high F-scores in the other data sets, that is, they were distinguishable from the model of the "average author." The model of the "average author" changes significantly from one data set to another. It is likely that we can model the "average" author by taking a small random sample of the other authors. This would alleviate the negative effects of class imbalance, and may allow this technique to scale to an unbounded number of possible authors. Such a technique has been shown to be effective in cases of moderate class imbalance: Koppel et al. in [8] adjusted for imbalanced classes by randomly discarding samples from the more prolific class until they had the same number of documents (500 word chunks) in both classes. The class imbalance in our experiments is significantly larger than in the work of Koppel et al., but since we are creating a rough approximation of an average author instead of modeling a specific author, such a technique may work here. It would be worth further investigation.

In Naïve Bayes classifiers, a possible approach to adjust for the class imbalance problem would be to divide the counts of all terms in the "other authors" class by the number of authors.

2. The Effect of Topic on Author Verification

In [8], Koppel et al. demonstrate that some authorship verification techniques are not affected by the influence of topic. We did not control for topic. Some of our authors discuss multiple topics, while others only write about a single topic. We examined only a small number of the blogs for topic; however,

the authors over which our classifier achieved a high F-score included authors discussing multiple topics as well as single topic authors. Further study is required to determine the effect of topic on the author verification task. In particular, it would be useful to know the effectiveness of the classifier, when the topics in the training data are distinct from those in the test data for the target author.

If the influence of topic does degrade the classifier, discarding the most discriminative terms, as in Koppel and Schler's unmasking [22], may allow a classifier to identify additional posts written by the same author, but written on a different topic without generating too many false positives. Koppel and Schler found that discarding a small number of terms prevents a classifier from being able to separate two documents from the same author on different topics, but significantly more terms must be discarded before the classifier cannot effectively separate the works of different authors.

Using only function words or the k most frequent words (which tend to be mostly function words) is a common technique used to limit the effects of topic in authorship attribution. Another technique that may mitigate the negative effects of topic would be to limit the features to the k most frequent character n-grams. This has been done with the k most frequent words; doing the same with character n-grams may help.

3. Applicability of Character N-grams to Foreign Languages

In [8], Koppel et al. demonstrate that an SVM using the word frequencies of the 250 most frequent words can be applied with equal effectiveness to English and Hebrew-Aramaic text. We found character trigrams and 4-grams to be the most discriminating feature in our experiments. Additional experiments would need to test if these features are effective in other languages.

4. Refinements to the Classifiers

Future Naïve Bayes experiments should use a more advanced smoothing or back-off techniques, such as Katz Backoff or Kneser-Ney Smoothing [24]. In our SVM experiments, we only used one data set. Future SVM experiments should test the effects of increased class imbalance on the SVM classifier.

5. Additional Noise in the Data

In most real world applications, the posts processed by the system will include posts by authors the system has not seen before. Future experiments should include posts in the test data from authors that were not included in the training data. Including a few of the target author's posts in the training data for the "other authors" would also make the experiment more realistic. While we might have a clean sample of the target author's works, the training sample for "other authors" most likely contains works of unknown authors, possibly including some from the target author.

6. Application of Koppel's Unmasking to blogs

Koppel and Schler' unmasking could be applied to some blogs. If the blogs contained meta-data tags indicating all posts are from the same author, the entire blog could be treated as one long document. Their technique could then be used to test if two such blogs were written by the same author using different screen-names. It would have to be tested to see if their techniques work well in this domain. The minimum number of posts or words needed for their methods to be effective would also have to be tested.

APPENDIX A: SVM: CALCULATING THE HYPERPLANE

The parameters $\vec{w}$ and b determine the width of the margin in an SVM and define the separating hyperplane. We will choose $\vec{w}$ and b to maximize the width of the margin [28]. The width of the margin is the same as the distance between the supporting hyperplanes. The formulas for the supporting hyperplanes are $y_j(\vec{w}^T\vec{x}_j + b) = 1$ and $y_k(\vec{w}^T\vec{x}_k + b) = 1$ for some points j, k, where $j \in \{\text{positive training examples}\}$, $k \in \{\text{negative training examples}\}$ and $j, k \in \{\text{support vectors}\}$ [28]. Recall that $y_i \in \{-1, 1\}$ is the label for the classes, with $+1$ indicating a positive example and -1 indicating a negative example. Therefore, $y_j = +1$ and $y_k = -1$. The distance between the supporting hyperplanes is the difference between the distances from the origin to the closest point on each of the supporting hyperplanes as shown in Figure 2 (distance between the support hyperplanes = |distance a – distance b|). The distance from the origin to the closest point on a hyperplane is found by minimizing $\vec{x}^T\vec{x}$ subject to $\vec{x}$ being on the hyperplane [28],

$$\min_{\|\vec{x}\|} \vec{x}^T\vec{x} + \lambda(\vec{w}^T\vec{x} + b - 1)$$

$$\frac{d}{d\vec{x}} = 0 = 2\vec{x} + \lambda\vec{w} = 0 \qquad \text{therefore,}$$

$$\vec{x} = -\frac{\lambda}{2}\vec{w} \qquad \text{substituting } \vec{x} \text{ into } w^T\vec{x} + b - 1 = 0\text{:}$$

$$-\frac{\lambda}{2}\vec{w}^T\vec{w} + b = 1 \qquad \text{therefore,}$$

$$\lambda = \frac{2(b-1)}{\vec{w}^T\vec{w}} \qquad \text{substituting } \lambda \text{ into } \vec{x} = -\frac{\lambda}{2}\vec{w}\text{:}$$

$$\vec{x} = \frac{1-b}{\vec{w}^T\vec{w}}\vec{w}$$

$$\vec{x}^T\vec{x} = \frac{(1-b)^2}{(\vec{w}^T\vec{w})^2}\vec{w}^T\vec{w} = \frac{(1-b)^2}{\vec{w}^T\vec{w}}$$

$$\|\vec{x}\| = \sqrt{\vec{x}^T\vec{x}} = \frac{|1-b|}{\sqrt{\vec{w}^T\vec{w}}} = \frac{|1-b|}{\|\vec{w}\|}$$ similarly, working out for $\vec{w}^T\vec{x} + b + 1 = 0$:

$$\|\vec{x}\| = \frac{|-1-b|}{\|\vec{w}\|}$$

Subtracting these two distances gives the margin size,

$$\frac{|1-b|}{\|\vec{w}\|} - \frac{|-1-b|}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|}.$$

To maximize the size of the margin, we must minimize $\|\vec{w}\|$, subject to the constraint $y_k(\vec{w}^T\vec{x}_k + b) \geq 1 \ \forall k$. This will give us the largest possible separation between classes [28].

To do so, we use the Karush Kuhn Tucker (KKT) setup using positive Lagrange multipliers and subtracting the constraints. We first write this equation as an unconstrained problem using Lagrange multipliers λ_k [30]:

$$\begin{aligned} L_p &= \frac{1}{2}\vec{w}^T\vec{w} - \sum_k \lambda_k[y_k(\vec{w}^T\vec{x}_k + b) - 1] \\ &= \frac{1}{2}\vec{w}^T\vec{w} - \sum_k \lambda_k y_k(\vec{w}^T\vec{x}_k + b) + \sum_k \lambda_k \\ &= \frac{1}{2}\vec{w}^T\vec{w} - \sum_k \lambda_k y_k \vec{w}^T\vec{x}_k - \sum_k \lambda_k y_k b + \sum_k \lambda_k \\ &= \frac{1}{2}\vec{w}^T\vec{w} - \vec{w}^T\sum_k \lambda_k y_k \vec{x}_k - b\sum_k \lambda_k y_k + \sum_k \lambda_k \end{aligned}$$

Using the KKT conditions, we can equivalently solve the dual problem, which is to maximize L_p with respect to λ_k, subject to the constraints that the gradient of L_p with respect to $\vec{w}$ and b are 0 and that $\lambda_k \geq 0$ [30]:

$$\frac{\delta L_p}{\delta \vec{w}} = 0 \Rightarrow \vec{w} = \sum_k \lambda_k y_k \vec{x}_k$$

$$\frac{\delta L_p}{\delta b} = 0 \Rightarrow \sum_k \lambda_k y_k$$

Substituting the above derivatives into L_p, we get

$$L_p = \frac{1}{2}\vec{w}^T\vec{w} - \vec{w}^T\sum_k \lambda_k y_k \vec{x}_k - b\sum_k \lambda_k y_k + \sum_k \lambda_k$$

$$L_d = \frac{1}{2}(\vec{w}^T\vec{w}) - \vec{w}^T(\sum_k \lambda_k y_k \vec{x}_k) - b(\sum_k \lambda_k y_k) + \sum_k \lambda_k$$

$$= \frac{1}{2}(\vec{w}^T\vec{w}) - \vec{w}^T(w) - b(0) + \sum_k \lambda_k$$

$$= -\frac{1}{2}(\vec{w}^T\vec{w}) + \sum_k \lambda_k$$

$$= -\frac{1}{2}\sum_k\sum_l \lambda_k\lambda_l y_k y_l (\vec{x}_k)^T \vec{x}_l + \sum_k \lambda_k$$

Which is maximized with respect to λ_k only, subject to the constraints $\sum_k \lambda_k y_k = 0$ and $\lambda_k \geq 0,\ \forall k$; which can be solved using quadratic optimization [30].

Only a small percentage of the λ_k's are greater than zero. The set of $\vec{x}_k$ with $\lambda_k > 0$ are the support vectors, which lie on the margin. All the support vectors satisfy the equation $y_k(\vec{w}^T\vec{x}_k + b) = 1$. The vector $\vec{w}$ is a weighted sum of these support vectors. Thus b can be calculated from any of these support vectors, however for numerical stability, it is calculated from all the support vectors and an average is taken [30]. The separating hyperplane is thus defined by $\vec{w}$ and b.

In many cases, the two classes are not linearly separable; it is not possible to separate them with a hyperplane. In such cases we desire to find a hyperplane that separates the classes as well as possible with the fewest errors. This is done by defining "slack variables," s_k, to represent the allowable deviation from the margin, thus relaxing

$y_k(\vec{w}^T\vec{x}_k + b) \geq 1$ to $y_k(\vec{w}^T\vec{x}_k + b) \geq 1 - s_k$.

Thus, allowing points to be s_k distance on the wrong side of the hyperplane. To prevent large slack variables from allowing any line to partition the data, we add another term to the Lagrangian to penalize large slacks [28], [30]. The Lagrangian equation without slack variables is:

$$L_p = \frac{1}{2}\vec{w}^T\vec{w} - \sum_k \lambda_k[y_k(\vec{w}^T\vec{x}_k + b) - 1]$$

Adding the slack variable and an additional term to penalize large slacks, the formula becomes:

$$L_p = \frac{1}{2}\vec{w}^T\vec{w} - \sum_k \lambda_k[y_k(\vec{w}^T\vec{x}_k + b) + s_k - 1] + \alpha\sum_k s_k \ .$$

APPENDIX B: EXAMINATION OF 20 DISTINCTIVE AUTHORS

Tables 14–16 present the F-scores and distinctive traits, if any, of the distinctive authors we examined. These authors are from the Naïve Bayes experiment on Data Set 4 (1000 Authors). Those without any boxes marked, wrote on varied topics, with varied post lengths and no discernable pattern.

10 Best F-scores of Authors with < 50 posts **Naïve Bayes: Data Set 4 (1000 authors)** **Target Author**	**Training Posts**	**Baseline F-score**	**Precision**	**Recall**	**F-score**	**Rank by F-score (out of 1000)**	**Single Topic**	**Limited Vocab**	**** Short Posts**	**Unique Spelling**	**Comments**
r3521040.male.33.Manufacturing.Cancer.xml	18	0.0004	0.4000	0.4000	0.4000	110					3 of 23 posts contain Arabic
r1237310.male.38.Arts.Taurus.xml	29	0.0007	0.7273	0.8000	0.7619	9	X	X			movie reviews
r3679805.female.15.Student.Virgo.xml	32	0.0010	0.5625	0.6923	0.6207	30					
r2541686.male.15.indUnk.Aries.xml	34	0.0005	0.5556	0.7143	0.6250	28					
r3873347.male.27.indUnk.Capricorn.xml	36	0.0010	0.3810	0.6154	0.4706	73	X				running/races/training
r2916521.female.16.Arts.Taurus.xml	37	0.0008	0.5263	0.9091	0.6667	22				X	
r3128727.male.23.indUnk.Scorpio.xml	42	0.0010	0.3810	0.6154	0.4706	72					most posts > 1000 words
r3428854.male.17.indUnk.Cancer.xml	44	0.0005	0.4375	1.0000	0.6087	32				X	5 of 51 posts contain Chinese
r2140894.female.26.indUnk.Pisces.xml	44	0.0008	0.8182	0.8182	0.8182	5	X				bicycle races
r3093523.female.25.Marketing.Sagittarius.xml	45	0.0007	0.4375	0.7778	0.5600	44	X*	X*	X*		cigar reviews
Total Training Posts:	**109110**										

* 38% of posts were varied topic, varied post length, and varied vocabulary.
62% of posts were single topic, short post length (~75 words), limited vocabulary.

**In blogs marked "Short Posts," most posts were ~75 words or less

Table 14. Distinctive Authors with Less than 50 Posts

5 Best F-scores of Authors with 50 to 100 Posts Naïve Bayes: Data Set 4 (1000 authors) Target Author	Training Posts	Baseline F-score	Precision	Recall	F-score	Rank by F-score (out of 1000)	Single Topic	Limited Vocab	Short Posts	Unique Spelling	Comments:
r2304236.female.15.indUnk.Scorpio.xml	51	0.0010	0.5417	1.0000	0.7027	15					*
r2539230.female.16.Student.Cancer.xml	54	0.0015	0.6154	0.7619	0.6809	19					*
r1956622.male.34.Technology.Libra.xml	75	0.0015	0.6923	0.8571	0.7660	7					*
r1970293.female.24.Technology.Aries.xml	97	0.0019	0.7241	0.8077	0.7636	8					*
r3794174.female.24.Museums-Libraries.Leo.xml	98	0.0018	0.6452	0.8333	0.7273	11					*
Total Training Posts:	**109110**										

*** All blogs examined in this group contained varied topics, varied post length, and no discernable pattern.**

Table 15. Distinctive Authors with 50 to 100 Posts

5 Best F-scores of Authors with > 100 Posts Naïve Bayes: Data Set 4 (1000 authors) Target Author	Training Posts	Baseline F-score	Precision	Recall	F-score	Rank by F-score (out of 1000)	Single Topic	Limited Vocab	** Short Posts	Unique Spelling	Comments:
r4283298.male.27.Arts.Taurus.xml	107	0.0022	0.9286	0.8667	0.8966	2					Multiple Authors*
r1103016.male.16.Student.Gemini.xml	151	0.0026	0.9310	0.7714	0.8438	3	X	X	X		~5 words/post
r2117806.male.24.Student.Aries.xml	190	0.0034	0.7679	0.9348	0.8431	4					
r2155904.male.17.Student.Virgo.xml	205	0.0041	0.8462	0.9821	0.9091	1					
r1679249.female.37.indUnk.Leo.xml	456	0.0083	0.6607	0.9737	0.7872	6	X	X	X		photography reviews
Total Training Posts:	**109110**										

* Every post started with an author signature
58% of posts were written by one author, most of which had > 200 words/post
42% of posts were written by one of several authors, most of these posts had < 50 words/post

**In blogs marked "Short Posts," most posts were ~75 words or less

Table 16. Distinctive Authors with More than 100 Posts

APPENDIX C: AUTHORS IN MULTIPLE DATA SETS

The posts in Data Set 1 and 2 (10 authors each) are included in Data Set 3 (100 authors) and Data Set 4 (1000 authors). The following tables present the scores achieved when identifying these authors in each of the data sets. This shows the effect of increasing class imbalance on two sets of authors included in multiple data sets.

Table 17 lists the scores achieved on the authors of Data Set 1. Table 18 and 19 show the scores on these authors when they were part of a larger data set.

Data Set 1 (10 authors)							
Target Author	**Training Posts**	**Baseline Precision**	**Baseline Recall**	**Baseline F-score***	**Precision**	**Recall**	**F-score**
r463180.male.24.indUnk.Taurus.xml	101	0.0625	1.0000	0.1176	0.5000	0.1667	0.2500
r3388015.male.23.Student.Leo.xml	98	0.1042	1.0000	0.1887	0.8750	0.7000	0.7778
r3348302.female.25.indUnk.Virgo.xml	98	0.1250	1.0000	0.2222	0.8889	0.6667	0.7619
r2323827.female.25.indUnk.Pisces.xml	104	0.0729	1.0000	0.1359	1.0000	0.5714	0.7273
r2862338.male.16.Student.Libra.xml	104	0.0938	1.0000	0.1714	0.7143	0.5556	0.6250
r1008329.female.16.Student.Pisces.xml	104	0.1250	1.0000	0.2222	0.6667	0.3333	0.4444
r2016512.female.17.Student.Taurus.xml	110	0.0625	1.0000	0.1176	1.0000	0.3333	0.5000
r2303699.female.23.Science.Aquarius.xml	102	0.1458	1.0000	0.2545	1.0000	0.2143	0.3529
r3051042.female.17.Student.Capricorn.xml	103	0.1354	1.0000	0.2385	0.8333	0.3846	0.5263
r3236014.female.17.Arts.Virgo.xml	113	0.0729	1.0000	0.1359	0.0000	0.0000	0.0000
Total Training Posts:	**1,037**						

Table 17. Naïve Bayes: Data Set 1 (10 authors) F-scores

Authors of Data Set 1 in Data Set 3 (100 authors)							
Target Author	**Training Posts**	**Baseline Precision**	**Baseline Recall**	**Baseline F-score***	**Precision**	**Recall**	**F-score**
r463180.male.24.indUnk.Taurus.xml	101	0.0025	1.0000	0.0051	0.2222	0.3333	0.2667
r3388015.male.23.Student.Leo.xml	98	0.0042	1.0000	0.0084	0.6000	0.6000	0.6000
r3348302.female.25.indUnk.Virgo.xml	98	0.0051	1.0000	0.0101	0.3571	0.4167	0.3846
r2323827.female.25.indUnk.Pisces.xml	104	0.0030	1.0000	0.0059	0.0000	0.0000	0.0000
r2862338.male.16.Student.Libra.xml	104	0.0038	1.0000	0.0076	0.3077	0.4444	0.3636
r1008329.female.16.Student.Pisces.xml	104	0.0051	1.0000	0.0101	0.0909	0.0833	0.0870
r2016512.female.17.Student.Taurus.xml	110	0.0025	1.0000	0.0051	0.3333	0.3333	0.3333
r2303699.female.23.Science.Aquarius.xml	102	0.0059	1.0000	0.0118	0.0000	0.0000	0.0000
r3051042.female.17.Student.Capricorn.xml	103	0.0055	1.0000	0.0109	0.2381	0.3846	0.2941
r3236014.female.17.Arts.Virgo.xml	113	0.0030	1.0000	0.0059	0.0000	0.0000	0.0000
Total Training Posts:	**20,969**						

Table 18. Naïve Bayes: Subset of Data Set 3 (100 Authors) F-scores

Authors of Data Set 1 in Data Set 4 (1000 authors)							
Target Author	**Training Posts**	**Baseline Precision**	**Baseline Recall**	**Baseline F-score***	**Precision**	**Recall**	**F-score**
r463180.male.24.indUnk.Taurus.xml	83	0.00088	1.00000	0.00176	0.03704	0.04167	0.03922
r3388015.male.23.Student.Leo.xml	84	0.00088	1.00000	0.00176	0.42105	0.33333	0.37209
r3348302.female.25.indUnk.Virgo.xml	92	0.00066	1.00000	0.00132	0.11538	0.33333	0.17143
r2323827.female.25.indUnk.Pisces.xml	79	0.00117	1.00000	0.00234	0.00000	0.00000	0.00000
r2862338.male.16.Student.Libra.xml	87	0.00095	1.00000	0.00190	0.18627	0.73077	0.29688
r1008329.female.16.Student.Pisces.xml	93	0.00084	1.00000	0.00168	0.07407	0.08696	0.08000
r2016512.female.17.Student.Taurus.xml	88	0.00102	1.00000	0.00205	0.33333	0.17857	0.23256
r2303699.female.23.Science.Aquarius.xml	92	0.00088	1.00000	0.00176	0.00000	0.00000	0.00000
r3051042.female.17.Student.Capricorn.xml	89	0.00099	1.00000	0.00197	0.24390	0.37037	0.29412
r3236014.female.17.Arts.Virgo.xml	99	0.00077	1.00000	0.00154	0.00000	0.00000	0.00000
Total Training Posts:	**109,110**						

Table 19. Naïve Bayes: Subset of Data Set 4 (1000 Authors) F-scores

Table 20 lists the scores achieved on the authors of Data Set 2. Table 21 and 22 show the scores achieved on these authors when they were part of a larger data set.

Data Set 2 (10 authors)							
Target Author	**Training Posts**	**Baseline Precision**	**Baseline Recall**	**Baseline F-score***	**Precision**	**Recall**	**F-score**
r899153.female.27.Religion.Gemini.xml	372	0.0686	1.0000	0.1284	0.6667	0.3871	0.4898
r1711947.male.17.Non-Profit.Capricorn.xml	373	0.0730	1.0000	0.1361	0.8824	0.9091	0.8955
r1197361.female.34.indUnk.Taurus.xml	368	0.0885	1.0000	0.1626	0.8049	0.8250	0.8148
r658958.male.24.Communications-Media.Le	381	0.0907	1.0000	0.1663	0.7188	0.5610	0.6301
r109656.male.36.LawEnforcement-Security.I	380	0.1062	1.0000	0.1920	0.9091	0.2083	0.3390
r3025353.male.35.Religion.Aquarius.xml	401	0.1128	1.0000	0.2028	0.8367	0.8039	0.8200
r316316.female.24.Education.Virgo.xml	409	0.1106	1.0000	0.1992	0.7368	0.5600	0.6364
r778441.male.27.Technology.Libra.xml	429	0.1173	1.0000	0.2099	0.8276	0.9057	0.8649
r686878.male.23.Sports-Recreation.Scorpio.	447	0.1040	1.0000	0.1884	0.8750	0.4468	0.5915
r698753.female.24.indUnk.Libra.xml	449	0.1283	1.0000	0.2275	0.7727	0.5862	0.6667
Total Training Posts:	**4,009**						

Table 20. Naïve Bayes: Data Set 2 (10 Authors) F-scores

Authors of Data Set 2 in Data Set 3 (100 authors)							
Target Author	**Training Posts**	**Baseline Precision**	**Baseline Recall**	**Baseline F-score***	**Precision**	**Recall**	**F-score**
r899153.female.27.Religion.Gemini.xml	363	0.0169	1.0000	0.0333	0.3111	0.3500	0.3294
r1711947.male.17.Non-Profit.Capricorn.xml	365	0.0174	1.0000	0.0341	0.5714	0.7805	0.6598
r1197361.female.34.indUnk.Taurus.xml	364	0.0186	1.0000	0.0366	0.3913	0.6136	0.4779
r658958.male.24.Communications-Media.Le	369	0.0224	1.0000	0.0439	0.3714	0.4906	0.4228
r109656.male.36.LawEnforcement-Security.I	383	0.0190	1.0000	0.0374	0.5000	0.2444	0.3284
r3025353.male.35.Religion.Aquarius.xml	397	0.0233	1.0000	0.0455	0.4348	0.7273	0.5442
r316316.female.24.Education.Virgo.xml	412	0.0199	1.0000	0.0390	0.2955	0.2766	0.2857
r778441.male.27.Technology.Libra.xml	442	0.0169	1.0000	0.0333	0.3441	0.8000	0.4812
r686878.male.23.Sports-Recreation.Scorpio.	440	0.0229	1.0000	0.0447	0.6522	0.2778	0.3896
r698753.female.24.indUnk.Libra.xml	464	0.0182	1.0000	0.0357	0.5652	0.6047	0.5843
Total Training Posts:	**20,969**						

Table 21. Naïve Bayes: Subset of Data Set 3 (100 Authors) F-scores

Authors of Data Set 2 in Data Set 4 (1000 authors)							
Target Author	**Training Posts**	**Baseline Precision**	**Baseline Recall**	**Baseline F-score***	**Precision**	**Recall**	**F-score**
r899153.female.27.Religion.Gemini.xml	318	0.00311	1.00000	0.00620	0.04028	0.20000	0.06706
r1711947.male.17.Non-Profit.Capricorn.xml	328	0.00286	1.00000	0.00569	0.25926	0.80769	0.39252
r1197361.female.34.indUnk.Taurus.xml	327	0.00296	1.00000	0.00591	0.10233	0.54321	0.17221
r658958.male.24.Communications-Media.Le	343	0.00289	1.00000	0.00577	0.07449	0.41772	0.12644
r109656.male.36.LawEnforcement-Security.I	356	0.00264	1.00000	0.00526	0.11458	0.15278	0.13095
r3025353.male.35.Religion.Aquarius.xml	356	0.00351	1.00000	0.00700	0.09677	0.71875	0.17058
r316316.female.24.Education.Virgo.xml	351	0.00395	1.00000	0.00788	0.14563	0.27778	0.19108
r778441.male.27.Technology.Libra.xml	375	0.00392	1.00000	0.00780	0.12821	0.70093	0.21676
r686878.male.23.Sports-Recreation.Scorpio.	400	0.00344	1.00000	0.00686	0.41818	0.24468	0.30872
r698753.female.24.indUnk.Libra.xml	395	0.00410	1.00000	0.00817	0.32278	0.45536	0.37778
Total Training Posts:	**109,110**						

Table 22. Naïve Bayes: Subset of Data Set 4 (1000 Authors) F-scores

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] E. Stamatatos, “A survey of modern authorship attribution methods,” J. Am. Soc. Inf. Sci. Technol., vol. 60, pp. 538–556, 2009.

[

[2] G. T. Gehrke, "Authorship discovery in blogs using Bayesian classification with corrective scaling," M.S. thesis, Naval Postgraduate School, Monterey, CA 2008.

[3] J. Rudman, "The state of authorship attribution studies: Some problems and solutions," *Computers and the Humanities,* vol. 31, pp. 351–365, 1998.

[4] T. C. Mendenhall, *The Characteristic Curves of Composition : [Word Lengths in the Writings of Dickens, Thackeray and Others].* New York: Science Co, 1887.

[5] O. de Vel, A. Anderson, M. Corney and G. Mohay, "Mining e-mail content for author identification forensics," *SIGMOD Rec.,* vol. 30, pp. 55–64, 2001.

[6] J. F. Burrows, "Word-patterns and story-shapes: the statistical analysis of narrative style," *Literary and Linguistic Computing,* vol.2, no.2, pp. 61, 1987.

[7] J. F. Burrows, "Not unless you ask nicely: the interpretative nexus between analysis and information," *Literary and Linguistic Computing,* vol.7, no.2, pp. 91, 1992.

[8] M. Koppel, J. Schler and E. Bonchek-Dokow, "Measuring differentiability: unmasking pseudonymous authors," *Journal of Machine Learning Research,* vol. 8, pp. 1261–1276, 2007.

[9] D. Madigan, A. Genkin, D. Lewis, S. Argamon, D. Fradkin and L. Ye, "Author identification on the large scale," *Proceedings of CSNA-05,* 2005.

[10] G. T. Gehrke, C. Martell, A. Schein and P. Anand, "Projecting away the class imbalance problem in author attribution," *IJSC*, Forthcoming.

[11] M. Koppel and J. Schler, "Exploiting stylistic idiosyncrasies for authorship attribution." *Proceedings of IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis,* pp. 69–72, 2003.

[12] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* London: Prentice Hall, Pearson Education International, 2009.

[13] M. Koppel, J. Schler, S. Argamon and E. Messeri, "Authorship attribution with thousands of candidate authors," *Proceedings of the 29th ACM SIGIR,* pp. 659, 2006.

[14] A. Abbasi and H. Chen, "Applying authorship analysis to extremist-group Web forum messages," *Intelligent Systems, IEEE,* vol. 20; 20, pp. 67–75, 2005.

[15] B. Kjell and O. Frieder, "Discrimination of authorship using visualization," *Information Processing & Management,* vol. 30, pp. 141–150, January 1994.

[16] R. S. Forsyth and D. I. Holmes, "Feature-finding for text classification," *Literary and Linguistic Computing,* vol. vol.11, no.4, pp. 163, 1996.

[17] F. Peng, D. Schuurmans, S. Wang and V. Keselj, "Language independent authorship attribution using character level language models," in *EACL '03: Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics,* 2003, pp. 267–274.

[18] V. Keselj, F. Peng, N. Cercone and C. Thomas, "N-gram-based author profiles for authorship attribution." *Proceedings of the Pacific Association for Computational Linguistics,* pp. 255–264, 2003.

[19] E. Stamatatos, "Ensemble-baseed author identification using character n-grams," *Proceedings of the 3rd International Workshop on Text-Based Information Retrieval,* pp. 41–46, 2006.

[20] R. Zheng, J. Li, H. Chen and Z. Huang, "A framework for authorship identification of online messages: writing-style features and classification techniques," *J. Am. Soc. Inf. Sci. Technol.,* vol. 57, pp. 378, 2006.

[21] E. N. Forsyth, "Improving automated lexical and discourse analysis of online chat dialog," M.S. thesis, Naval Postgraduate School, Monterey, CA, 2007.

[22] M. Koppel and J. Schler, "Authorship verification as a one-class classification problem," in *ICML '04: Proceedings of the Twenty-First International Conference on Machine Learning,* 2004, pp. 62.

[23] I. H. Witten and T. C. Bell, "The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression," *Information Theory, IEEE Transactions on,* vol. 37, issue 4, pp. 1085–1094, 1991.

[24] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Harvard University, Tech. Rep. TR-10-98, 1998.

[25] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing.* Cambridge, Mass.: MIT Press, 6th edition, 2003.

[26] B. Dorr and C. Monz. (October 6, 2004). CMSC 723: Introduction to computational linguistics, lecture 5. [Online]. Available: http://www.umiacs.umd.edu/~christof/courses/cmsc723-fall04/lecture-notes/Lecture5-smoothing-6up.pdf (accessed August 14, 2009).

[27] M. Hammond. (2003). Linguistics 696f: Smoothing. [Online]. Available: http://dingo.sbs.arizona.edu/~hammond/ling696f-sp03/ho5-696f.pdf (accessed August 14, 2009).

[28] J. P. Lewis. (December 2004), A short SVM (support vector machine) tutorial. [Online]. Available: http://scribblethink.org/Work/Notes/svmtutorial.pdf (accessed June 26, 2009).

[29] J. Tam, "Detecting age in online chat," M.S. thesis, Naval Postgraduate School, Monterey, CA, 2009.

[30] E. Alpaydin, *Introduction to Machine Learning.* Cambridge, Mass.: MIT Press, 2004.

[31] Wikipedia. (July 28, 2009). Support vector machine. [Online]. Available: http://en.wikipedia.org/wiki/Support_vector_machine (accessed August 1, 2009).

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

[33] J. Schler, M. Koppel, S. Argamon and J. W. Pennebaker. (2006) The blog authorship corpus. [Online]. Available: http://u.cs.biu.ac.il/~koppel/BlogCorpus.htm (accessed August 20, 2009)

.

[34] Google. (2009). Google search for terms: "dear susan the universal scapegoat of blogs" "July" "2004". [Online]. Available: http://www.google.com (accessed January 30, 2009).

[35] Google. (2009). Google search for terms: "dear susan the universal scapegoat of blogs" "2004". [Online]. Available: http://www.google.com (accessed January 30, 2009).

[36] Blogger. (2009). Blogger. [Online]. Available: http://www.blogger.com (accessed August 21, 2009).

[37] Google Research Blog. (August 3, 2006). Google research: All our n-gram are belong to you. [Online]. Available: http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html (accessed August 24, 2009)

[38] R. Fan, K. Chang, C. Hsieh, X. Wang and C. Lin, "LIBLINEAR: A Library for Large Linear Classification," *Journal of Machine Learning Research,* vol. 9, pp. 1871–1874, 2008.

[39] R. Fan, K. Chang, C. Hsieh, X. Wang and C. Lin. (August 8, 2009). LIBLINEAR: A library for large linear classification (download site). [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/liblinear (accessed August 20, 2009).

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Fort Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

5. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

6. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California

7. Marine Corps Tactical Systems Support Activity (Attn: Technical Director)
   Camp Pendleton, California

8. Dr. Craig Martell
   Naval Postgraduate School
   Monterey, California

9. Dr. Andrew Schein
   Naval Postgraduate School
   Monterey, California

10. Captain David Dreier
    Marine Corps Tactical Systems Support Activity
    Camp Pendleton, California