Spring 2014

The Open Class Authorship Attribution Problem: A Comparison of Mixture-of-Experts Methods within the JGAAP Framework

James Orlo Overly

THE OPEN CLASS AUTHORSHIP ATTRIBUTION PROBLEM: A COMPARISON OF MIXTURE-OF-EXPERTS METHODS WITHIN THE JGAAP FRAMEWORK

A Thesis

Submitted to the McAnulty Graduate School of Liberal Arts

Duquesne University

In partial fulfillment of the requirements for

the degree of Master of Science

By

James O. Overly

May 2014

THE OPEN CLASS AUTHORSHIP ATTRIBUTION PROBLEM: A COMPARISON OF MIXTURE-OF-EXPERTS METHODS WITHIN THE JGAAP FRAMEWORK

By

James O. Overly

Approved November 19, 2013

Patrick Juola, Ph.D. Associate Professor Thesis Advisor, Committee Chair	John Kern, Ph.D. Associate Professor Committee Member
Donald Simon, Ph.D. Associate Professor Director of Graduate Studies	James C. Swindal, Ph.D. Dean, McAnulty College

ABSTRACT

THE OPEN CLASS AUTHORSHIP ATTRIBUTION PROBLEM: A COMPARISON OF MIXTURE-OF-EXPERTS METHODS WITHIN THE JGAAP FRAMEWORK

By

James O. Overly

May 2014

Thesis supervised by Patrick Juola, Ph.D.

In this paper, we seek to describe, test, evaluate, and compare methods of open class attribution that utilize multiple unique closed class attributions in a voting framework. By applying statistical techniques to the proportion of closed class attributions indicating individual candidate authors, we seek to determine if the author is present in the set of suspected authors or not. The final answer to an open class attribution problem is either one of the authors in the set of candidate authors or "None of the above."

We test nine different methods of open class attribution grouped into three distinct voting paradigms. We find that the most effective method is a voting method in which each closed class attribution votes equally for its top two most likely authors. Accuracies in this method are statistically better than chance and, in total, are the best out of all nine methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ABBREVIATIONS

$S(k = k_0)$ - Single vote method in which each voter gets one vote and casts it for one candidate. The number of voters is decided by equation 6 with a k value of k_0

R_0 - Runoff voting method in which each voter gets one vote and casts it for one candidate. After each round of voting ends, the candidates whose vote proportions are significantly lower that the leader's proportion are removed from the candidate set and a new round of voting commences.

R_A - Runoff voting as above; however, after each round of voting, the number of voters the subsequent round is determined using equation 6 to find a sufficient sample size to detect the difference between the top two contenders.

$R_{A_{CI}}$ - Runoff voting as with R_A; however, a 95% confidence interval is placed around the difference between the top two contenders. The number of voters in the subsequent round is determined using equation 6 to find a sufficient sample size to detect the infimum of this confidence interval.

A_n - Approval voting method in which each voter may apportion n equally weighted votes among candidates. Multiple votes for one candidate are allowed.

A_{ALL} - Approval voting method in which a candidate's "distance" from perfection determines the proportion of the vote that the candidate will receive. For example, a candidate twice as far away from perfection as another will receive half the vote amount that the other receives.

1 Introduction

Authorship attribution, and the related field of authorship verification, is the process by which a third party can make educated statements concerning the true author of an anonymously or disputably authored document. Closed-class authorship attribution is a technique in which an author is chosen from a given set of candidates. Open-class authorship attribution, on the other hand, is a technique in which either an author is chosen from a given candidate set or it is stated that the author is not among those in the candidate set. Numerous researchers have cited the inability of authorship attribution methods to deal with open candidate sets and possibly small data sets as one of the fundamental problems of modern authorship attribution studies. [5] [9] [10] [12] [18] Multivariate methods for analyzing and assigning an author to a text must move beyond small closed sets of authors. The introduction of a "Don't Know" category has been posited, implying that the algorithm was uncertain of the candidate author to which a document should be assigned. We wish to move one step further and use statistical techniques to reliably determine if an author is or is not in a candidate set. The method we use relies on the multitudinous methods of authorship attribution accessible through the Java Graphical Authorship Attribution Program (JGAAP). Though each method will give as an answer one of a small closed author set, we are able to combine many of these decisions into one overarching decision which allows a "None of the Above" answer. Statistical methods are used to generate the number of methods from JGAAP to utilize, as well as to make the final decision concerning which author to attribute (or not attribute) to the document in question.

2 Background

In this section we provide an overview of types of attribution, methods of attribution, and their applications. Following that, we briefly discuss the JGAAP framework which was the testing environment for our experiments. We then examine how the use of committee machines can be used to create a new attribution method which we dub *mixture-of-experts*

attribution. Finally, we explain three separate voting principles that were adapted for use in mixture-of-experts attribution experiments.

2.1 Types of Attributions

Authorship attribution, and the related field of authorship verification, is the process by which a third party can make educated statements concerning the true author of an anonymously or disputably authored document. A closed class authorship attribution uses a list of candidate authors. From this list, the program is forced to select one of the authors that is "most like" the author of the disputed document. This chosen author is then reported as the true author. An open class authorship attribution uses a list of candidate authors as well; however, it will either select one of the authors in the list or state than none of the listed authors is the true author. Authorship verification is essentially an open class attribution with a set of only one suspected author. The answer is either that the suspected author is the true author or not.

One can liken the three types of attribution to exercises used by law enforcement to attribute a crime to someone. The case of a closed class attribution can be thought of as asking a witness to choose the person that is most like the criminal from a small group of suspects. In this case, the witness MUST choose someone in the group, even if the true perpetrator is not in the group at all. An open class attribution would have the same group of suspects; however, unlike the closed-class case, "None of the above," is an acceptable answer to an open class attribution. Finally, authorship verification would have only one suspect. The witness would be asked if that suspect was the criminal. The only acceptable answers are "Yes" or "No."

2.2 Methods of Authorship Attribution

Authorship attribution was a natural outgrowth of one human's analysis of another. Handwriting recognition, fingerprint analysis, voice recognition, and psychological profiling are all essentially methods of attempting to accurately assign a characteristic to another

person. During the time of Morse code, operators were anecdotally able to identify other operators merely by the way that they tapped in the same letters or message, called an operators "fist." Today some people can determine whether a social networking message, text message, or Twitter tweet was sent by the owner of an account or a hacker by the use (or misuse) of punctuation, abbreviations, or certain words. [17] Human analysis has been a powerful and driving force underneath many of the computational methodologies applied today. In the mid 1900's, nontraditional methods, meaning simply methods other than human analysis, began to arise. These mainly featured computer programs identifying features in the text and statistical analyses of these features. Most methods can be described in one of three ways [8]:

- Unitary invariant - Methods in this class seek to examine one feature or function of the text that will assumedly identify the author.
- Multivariate analysis - Methods in which statistical multivariate analysis is applied to numerous textual feature sets to discriminate between authors
- Machine Learning - Modern machine learning methods are applied to numerous training documents to create new classifiers before being applied to anonymous documents.

The features analyzed by these methods are inherent in a set of documents. For example, one of the first nontraditional methods of authorship attribution looked at function words, words such as: *of*, *be*, *in*, and *the*, that carry little individual content meaning but rather serve to express relationships with other words in a sentence. The theory was that an author does not consciously choose these words but rather the words (and more importantly, the frequencies of these words) will flow naturally from an author's personal style of writing [15]. Other simple measures that are assumed to be indicators of individual authorship include the following [8]:

- Complexity measures, such as word length, sentence length, syllables, letter distribution, number of words per sentence, or more complicated measures such as words appearing only once (*hapax legomena*) or twice (*dis legomena*) in a document.

- Function words, as described above.
- Parts of speech, such as noun, transitive or intransitive verb, direct or indirect object, and, often, the ordering of these parts of speech.
- Functional lexical taxonomies, which classify words using trees rooted with parts of speech and terminate with leaves of words themselves. In between can be a number of other classifications. An example trace from root to leaf may be *conjunction* $\rightarrow$ *conjunctionEnhancement* $\rightarrow$ *conjunctionSpatiotemporal* $\rightarrow$ ***beforehand***.
- Content words, or words that would not be considered function words. The set is obviously larger, but sets such as synonym choice (using "big" instead of the word "large" or vice-versa) or rare words can be used as feature sets.
- Character n-grams, or blocks of a document "n" characters in length. For example, the character bigrams of the word *quick* are "qu", "ui", "ic", and "ck". Frequencies of character n-grams have been shown to apply relatively well across a number of disciplines.
- Morphological analysis, the analysis of meaningful prefixes and suffixes. This feature set is more useful in languages such as Greek or Hebrew with a richer morphology than English.
- Error analysis, the examination of common written or typed errors such as repeated letters, letter substitution, letter inversion, or conflated words (missing space between words, such as *no one* becoming *noone*).
- Formatting and structure, especially in digital media such as email, blogs, and computer code.

The main idea is that any of these measures should be able to inform an investigator as to which of a set of documents is most like an anonymous document. However, there is no silver bullet; some features may work better than others on average or across a wider range of scenarios, but none performs perfectly in every scenario. Thus, research into evaluation

of known features, and discovery or creation of new features of a document that can be used in an authorship attribution process, is continuous and meaningful. [3] [5]

2.3 Applications of Authorship Attribution

So why would we wish to perform one of these attributions? There are numerous applications.

2.3.1 Disputed Authorship

The first application is obviously to properly attribute the author of disputed, anonymously, or pseudonymously written works. Many times, influential works are printed anonymously or pseudonymously due to the nature of the material. Inflammatory works are often published anonymously in newspapers or pamphlets to protect the author's reputation, social standing, or even life. After time passes, interested parties may wish to determine the true author of a work or set of works. Sometimes, many different authors will have claimed authorship by that point. At other times, no one may have ever claimed authorship, but experts have a fairly good idea of a small set of people who could have penned the works. Other times, a work may be included in another author's corpus based on its subject matter and time period even though he or she did not pen it.
Specific historic applications of this type of authorship abound. Mosteller and Wallace's [15] investigation into the true authorship of the disputed articles of the Federalist Papers is often referred to as one of the first applications of non-traditional authorship attribution, meaning that the attribution was not based on a human's professional opinion. All of the articles were published under the pseudonym of "Publius" but are believed to be the works of James Madison, Alexander Hamilton, and John Jay. After his death, a list of authorship made by Alexander Hamilton was found. James Madison also had a list that did not agree with Hamilton's in 12 of the 73 essays. Mosteller and Wallace used an analysis of function words to attribute many of the disputed works to Madison. Numerous statistical analyses and applications of authorship attribution have upheld Madison's list as the most likely list of true authors. [5] [14] [18]

Another significant case of authorship attribution involves one of the greatest literary figures in history: Shakespeare. Some scholars take issue with the discrepancy between Shakespeare's reputation and his verse. From the 19th century to present, dozens of candidates have been hypothesized as the true author of some of the bard's plays, some less seriously than others. Few have attracted any significant number of believers; among them are Sir Francis Bacon; Edward deVere, 17th Earl of Oxford; Christopher Marlowe; and William Stanley, 6th Earl of Derby. [13] [21] Various reasons for believing in one candidate over another can be construed; the point, however, is that the documents are disputed even though they have historically been attributed to one source.

More recently, J. K. Rowling, the author of the famous *Harry Potter* books was the center of an authorship investigation. The book *The Cuckoo's Calling* was written and published in April of 2013. The author was supposedly a new writer by the name of Robert Galbraith. However, linguistic analysis using the JGAAP system revealed that the new writer's book was remarkably similar in many ways to a previous work of Rowling's, *Casual Vacancy*. Rowling confessed to the pseudonym after the analysis had been run and the results had been independently confirmed using a separate system from the United Kingdom. [4]

2.3.2 Multiple Authorship

A second, related application is attribution of works written by multiple authors; these works are not the result of a single author, but rather are an amalgamation of several authors. These authors may be working in concert or they may be responsible for a series of editions, each slightly different than the last. Attribution techniques can be used to tease apart which sections of a given writing are the work of the same author without knowing who the authors are. Numerous historical documents fall under this category. In particular, many sections of the Bible have been studied to determine with confidence which sections were written by the same person. The synoptic gospels (Matthew, Mark, and Luke) are theoretically each written by the author for which they are named; however, numerous Bible critics have explained the striking parallelism between the documents as

evidence of authorship by two to four different authors. Even though we do not know the authors themselves or any of their other writings, we can identify the segments of a document (or set of documents) that were written by the same person or taken from the same source. [7]

2.3.3 Forensic Analysis

There are also numerous applications of authorship attribution in the field of forensics. Attribution methods can be used to analyze threatening letters, emails, and blog posts. Determining the author of these writings could prohibit intended violence from occurring. [11] As techniques become more robust and accurate, they may be used as evidence in courtrooms or as probable cause for warrants or arrests. These models could also be used to analyze suicide notes to determine whether a suicide was genuine or if there was foul play involved with an attempt to cover it up. There are also applications in cybercrime; malicious source code and computer viruses may be able to be attributed to one of a number of hackers or cybercriminals based on other instances of their computer code. [16] An area related to authorship attribution, authorship profiling, also is useful in forensics and benefits greatly from research into attribution methods. In this problem, we do not try to use a written work to identify an author by name, but rather by description: gender, age, race or ethnicity, geographic location, psychological profile, etc. This could be used to narrow down lists of suspects in a crime or even to determine where to look for suspects. [5] [11]

2.3.4 Plagiarism Detection

Finally, authorship attribution can be an indispensible resource for plagiarism detection. Research in this field allows us to determine accurately whether one person has indeed done his or her own work, or not (another person's writing, a copy-and-pasted paper from the internet, etc.). [19] Teachers assigning and grading research papers, college professors and advisers identifying original research, and publishers and editors looking at submitted papers, essays, books, and articles are all people who find plagiarism detection applicable

and useful. Consider that turnitin.com, the website for a leading plagiarism detection software for educators and students, claims over a million faculty users and 20 million student users. Any advances made in the area of plagiarism detection are immediately usable and even marketable to the general population.

2.4 The JGAAP Framework

The Java Graphical Authorship Attribution Program (JGAAP) is a free, modular, Java-based program available through the Evaluating Variation in Language Laboratory at evllabs.com/jgaap/w. It is used for textual analysis, text categorization, and authorship attribution, specializing in closed class attribution methods. Given a list of documents with known authors, called training documents, and a number of documents with unknown authorship, the program will analyze the training documents using user-defined parameters. The unknown document is then attributed to the author of the training document most closely resembling the unknown document within the user's parameters. The parameters are canonicizers, event sets, and analysis methods. Canonicizers are actions taken to isolate, remove, or unify certain characteristics of documents, such as unifying the case (all lowercase) or stripping the punctuation. Multiple canonicizers can be used in one experiment. Event sets reduce the documents to a set of identifiable and comparable items that are to be closely inspected, such as the sentence length, character n-grams, or parts of speech. Analysis methods detail how the documents' event sets are to be compared. These include a varied selection of distance-based and graphical methods. Only one event set and analysis method can be selected per experiment. Most methods fall under the class of multivariate invariant methods from above; they analyze a numerical distribution associated with a certain textual or stylistic feature for each document. The most important thing that JGAAP offers is a large and varied set of each of the three parameters, meaning that thousands of different methods can be created. Different conclusions may be reached by different combinations of parameters. This, coupled with the digital freedom to rewrite and adapt the Java code to a given experiment, makes JGAAP an excellent testing ground for authorship attribution experiments

2.5 Committee Machines

Committee machines, or ensemble machines, are machine learning methods that combine numerous individual classifications to arrive at a single classification.[20] A majority vote is a simple type of committee machine. Numerous entities are asked to give an answer to a single question; each entity gives an answer, and the answer that occurs the most often is considered the committee machine's answer to the question. Committee machines will reduce statistical and computational error as long as the individual committee members are relatively independent (errors are uncorrelated) and reasonably accurate (better than random guessing). [20] Methods have been developed to control the correlation of errors among similar committee members as well as to boost weak learners (slightly better than random guessing) to strong learners (probably approximately correct). There are variations of this idea that include weighting the decisions of committee members overall or assigning different weights based on the relevance of committee members to the subject.

2.6 Mixture-of-Experts Open Class Attribution

Given that there are a substantial number of closed class methods of authorship attribution, we seek to use these numerous closed class attribution methods to build an open class attribution method by creating a committee machine. Each closed class method will act like an expert on one particular characteristic of authorial style. We can conceive of each individual closed class attribution method as "voting" for the candidate author that it regards as the most likely true author. We begin by making the not-unreasonable assumption that each individual closed class attribution method performs at some level better than random chance. In other words, if there are N authors being tested, then any given attribution method will return the correct author with probability greater than $\frac{1}{N}$. Thus, by performing a sufficient number of independent attributions and keeping track of the number of attributions for each individual candidate author, we should be able to infer the true author of an unattributed document by comparing the proportion of these attributions in favor of each author. Under this assumption, by the law of large numbers,

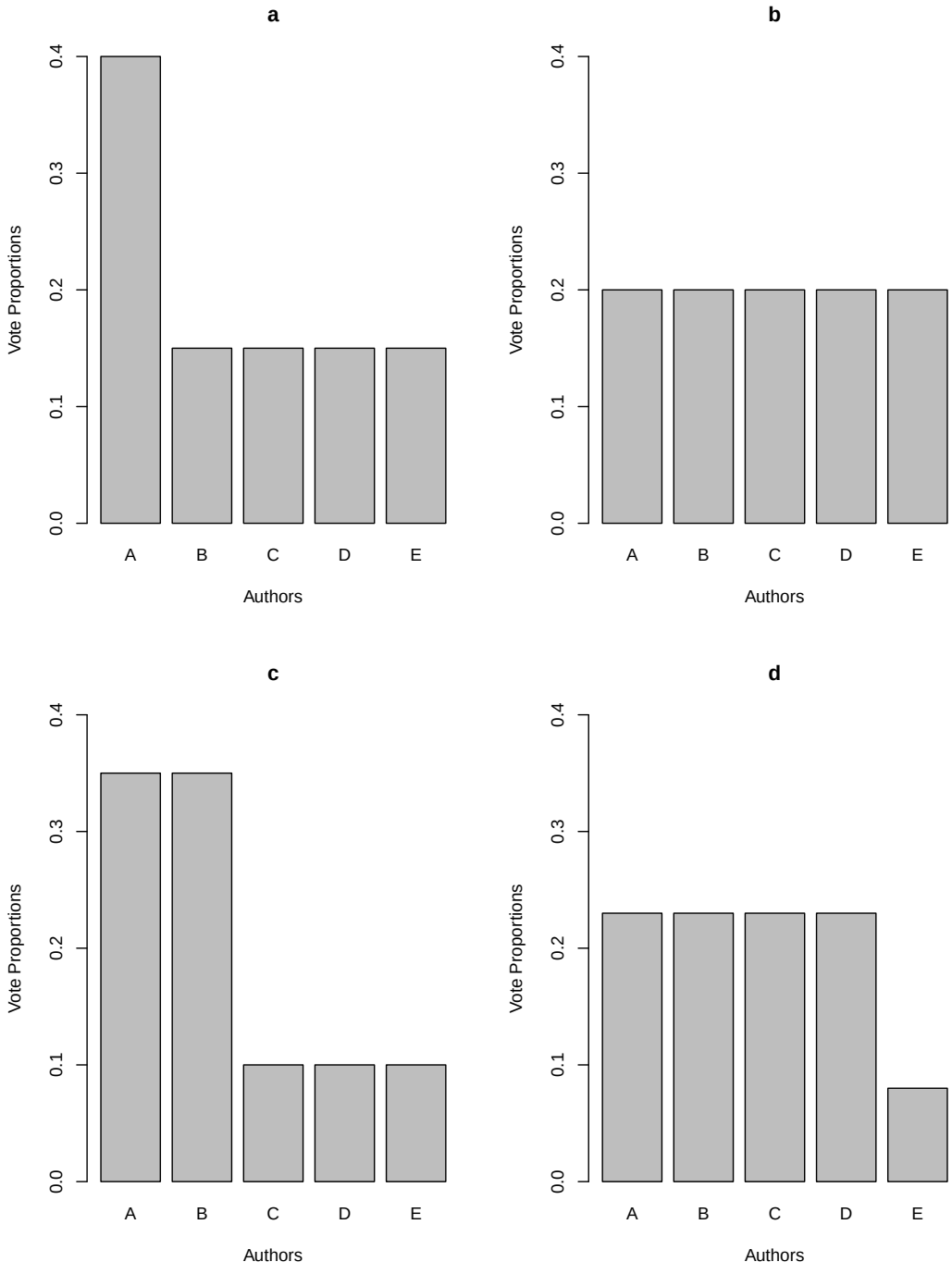

Figure 1: Mixture-of-Experts Open Class Attribution Voting Proportion Examples

the true author will gain a higher proportion of the attributions.

This can be thought of in terms of the "Ask the Audience" lifeline from the television game show *Who Wants To Be A Millionaire?* Each member of the audience is asked to choose the answer to the question to the best of their ability, even if they are not positive their answer is correct. The idea is that the true answer will receive a significantly higher proportion of votes than the incorrect answers, as in Figure 1a, provided the number of persons polled is large enough.

Up to this point, the method makes use of a committee machine to create a new and hopefully more accurate closed class attribution method. To create an open class attribution method, we must be able to interpret some subset of the possible outcomes as a "None of the above" answer. To do this, we look at the case where there is one or more authors who cannot be judged as significantly different from the author with the highest proportion of attributions. In other words, not enough of our "experts" agree that a given author is the true author, as in Figures 1b, 1c and 1d. In this case, we can make the claim that, given that we performed a sufficient number of attributions in the first place, the true author would have received a significantly higher proportion of attributions *if* s/he had been present in our set of possible authors. Thus, statistical "ties" are interpreted as indicating that the true author is not among the candidate authors.

2.7 Voting Principles and Applications

The mixture-of-experts system can be implemented in numerous ways. It can be used as a simple and straightforward single vote method; it can be used as a method for evaluating a system of approval voting; or it can be used in conjunction with rounds of voting and culling, as in runoff voting. The main question is: Which, if any, of these implementations performs better than any of the others?

2.7.1 Single Vote Method

The case of a single vote open class attribution is straightforward: run a number of closed class attributions (the number to be determined by the mathematics in the following

sections) and use each attribution as one vote for the attributed author. We perform inference on the resulting proportions in order to answer the open class attribution question. This is the simplest method, where every attribution is essentially one "expert's" vote in favor of one author from our candidate set; we search only for the author who receives the highest number of votes. This method we refer to as a single vote open class attribution, abbreviated as S.

2.7.2 Runoff Voting Methods

Runoff voting methods assume that we are not necessarily immediately looking for the true author, but rather that we are attempting to weed out the obvious distractor authors to give us a better pool of potential authors from which to draw conclusions. "Obvious distractor", in the case of this set of experiments, means "having statistically significant lower proportions of votes than the highest voted author." Therefore, each round partitions the set of authors into two disjoint sets: the authors who are not statistically different from the "most likely" author, and the authors who are statistically different. The training documents written by the authors in the second set are removed from the experiment. This effectively removes these authors from the pool of candidate authors. The experiment is then run again with the newest author set until a conclusion concerning the authorship can be reached. If only one author remains after a round of voting, that author is declared the true author of the document. If the experiment reduces to a case where none of the authors are statistically different from the highest voted author, then the claim is that none of the above authors can be attributed to the document. This method we call runoff voting (with no adjustments), abbreviated R_0.

This runoff voting can take place in many different ways. The simplest is the method described above; rounds of voting with no changes to the parameters between rounds. However, from a statistical point of view, we have new information after one round of voting has commenced. Take as an example the case of four potential authors (A, B, C, and D) with the following vote proportions: $[\hat{p}_A = .45, \hat{p}_B = .4, \hat{p}_C = .07, \hat{p}_D = .08]$. Authors C and D would be dismissed, but another round would be required to determine

anything about authors A and B. However, in light of the first experiment, we *expect* to get a difference $\hat{p}_A - \hat{p}_B = .45 - .4 = .05$. We wish to know if this difference is significant or if it is merely an expected variation of two (actually equal) random variables. To do so, we may adjust our sample size calculations (explained in the methods section of this paper) so that we may test for a variation below our current threshold. Using an expected difference between our proportions $E = .05$ and the fact that there are only 2 authors in the next round (A and B), we can determine a sample size that will be large enough to ensure, with 95% accuracy, that any difference of E or more is a true difference between the two values. This method we call runoff voting with adjusted k value, abbreviated as R_A.

A third option is to assume that the expected difference value E itself is a normal random variable and to place a confidence interval around it using the same calculation for a difference of dependent proportions and choosing the smallest expected difference value that is within the 95% confidence interval. This method we call runoff voting with confidence interval adjusted k value, abbreviated as $R_{A_{CI}}$.

2.7.3 Approval Voting Methods

Approval voting is a plurality voting that allows each member to vote for multiple candidates. Each candidate that is "approved" receives one vote from that voting member. Approval voting allows each voting member to approve of exactly n candidates, allowing the vote to be split among different candidates. This takes into account that each attribution method within the JGAAP framework that was used for the purposes of this experiment works by associating a distance with each training document. These distances represent how closely the training document matched the test document; the smaller the distance, the closer the match with the document being tested.

Obviously, the author of the document with the smallest distance is attributed first place, but the question remains: is second place really as much of a loser as third, or fifth, or tenth place? If one of four authors garners 33% of the first place votes, but takes last place the rest of the time, is it really the best attribution to make? Or is it possible that a second author, who takes first place in only 20% of the attributions but takes second place

the rest of the time, is a more accurate choice? Note that, due to the fact that some authors may have multiple training documents, the same author may receive multiple votes from the same "expert" voter. Thus, the approval voting method can be used in as many ways as the number of training documents (N) for the problem at hand by splitting each attribution's votes among the top n training documents. We wish to normalize each voter to having only one vote. Thus, of the whole vote, $\frac{1}{n}$ of that vote will be added to the authors of each of the top-placed n documents, as long as $n < N$. These methods we call approval methods with n equal votes, abbreviated as A_n.

In the case where $n = N$, the above process would be a useless measure, since each author would get the exact proportion of votes as the proportion of training documents by that author. Therefore, in the case where $n = N$, the proportion of the vote going to each document is proportional to the document's distance from all the other document matches. For example, a document that is at a distance of 4 from a perfect match garners twice as much of the vote as a document at distance 8 while only getting half as much of the vote as a document at a distance of 2. The calculation that provides the exact values for this partitioning is:

$$v_i = \frac{1}{d_i \sum_{j=1}^{n} \frac{1}{d_j}},$$

where v_i is the proportion of a vote to be attributed to the i^{th} document and d_i is the distance associated with the i^{th} document. Note that the sum of all of the vote proportions from each closed class attribution is normalized to one vote. This method we call approval voting using normalized proportional distances, abbreviated A_{ALL}.

3 Materials and Methods

In this section we give an overview of the AAAC corpus as well as the mechanics behind closed class attribution methods available in JGAAP. We also explain the statistical methods used to determine appropriate levels of confidence, inference methods, and sample

sizes to use during experimentation.

3.1 The AAAC Corpus

All open class attribution methods in this set of experiments were tested upon the Ad-hoc Authorship Attribution Competition (AAAC) corpus. This set of documents includes 13 different problem sets labeled A through M. The corpus represents a wide variety of genres, languages, lengths. Each set is also comprised of various numbers of authors and various numbers of training documents for each author.

- Problem A (English) Fixed-topic essays written by 13 US university students.
- Problem B (English) Free-topic essays written by 13 US university students.
- Problem C (English) Novels by 19th century American authors (Cooper, Crane, Hawthorne, Irving, Twain, and "none-of-the-above"), truncated to 100,000 characters.
- Problem D (English) First act of plays by Elizabethan/ Jacobean playwrights (Johnson, Marlowe, Shakespeare, and "none-of-the-above").
- Problem E (English) Plays in their entirety by Elizabethan/ Jacobean playwrights (Johnson, Marlowe, Shakespeare, and "none-of-the-above").
- Problem F ([Middle] English) Letters, specifically extracts from the Paston letters (by Margaret Paston, John Paston II, and John Paston III, and "none-of-the-above" [Agnes Paston]).
- Problem G (English) Novels, by Edgar Rice Burrows, divided into "early" (pre-1914) novels, and "late" (post-1920).
- Problem H (English) Transcripts of unrestricted speech gathered during committee meetings, taken from the Corpus of Spoken Professional American-English.
- Problem I (French) Novels by Hugo and Dumas (pére).

- Problem J (French) Training set identical to previous problem. Testing set is one play by each, thus testing ability to deal with cross-genre data.
- Problem K (Serbian-Slavonic) Short excerpts from The Lives of Kings and Archbishops, attributed to Archbishop Danilo and two unnamed authors (A and B). (Data obtained from Alexandar Kostic.)
- Problem L (Latin) Elegaic poems from classical Latin authors (Catullus, Ovid, Propertius, and Tibullus).
- Problem M (Dutch) Fixed-topic essays written by Dutch university students. (Data obtained from Harald Baayen)

[6]

3.2 Closed Class Methods in JGAAP

Tests were conducted on the AAAC corpus using the JGAAP system and Java code designed to implement mixture-of-experts open class attributions. Each document in the corpus was tested with each of nine different mixture-of-experts attribution methods. The results were recorded and analyzed to compute accuracies for each given method. The accuracies were then further analyzed to determine which attribution methods performed the best.

To perform one closed class attribution in JGAAP, we need three items: a (set of) canonicizers, an event set, and an analysis method. At the time of testing, there were 11 canonicizers, 63 event sets drivers, and 32 analysis methods. This gives a grand total (counting all possible combinations of multiple canonicizers) of $(2^{11} - 1)(63)(32) = 4,128,768$ assumedly distinct closed class attribution methods. This study was unable to make use of all 4 million-plus methods. Some combinations of canonicizers, event sets, and analysis methods are not usable or advisable for one reason or another. For example, pairing a canonicizer which strips all punctuation with an event set that measures sentence length is not a good idea; with all punctuation removed, all of the

documents in question would consist of only one, long sentence! Other examples include items such as the null event set which often result in no usable data regardless of the combination in which they are used, as well as the "JW Cross Entropy" and "Levenshtein Distance" analysis methods that may work at an average level but proved to take a great deal of time when used, slowing down the process of an open class attribution. Many of these items or combinations were specifically excluded from use during the experiments.

For the purposes of these experiments, every closed class attribution was built from a unique triplet of one canonicizer, one event set and one distance-based analysis method. There were 11 canonicizers, 63 event sets drivers, and 14 analysis methods that were distance-based methods. Therefore, the number of methods available for preliminary testing was $(11)(63)(14) = 9702$.

Some canonicizers, event sets, and analysis methods were removed from the sample space for reasons of efficiency. Some items removed based on efficiency were removed based on the amount of time it took the computer to perform a single closed class attribution incorporating that item. Also, some methods will, under certain conditions, return values that make little or no sense, such as claiming all of the candidate documents are exactly like the test document. This amounts to claiming that the "distance" from the test document to each candidate document is 0. Thus, when the program found insufficient variance in the results of a closed class attribution (difference between the maximum and minimum distances from the test document was less than .00001) it attributed randomly among the candidate authors. Items removed based on this measure were items that, in preliminary analyses, had all, or a large proportion of, the closed class attributions that incorporated them assigned at random due to insufficient variance in the results. Items that produced errors at runtime were also removed.

In total, three canonicizers, three event sets, and four analysis methods were removed from the testing sample space. Thus, the true number of methods available during the testing phase of this study was $(8)(60)(10) = 4800$. In order to maintain the efficiency of open class attribution methods requiring large numbers of closed class attributions in each phase, a maximum number of attributions per phase was set at 2500. This allows for

relatively quick open class attributions while still allowing for a high degree of confidence based off the sample size.

As stated above, the closed class methods that were used can be described uniquely by describing their three individual parts: canonicizer, event set, and distance analysis. Appendices A, B, and C give an exhaustive list of definitions of each canonicizer, event set, and analysis method used in this study.

3.3 Multiple Comparisons Tests: Bonferroni Correction

Consider an experiment to be run at the $\alpha = .05$ confidence level consisting of 20 different comparisons under a null hypothesis of equality. If each individual comparison is evaluated at $\alpha = .05$, we are 95% confident that each of our significant comparisons is a true difference and not simply a result of random chance; however, the experiment as a whole has $\alpha = 1 - (.95)^{20} = .6415$ as a confidence level, far higher than the desired $\alpha = .05$. In this case, there is a 64.15% chance that rejecting the null hypothesis of equality is simply due to random chance in one or more of the individual tests. In order to raise the confidence of the experiment to the required level there are numerous methods for adjusting the α values for the individual tests based on the desired overall alpha value and the number of comparisons. This paper makes use of a very simple method known as the Bonferroni correction. Using this method, the α value for each individual test in a multiple comparisons scenario is simply the experiment's desired α value divided by the number of comparisons being made. In the above example, there are 20 comparisons with a desired overall α of .05; this implies that each individual comparison should be carried out at an α level of $.05/20 = .0025$. This gives us 99.75% confidence in each individual comparison and $1 - (.9975)^{20} = .0488$ as the overall α value for the experiment. Notice that the Bonferroni correction is slightly more restrictive than desired, increasing the chance of failing to reject the null hypothesis even when we should have done so at the desired confidence level. However, it can be shown that, no matter the number of comparisons in the experiment, the overall α value can never drop below $1 - e^{-\alpha}$; in the case of $\alpha = .05$, as used in the experiments in this paper, the minimum possible overall α value is approximately .04877.

3.4 Inference

The attributed author of an open class attribution is the author who receives a statistically significantly larger number of attributions, or votes, from a number of closed class attribution methods. Given the proportions of closed class attribution methods that voted for our candidate authors, one must statistically test the difference between each pair. Our null hypothesis is that, for any pair of candidate authors A_i and $A-j$, both proportions we are testing are equal to one another. Thus:

$$H_0 : p_{A_i} = p_{A_j}$$

$$H_a : p_{A_i} \neq p_{A_j}(claim)$$

where p_{A_i} is the true proportion of votes for author i. The proportions are not independent (for example, they cannot sum to more than one). The standard error for the difference between two dependent proportions is

$$SE = \sqrt{\frac{\hat{p}_{A_i} + \hat{p}_{A_j} - (\hat{p}_{A_i} - \hat{p}_{A_j})^2}{n}}, \tag{1}$$

where $\hat{p}_{A_i}$ is the point estimate of the proportion of votes for author i and n is the number of votes in total (see Appendix A for derivation). [1] [2] Therefore, a two-tailed test of the difference between the parameters p_{A_i} and p_{A_j} can be tested using a z-test where the test statistic is

$$z = \frac{\hat{p}_{A_i} - \hat{p}_{A_j}}{\sqrt{\frac{\hat{p}_{A_i} + \hat{p}_{A_j} - (\hat{p}_{A_i} - \hat{p}_{A_j})^2}{n}}}. \tag{2}$$

The z-value returned by (2) represents a probability p from the standard normal table. This probability must be compared to an adjusted α value. The adjustment is determined by the cardinality of the set of authors. If we wish to have a level of confidence α in our comparison of all candidate authors, each of $\frac{N*(N-1)}{2}$ pairwise tests must be conducted at a higher α level. Using a Bonferroni correction for multiple comparisons, we find that the

adjusted alpha level at which our individual tests must be conducted is $\alpha_{adj} = \frac{2\alpha}{N(N-1)}$, where N is the number of authors in the problem. Thus, for the individual 2-tailed test to be considered significant, p must be less than $\frac{\alpha_{adj}}{2} = \frac{\alpha}{N(N-1)}$. For each individual test, we consider $p < \frac{\alpha_{adj}}{2}$ evidence that a true difference does exist between p_{A_i} and p_{A_j}.

3.5 Sample Size

The proportions used in the inference step of the verification process must be the result of a sufficiently large number of trials in order to return reliable data concerning the true differences. However, one hopes to minimize the number of trials necessary to detect a given difference in the proportions. Using the error formula

$$E = z_c * SE \tag{3}$$

and using equation (1), one can solve for n, the number of trials, which takes the form

$$n = \frac{z_c^2[\hat{p}_{A_i} + \hat{p}_{A_j} - (\hat{p}_{A_i} - \hat{p}_{A_j})^2]}{E^2}. \tag{4}$$

where z_c is the critical z-value appropriate for the α level of the experiment and E is the acceptable margin of error about the difference of proportions.

Since each individual test will take place at an adjusted α level, the z-score z_c here must also be adjusted. Thus, let z_{adj} be the z-value that has an area of $\frac{\alpha_{adj}}{2} = \frac{\alpha}{N(N-1)}$ to its right. Let us also use the assumption that no author is a priori assumed to receive any greater proportion of votes than any other, i.e. $\hat{p}_{A_i} = \frac{1}{N}$ for all authors in the candidate set. Then the equation reduces to

$$n = \frac{2z_{adj}^2}{E^2 N}. \tag{5}$$

Finally, if we assume E is inversely proportional to $\frac{1}{N}$, we get

$$n = 2Nk^2 z_{adj}^2, \tag{6}$$

where k is the proportionality constant between E and $\frac{1}{N}$. Thus, as k increases, E decreases, and smaller true differences can correctly be identified at the expense of an increase in sample size.

In order to ensure a minimum amount of accuracy in any method using adjusted k values, a minimum k value should be set as a default for whenever the calculations provide too small of a k value; moreover, since the k value is directly related to the sample size, a maximum sample size should be included in order to prevent an experiment from suggesting an extremely large sample size due to an extremely small expected difference. The experiments conducted herein used a minimum k value of 2 (i.e. true differences of $\frac{1}{2N}$ should be detectable) and a maximum sample size of 2500.

3.6 Notes on Significance

In the discussion we distinguish between two types of significance: statistical and clinical. Statistical significance of a variable or a difference of variables will be assigned based on an appropriate correction for multiple simultaneous tests at the $\alpha = .05$ confidence level; the method used in these experiments is the Bonferroni correction. For example, all open class attribution methods use a threshold of statistical significance to determine which author, if any, to label as the true author of a document. However, in many tests comparing our methods, the Bonferroni correction is so large that many comparisons that are significant in practice do not register as statistically significant. We therefore will also discuss clinical significance. Clinical significance will be assigned based on the significance of an individual test at the $\alpha = .05$ confidence level in the case of multiple comparisons.

4 Results

The experiments that were run imply that A_2, an approval voting scheme in which each attribution was able to vote equally for two authors, outperforms chance as well as all other experiments. The A_2 had a 43.38% in-bound accuracy and a 52.33% out-of-bounds accuracy when each problem set was tested as a group. It had a 38.78% in-bound accuracy

Table 1: Per-Document Accuracies

	In-Bound	Out-of-Bounds	Average	Combined
Random	17.09%	17.09%	17.09%	3.66%
$S(k=2)$	37.76%	58.16%	47.96%	18.37%
$S(k=3)$	48.98%	46.94%	47.96%	25.51%
R_0	50.00%	48.98%	49.49%	26.53%
R_A	60.20%	19.39%	39.80%	14.29%
$R_{A_{CI}}$	59.18%	18.37%	38.78%	13.27%
A_2	38.78%	69.39%	54.08%	23.47%
A_3	32.65%	69.39%	51.02%	23.47%
A_4	26.53%	68.37%	47.45%	17.35%
A_{ALL}	9.18%	86.73%	47.96%	5.10%

Table 2: Per-Set Accuracies

	In-Bound	Out-of-Bounds	Average	Combined
Random	22.08%	22.08%	22.08%	5.66%
$S(k=2)$	40.64%	40.75%	40.70%	14.70%
$S(k=3)$	50.32%	31.86%	41.09%	18.20%
R_0	49.58%	35.18%	42.38%	19.95%
R_A	55.75%	17.81%	36.78%	11.45%
$R_{A_{CI}}$	54.00%	18.11%	36.06%	10.15%
A_2	43.38%	52.33%	47.85%	23.54%
A_3	36.26%	53.82%	45.04%	22.54%
A_4	31.13%	48.97%	40.05%	16.35%
A_{ALL}	18.08%	70.51%	44.29%	8.46%

and a 69.39% out-of-bounds accuracy when each document was tested independently. Average per-set accuracy over the two scenarios was 47.85%, and average per-document accuracy was 54.08%. The combined per-set accuracy was 23.54%, and the combined per-document accuracy was 23.47%. These were all significantly better than random guessing would be.

The next best method indicated by the data would seem to be R_0, simple runoff voting in which obvious distractor authors are culled and the test is readministered using only the remaining authors. This method also outperforms chance, and in some cases even outperforms the top ranking indexed voting method. Using simple runoff voting had a 49.58% in-bound accuracy and a 35.18% out-of-bounds accuracy when each problem set was tested as a group. It had a 50.00% in-bound accuracy and a 48.98% out-of-bound accuracy when each document was tested independently. Average per-set accuracy over the two scenarios was 42.38%, and average per-document accuracy was 49.49%. The combined per-set accuracy was 19.95%, and the combined per-document accuracy was 26.53%.

5 Discussion

In this section we explain the interpretation of our experiments and discuss the measures used to determine the accuracy of open class attribution methods. We then examine these accuracies and compare them against random chance and against each other in an attempt to determine which method is the best. Finally, we examine numerous summary statistics for each method and isolate trends in the data to help support our conclusions.

5.1 Interpretation

After choosing an open class attribution method, performing an appropriate number of individual closed class attributions, and performing inferences between our authors at appropriately adjusted α levels, we must interpret the results. After each inference between p_{A_i} and p_{A_j}, we log whether or not the result was significant (i.e. whether or not $p < \alpha_{adj}$) and, if significant, the nature of that difference (i.e. positive or negative). A

significant positive difference registers as evidence for author i, whereas a significant negative difference is evidence for author j. There are essentially only two cases:

- Case 1: There exists an i such that the proportion of attributions in favor of A_i is significantly greater that those in favor of A_j for all $j \neq i$. In this case, we decide that the author A_i is indeed the true author of the document.
- Case 2: Otherwise, there are at least two authors who are "tied" (in the sense of statistical significance) for the highest proportion of attributions. In this case, we conclude that none of the authors is the true author, since the true author would have received a significantly higher proportion of attributions if s/he had been present in the candidate set.

In Case 2, it may be that the true differences in the proportions are simply not large enough to be identified with our initial sample size estimate. A larger value for k in equation (6) may be used in order to reduce the chances of a false Case 2.

5.2 Accuracy Measures

All methods were tested within the JGAAP framework on the AAAC corpus in both an in-bound and out-of-bounds (OOB) scenario. The in-bound scenario included all of the available training documents in the problem set; thus all the authors were possibilities for each individual closed class attribution; the out-of-bounds scenario removed the training documents written by the true author of the test document. This tested the open class attribution methods both in cases where the true author was among the suspects and in cases where the true author was not.

Note that due to the nature of the out-of-bounds testing procedure, all test documents in document sets with only two possible authors were incorrectly attributed in the OOB case; after removing the true author, there was only one possible author left for the closed class attribution methods to pick, which they inevitably did. This essentially turned an open class attribution problem into an authorship verification problem, which the program was

not designed to handle. The results are included since they do reflect a result given by the program, and future work may include a method to integrate verification functionality into this methodology.

Each scenario was evaluated individually in a per-set sense and a per-document sense. The per-set accuracy is computed by summing the straight accuracies from each set and dividing by 1300 (the maximum sum of the accuracies of the 13 sets). The per-document accuracy is computed by summing the total number of correct attributions overall and dividing by the total number of attempted attributions. In essence, the per-set accuracy identifies each set in the AAAC corpus as an independent entity, whereas the per-document accuracy sees each document as an independent entity. For each open class attribution method, the per-set accuracies were averaged and the per-document accuracies were averaged to get two average accuracies.

A final combined accuracy measure was used to determine the accuracy of the overall method using both the in-bound and out-of-bounds scenario. This measure counts only the documents where *both* in-bound and out-of-bounds attributions were correct as successful attributions. Consider an open class attribution method which answered ten out of ten in-bound attributions correctly, but only one of ten out-of-bounds attributions correctly. The combined accuracy measure would be 10%, since only in one out of ten tests did the method answer *both* the in-bound and out-of-bounds questions correctly. This measure is evaluated in both a per-set sense and a per-document sense in the same way as described above.

5.3 Accuracy Tests

Please note the following symbols used in tables displaying statistical test results:

- a double asterisk (**) denotes a value that is statistically significant in terms of the entire test's $\alpha = .05$ level, including any correction for multiple comparisons.
- a single asterisk (*) denotes a value that is clinically significant individually at the $\alpha = .05$ level.

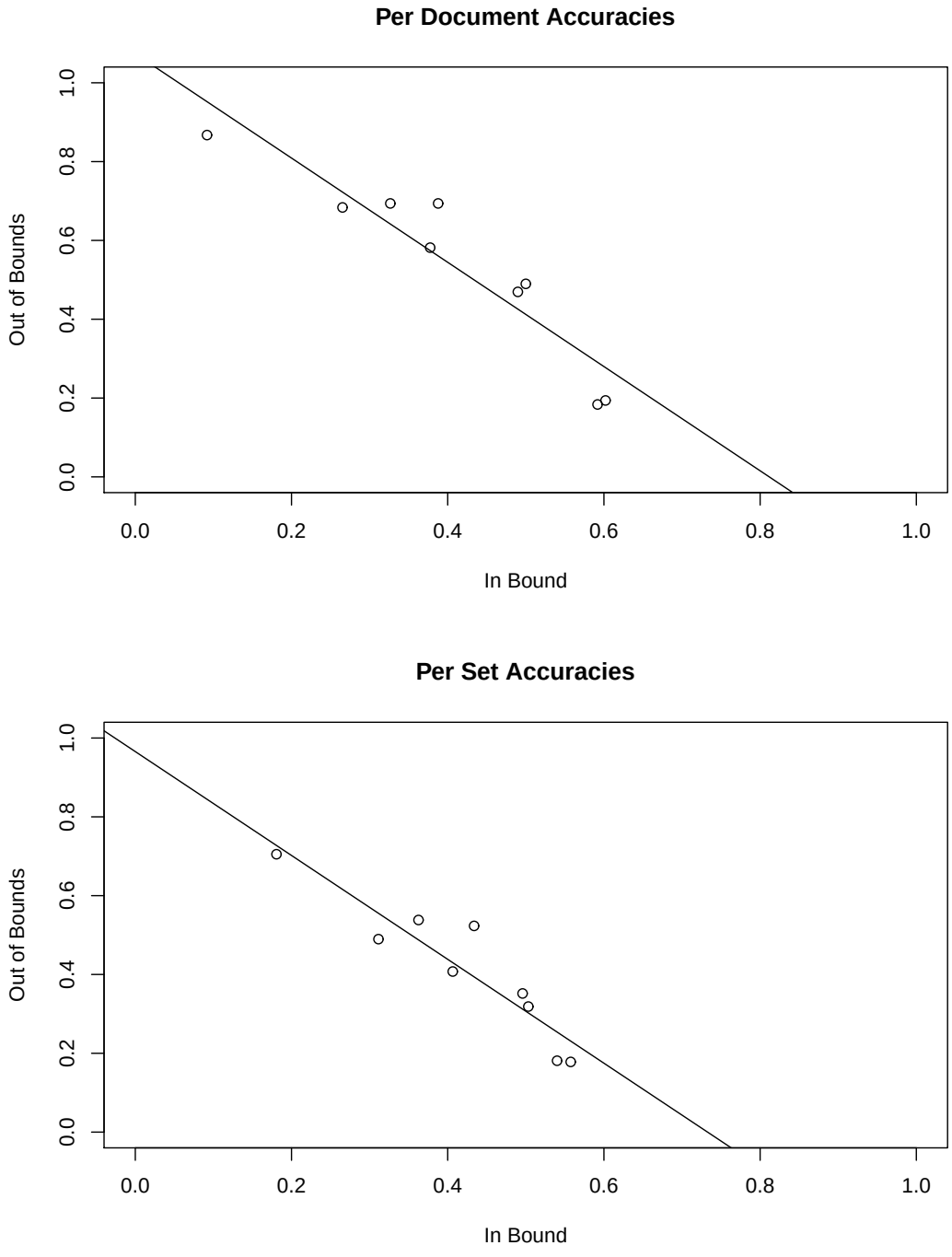

Figure 2: In-Bound and Out-of-Bounds Accuracies Across Open Class Attribution Methods

As can be seen in figure 2, accuracies in the in-bound and out-of-bounds cases are significantly negatively correlated; a method with a high in-bound accuracy measure usually had a low out-of-bounds accuracy measure and vice-versa. Per-document in-bound and out-of-bounds accuracy correlation was $r = -.9333$ with a least squares regression line of $\hat{y} = -1.3226x + 1.0735$ with $r^2 = .8711$. Per-set in-bound and out-of-bounds accuracy correlation was $r = -.9255$ with a least squares regression line of $\hat{y} = -1.3179x + 0.9656$ with $r^2 = .8566$. An open class attribution method with a significantly high in-bound accuracy measure could effectively be hamstrung by a significantly low out-of-bounds accuracy. Therefore, while we will examine all of our accuracy measures, we will pay special attention to the average and combined accuracies. These measures will each give a single value that will not be "made up for" in a different accuracy measure. For example, a method that simply responds "None of the above" in every attribution in our testing framework will have a 100% per-document out-of bounds accuracy but only a 3.06% per-document in-bound accuracy. However, the average per-document accuracy would be 51.53% and the combined per-document accuracy would be 3.06%. Thus, the average accuracies and the combined accuracies give a much more complete picture of how well any individual open class attribution method performed.

5.3.1 Tests Against Chance

Table 3: In-Bound Accuracy Tests Against Chance

	Per-Doc Accuracy	p-value	Per-Set Accuracy	p-value
Random	17.09%	-	22.08%	-
$S(k=2)$	37.76%	0.00059*	40.64%	0.00255*
$S(k=3)$	48.98%	<.00001**	50.32%	0.00002**
R_0	50.00%	<.00001**	49.58%	0.00003**
R_A	60.20%	<.00001**	55.75%	<.00001**
$R_{A_{CI}}$	59.18%	<.00001**	54.00%	<.00001**
A_2	38.78%	0.00036*	43.38%	0.00074*
A_3	32.65%	0.00586	36.26%	0.01448
A_4	26.53%	0.05477	31.13%	0.07575
A_{ALL}	9.18%	0.94933	18.08%	0.75788

Table 4: Out-of-Bounds Accuracy Tests Against Chance

	Per-Doc Accuracy	p-value	Per-Set Accuracy	p-value
Random	17.09%	-	22.08%	-
$S(k=2)$	58.16%	<.00001**	40.75%	0.00243*
$S(k=3)$	46.94%	<.00001**	31.86%	0.06149
R_0	48.98%	<.00001**	35.18%	0.02125
R_A	19.39%	0.33852	17.81%	0.77292
$R_{A_{CI}}$	18.37%	0.40744	18.11%	0.75583
A_2	69.39%	<.00001**	52.33%	<.00001**
A_3	69.39%	<.00001**	53.82%	<.00001**
A_4	68.37%	<.00001**	48.97%	0.00004**
A_{ALL}	86.73%	<.00001**	70.51%	<.00001**

Table 5: Average Accuracy Tests Against Chance

	Per-Doc Accuracy	p-value	Per-Set Accuracy	p-value
Random	17.09%	-	22.08%	-
$S(k=2)$	47.96%	<.00001**	40.70%	0.00004**
$S(k=3)$	47.96%	<.00001**	41.09%	0.00003**
R_0	49.49%	<.00001**	42.38%	<.00001**
R_A	39.80%	<.00001**	36.78%	0.0007*
$R_{A_{CI}}$	38.78%	<.00001**	36.06%	0.00115*
A_2	54.08%	<.00001**	47.85%	<.00001**
A_3	51.02%	<.00001**	45.04%	<.00001**
A_4	47.45%	<.00001**	40.05%	0.00006**
A_{ALL}	47.96%	<.00001**	44.29%	<.00001**

Table 6: Combined Accuracy Tests Against Chance

	Per-Doc Accuracy	p-value	Per-Set Accuracy	p-value
Random	3.66%	-	5.66%	-
$S(k=2)$	18.37%	0.0005*	14.70%	0.01817
$S(k=3)$	25.51%	<.00001**	18.20%	0.00338
R_0	26.53%	<.00001**	19.95%	0.00138*
R_A	14.29%	0.00463	11.45%	0.07377
$R_{A_{CI}}$	13.27%	0.00785	10.15%	0.12206
A_2	23.47%	0.00003**	23.54%	0.0002**
A_3	23.47%	0.00003**	22.54%	0.00034**
A_4	17.35%	0.00089*	16.35%	0.00841
A_{ALL}	5.10%	0.31094	8.46%	0.22197

As benchmarks, the theoretical accuracies were computed. The theoretical accuracies assume that each open class attribution is independent of all others and is a random guess with the answer chosen uniformly from the set of N authors as well as one "None of the above" possibility, creating essentially a closed set of $(N+1)$ equally likely possible choices. The theoretical probabilities for each accuracy measure can be found in Tables 3 through 6. Per-set and per-document average and combined accuracies were each compared to theoretical values using a z-test for two independent proportions. These four tests were performed simultaneously with accuracy data from all nine open class attribution method variations. With four tests on each of nine methods, there were 36 different comparison tests to perform. Thus, utilizing the same Bonferroni correction method used in the inference section above, we can arrive at a critical value for our z-test to interpret the significance of any improvements over random guessing.

Each of the methods tested had a significantly higher average per-document accuracy than random chance would allow; the least significant comparison had a z-score of 4.784 corresponding to $p < .0000009$. Only two attribution methods' accuracies were statistically significantly higher than chance in all three of the remaining accuracy measures. Both of these attribution methods were indexed approval voting methods using a small number of equal votes. There were also two methods that did *not* score statistically significantly higher than chance in *any* of the other three accuracy measures. Both of these were instances of runoff voting with adjusted k values.

5.3.2 Comparison of Methods

We can also compare these open class attribution methods to each other. There are nine open class attribution methods each with four overall accuracies we would like to test. Thus there are ${}_9C_2 = 36$ ways to choose two methods to compare and 4 accuracies to compare for each pairing, giving a total of $36 * 4 = 144$ different comparisons to be made. The Bonferroni correction inflates our α_{adj} value for each comparison to .000174, a very low value indeed. Even with this small value for α, four of the 144 individual tests registered as statistically significant. The comparisons of $S(k = 3)$, R_0, A_2 and A_3 all had combined

Table 7: Per-Document Average Accuracy - Method Comparison P-Values

		$S(k=2)$	$S(k=3)$	R_0	R_A	$R_{A_{CI}}$	A_2	A_3	A_4	A_{ALL}
		47.96%	47.96%	49.49%	39.80%	38.78%	54.08%	51.02%	47.45%	47.96%
$S(k=2)$	47.96%									
$S(k=3)$	47.96%	0.5								
R_0	49.49%	0.38089	0.38089							
R_A	39.80%	0.05171	0.05171	0.02678						
$R_{A_{CI}}$	38.78%	0.03329	0.03329	0.01634	0.41807					
A_2	54.08%	0.11267	0.11267	0.18149	0.0023	0.00119				
A_3	51.02%	0.27222	0.27222	0.38093	0.01282	0.0074	0.27196			
A_4	47.45%	0.45973	0.45973	0.34301	0.06329	0.04148	0.09453	0.23972		
A_{ALL}	47.96%	0.5	0.5	0.38089	0.05171	0.03329	0.11267	0.27222	0.45973	

Table 8: Per-Set Average Accuracy - Method Comparison P-Values

		$S(k=2)$	$S(k=3)$	R_0	R_A	$R_{A_{CI}}$	A_2	A_3	A_4	A_{ALL}
		40.70%	41.09%	42.38%	36.78%	36.06%	47.85%	45.04%	40.05%	44.29%
$S(k=2)$	40.70%									
$S(k=3)$	41.09%	0.46834								
R_0	42.38%	0.36751	0.39781							
R_A	36.78%	0.21289	0.19057	0.1283						
$R_{A_{CI}}$	36.06%	0.17249	0.15299	0.09988	0.44113					
A_2	47.85%	0.07689	0.08898	0.13818	0.01323	0.00898				
A_3	45.04%	0.19234	0.21478	0.29768	0.04804	0.03502	0.28846			
A_4	40.05%	0.44809	0.41686	0.31957	0.25269	0.20781	0.05981	0.15874		
A_{ALL}	44.29%	0.2356	0.26072	0.35115	0.06479	0.04813	0.23987	0.44081	0.19741	

Table 9: Per-Document Combined Accuracy - Method Comparison P-Values

		$S(k=2)$	$S(k=3)$	R_0	R_A	$R_{A_{CI}}$	A_2	A_3	A_4	A_{ALL}
		18.37%	25.51%	26.53%	14.29%	13.27%	23.47%	23.47%	17.35%	5.10%
$S(k=2)$	18.37%									
$S(k=3)$	25.51%	0.11348								
R_0	26.53%	0.08542	0.43534							
R_A	14.29%	0.21976	0.02453	0.01672						
$R_{A_{CI}}$	13.27%	0.16385	0.01507	0.01001	0.41791					
A_2	23.47%	0.18995	0.36987	0.31035	0.05022	0.03254				
A_3	23.47%	0.18995	0.36987	0.31035	0.05022	0.03254	0.5			
A_4	17.35%	0.42603	0.08187	0.06016	0.27852	0.21373	0.14381	0.14381		
A_{ALL}	5.10%	0.00196	0.00004**	0.00002**	0.0149	0.02393	0.00012**	0.00012**	0.00331	

Table 10: Per-Set Combined Accuracy - Method Comparison P-Values

		$S(k=2)$	$S(k=3)$	R_0	R_A	$R_{A_{CI}}$	A_2	A_3	A_4	A_{ALL}
		14.70%	18.20%	19.95%	11.45%	10.15%	23.54%	22.54%	16.35%	8.46%
$S(k=2)$	14.70%									
$S(k=3)$	18.20%	0.2545								
R_0	19.95%	0.16607	0.3779							
R_A	11.45%	0.24949	0.09168	0.05097						
$R_{A_{CI}}$	10.15%	0.16699	0.05308	0.02756	0.38495					
A_2	23.54%	0.05781	0.17875	0.27084	0.01292	0.00613				
A_3	22.54%	0.07926	0.22512	0.32827	0.0193	0.00948	0.43414			
A_4	16.35%	0.37523	0.36579	0.25672	0.16064	0.10031	0.10378	0.13655		
A_{ALL}	8.46%	0.0861	0.02245	0.01064	0.24262	0.3421	0.00199	0.00323	0.047	

per-document accuracies significantly higher than A_{ALL} with 95% confidence. This certainly establishes at least a partial ordering of the nine methods tested in that A_{ALL} is certainly less effective than the four methods whose accuracies were significantly higher. It is also worth noting that the $S(k = 2)$ was not statistically significantly higher than A_{ALL} but $S(k = 3)$ was. This is at least experimental evidence that the k parameter does indeed increase that accuracy of a open class attribution method, at least to some extent.

Even if we cannot claim statistically significant differences between many of the accuracies, we can still take note of some characteristic similarities and differences among the methods tested. The maximum accuracies in three of the four accuracy categories highlight an approval voting method with two equal votes as the best method of the experiment. However, the combined per document accuracy was greatest for runoff voting with no adjustments. Thus, while we cannot make a statistical statement about whether one method is better than the other, it seems safe to claim that these two methods do very well. This is upheld by the statistically significant comparison above: both these methods are listed therein.

Overall, there were three general methods of open class attribution tested: single voting, approval voting, and runoff voting. General statements can be made within some of these broad categories. Approval voting was most effective with a small number of equal votes. As the number of equal votes increased, the in-bound and out-of-bounds accuracies tended to decrease. The extreme case of approval voting using normalized proportional distances performed a great deal better in the out-of-bounds case but much worse in the in-bound case. Runoff voting seems to do best in the simplest case. Any adjustment of the k value in response to previous rounds increased the in-bound accuracies while drastically decreasing out-of-bounds accuracies. The net result of these differences was to greatly lower average and combined accuracies.

Each of the open class attribution methods tested were developed on top of a method of inference with a parameter that would (theoretically) enable a tester to arbitrarily increase the accuracy of a method. This parameter, k, was a measure of the acceptable difference in vote proportions for candidate authors that could be called significant. This was

accomplished by increasing the number of closed class attributions required for the test, also parameterized by k. Two tests of single voting were run: one used a k value of two, the other used a k value of three. The evidence supports the theory that an increased k value will increase accuracy. While an increase from $k = 2$ to $k = 3$ was not sufficient to improve the accuracy significantly in a statistical sense, the accuracies for this experiment did indeed increase (or stayed constant) across all four accuracies involving both in-bound and out-of-bounds tests. Note, however, that an increased k value, while increasing in-bound test accuracy, decreased out-of-bound test accuracy even though average accuracy and combined accuracy increased.

5.3.3 Confusion Matrices

The results of each method were also transformed into a 2x2 confusion matrix, as in Table 11. Various summary statistics were calculated, as outlined below:

Table 11: Confusion Matrix for A_2

	Open Class Attribution Method Action		
	Assigns author	Assigns NOTA	TOTAL
Correct	37	69	106
Incorrect	38	52	90
TOTAL	75	121	

- Sensitivity - $\frac{CAA}{95}$; measures the percent of author attributions made correctly out of the number that *should* have been made.
- Specificity - $\frac{CNOTA}{101}$; measures the percent of "None of the Above" attributions made correctly out of the number that *should* have been made.
- AUC - area under the receiver-operator characteristic curve created using precision (PPV) and recall (sensitivity).
- F-score - harmonic mean of precision (PPV) and recall (sensitivity)
- Positive Predictive Value (PPV) - $\frac{CAA}{TAA}$; measures the percent of author attributions made correctly out of the number that *were* made.

- Negative Predictive Value (NPV) - $\frac{CNOTA}{TNOTA}$; measures the percent of "None of the Above" attributions made correctly out of the number that *were* made.

The summary statistics for each method are given in the Table 12.

Table 12: Summary Statistics from Confusion Matrices

	Sensitivity	Specificity	AUC	F-score	PPV	NPV
$S(k=2)$	0.3789	0.5743	0.5915	0.3850	0.3913	0.5577
$S(k=3)$	0.5053	0.4554	0.5387	0.4324	0.3780	0.6667
R_0	0.5158	0.4752	0.5300	0.4601	0.4153	0.6154
R_A	0.6211	0.1881	0.4153	0.4538	0.3576	0.6129
$R_{A_{CI}}$	0.6000	0.1881	0.4192	0.4419	0.3497	0.5758
A_2	0.3895	0.6832	0.5949	0.4353	0.4933	0.5702
A_3	0.3158	0.6931	0.6122	0.3774	0.4688	0.5303
A_4	0.2421	0.6931	0.6516	0.2987	0.3898	0.5109
A_{ALL}	0.0632	0.8713	0.7928	0.1034	0.2857	0.5029

Table 13: Ranks of Accuracy Measures from Confusion Matrices

	Sensitivity	Specificity	AUC	F-score	PPV	NPV	Sum of Ranks
$S(k=2)$	6	5	5	6	4	6	32
$S(k=3)$	4	7	6	5	6	1	29
R_0	3	6	7	1	3	2	22
R_A	1	8	9	2	7	3	30
$R_{A_{CI}}$	2	8	8	3	8	4	33
A_2	5	4	4	4	1	5	23
A_3	7	2	3	7	2	7	28
A_4	8	2	2	8	5	8	33
A_{ALL}	9	1	1	9	9	9	38

The methods were examined using a completely ranked system for each extra accuracy measure. Each open class attribution was ranked 1 (best) through 9 (worst) in each extra accuracy measure, resulting in Table 13. A non-parametric Wilcoxon Rank-Sum test was used to compare the methods based on these rankings; none of the tests returned statistically significant values. The critical value for $\alpha = .05$ two-tailed test in the Wilcoxon Rank-Sum test with N = 6 is 21; the largest ranked sum we can achieve with our data is 15.

Examination of Tables 12 and 13 does uphold the conclusions reached by comparing the pure accuracies. The rankings for each attribution method were added together to arrive at one number to associate with each method. These sums of ranks could then be compared to one another, albeit not in a statistical way; the lower the sum of ranks, the better the method (NOTE: these sums of ranks are NOT the values used in the Wilcoxon Rank-Sum test). Our conclusions concerning the best methods are upheld. Both the approval voting method with two equal votes, A_2, and runoff voting with no adjustments, R_0, scored very well in terms of the sum of ranks. Note that A_2 does not rank any worse than 5 in any accuracy measure. However, the worst method in terms of the extra accuracy measures is undoubtedly the approval voting using normalized proportional distances, A_{ALL}. It ranks worst (9) in four out of six measures.

6 Conclusions and Future Work

We have tested nine methods of combining multiple closed class authorship attribution methods to answer one open class attribution problem. Out of nine candidate methods built from three voting principles, all performed better than chance in at least one accuracy measure utilized. One method, an approval voting scheme where each vote is equally split between two different training documents, outperformed all others in three out of four accuracy measures specifically designed to include information from both in-bound (author present) and out-of-bounds (author not present) scenarios. Thus, the mixture-of-experts approach to solving the open class attribution problem looks promising. Many of the comparisons of the techniques used in this experiment were inconclusive mainly due to the relatively large number of comparisons to perform. Comparing every combination of two of nine methods yielded 36 different comparisons. This large number of comparisons drove the necessary confidence level for each test so high that very few accuracies were different enough to make any claims. Had only two methods been tested there would have been only one test with a significantly lower critical value for the same level of confidence. Thus, further research may continue testing and analyzing the methods

tested here in smaller scope: perhaps only one method built from each voting principle could be tested instead of many.

Further research into the accuracies and types of methods used to build a mixture-of-experts attribution may allow for significant increases in these accuracies. For example, determining assorted weights to associate with various closed class attribution methods would enable a vote from a more reliable method to add a larger value to the proportion for the winning author. Another similar alternative would be to find a distribution of such weights for each individual method, and a random value from the proper distribution could be chosen each time the method was used.

All of the open class attribution methods tested here were created using a large number of closed class attributions. Increasing the number of available closed class attribution methods would allow for larger k values to be used in sample size calculations, thus allowing for decreased standard error values in the inference portion of the analysis. Adding functionality for non-distance based analysis methods would serve this purpose as well as adding information from a "blind spot" in this analysis. Any new event sets would also create new methods to be used and new information to add to an open class attribution.

References

[1] Analysis of two dependent proportions. http://www.docstoc.com/docs/63760038/Analysis-of-Two-Dependent-Proportions. Accessed June 12, 2012.

[2] Abebe, A., Daniels, J., McKean, J. W., and Kapenga, J. A. Statistics and data analysis. http://www.stat.wmich.edu/s160/book/book.html. Accessed September 30, 2012.

[3] Grieve, J. Quantitative authorship attribution: An evaluation of techniques. *Literary and Linguistic Computing 22*, 3 (2007), 251–270.

[4] Hughes, V. How forensic linguistics outed j.k rowling (not to mention james madison, barack obama, and the rest of us. *National Geographic* (2013).

[5] Juola, P. Authorship attribution. *Foundations and Trends in Information Retrieval 1*, 3 (2008), 233–334.

[6] Juola, P., Sofko, J., and Brennan, P. A prototype for authorship attribution studies. http://www.mathcs.duq.edu/ juola/papers.d/jsb-aut.pdf. Accessed June 12, 2012.

[7] Koppel, M., Akiva, N., Dershowitz, I., and Dershowitz, N. Unsupervised decomposition of a document into authorial components. http://www.dershowitz.net/files/unsupervised-decomposition-of-a-document-into-authorial-components.pdf. Accessed June 13, 2012.

[8] Koppel, M., Schler, J., and Argamon, S. Computational methods in authorship attribution. *JASIST 60*, 1 (2009), 9–26.

[9] Koppel, M., Schler, J., and Argamon, S. Authorship attribution in the wild. *Language Resources and Evaluation 45*, 1 (2011), 83–94.

[10] Koppel, M., Schler, J., Argamon, S., and Winter, Y. The "fundamental problem" with authorship attribution. *English Studies 93*, 3 (2012), 284–291.

[11] Koppel, M., Schler, J., and Messeri, E. Authorship attribution in law enforcement scenarios. http://u.cs.biu.ac.il/ koppel/papers/authorship-NATO-bookchapter-final_2_.pdf. Accessed June 13, 2012.

[12] Luyckx, K., and Daelemans, W. The effect of author set size and data size in authorship attribution. *Literary and Linguistic Computing 26*, 1 (2010), 35–55.

[13] Malyutov, M. Authorship attribution of texts: a review. http://www.math.neu.edu/ malioutov/marlowe7.pdf, 2006. Accessed June 13, 2012.

[14] Malyutov, M. B., Wickramasinghe, C. I., and Li, S. Conditional complexity of compression for authorship attribution. http://edoc.hu-berlin.de/series/sfb-649-papers/2007-57/PDF/57.pdf, 2007. Accessed June 13, 2012.

[15] Mosteller, F., and Wallace, D. L. *Inference and Disputed Authorship: The Federalist*. Addison-Wesley, 1964.

[16] Rosenblum, N., Zhu, X., and Miller, B. P. Who wrote this code? identifying the authors of program binaries. ftp://ftp.cs.wisc.edu/paradyn/papers/Rosenblum11Authorship.pdf. Accessed June 17, 2012.

[17] Silva, R. S., Laboreiro, G., Sarmento, L., Grant, T., Oliveira, E., and Maia, B. 'twazn me! ;(' automatic authorship analysis of micro-blogging messages. *Natural Language Processing and Information Systems* (2011), 161–168.

[18] Stamatamos, E. A survey of modern authorship attribution methods. http://www.icsd.aegean.gr/lecturers/stamatatos/papers/survey.pdf. Accessed June 15, 2012.

[19] Stein, B., Koppel, M., and Stamatatos, E. Plagiarism analysis, authorship identification, and near-duplicate detection. *ACM SIGIR 41*, 2 (2007), 68–71.

[20] Tresp, V. Committee machines. http://www.brauer.informatik.tu-muenchen.de/ trespvol/papers/combine_incl_proof.pdf, 2001. Accessed June 16, 2012.

[21] Zhao, Y., and Zobel, J. Searching with style: Authorship attribution in classic literature. http://ww2.cs.mu.oz.au/ jz/fulltext/acsc07zz.pdf. Accessed June 13, 2012.

A Derivation of Equation 1

The standard error cited in [1] contains the equation:

$$SE(p_d) = \frac{1}{n}\sqrt{b + c - \frac{(b-c)^2}{n}}$$

where b and c are the number of "votes" for B and C respectively, and n is the total number of "votes." With a little algebra, we can convert this into a form that is usable with proportions such as the ones used in this experiment. First, pull the term $\frac{1}{n}$ inside the square root and distribute.

$$SE(p_d) = \sqrt{\frac{1}{n^2}(b + c - \frac{(b-c)^2}{n})}$$

$$SE(p_d) = \sqrt{\frac{1}{n}(\frac{b}{n} + \frac{c}{n} - \frac{(b-c)^2}{n^2})}$$

$$SE(p_d) = \sqrt{\frac{1}{n}(\frac{b}{n} + \frac{c}{n} - \frac{(b-c)}{n}\frac{(b-c)}{n})}$$

$$SE(p_d) = \sqrt{\frac{1}{n}(\frac{b}{n} + \frac{c}{n} - (\frac{b}{n} - \frac{c}{n})(\frac{b}{n} - \frac{c}{n}))}$$

Note that $\frac{b}{n}$ is simply the proportion of votes for B, $\hat{p}_B$; similarly, $\frac{c}{n}$ is the proportion of votes for C, $\hat{p}_C$. Thus:

$$SE(p_d) = \sqrt{\frac{1}{n}(\hat{p}_B + \hat{p}_C - (\hat{p}_B - \hat{p}_C)(\hat{p}_B - \hat{p}_C))}$$

$$SE(p_d) = \sqrt{\frac{1}{n}(\hat{p}_B + \hat{p}_C - (\hat{p}_B - \hat{p}_C)^2)}$$

$$SE(p_d) = \sqrt{\frac{\hat{p}_B + \hat{p}_C - (\hat{p}_B - \hat{p}_C)^2}{n}}$$

Changing the subscripts (which are just dummy variables), this is the equation used in this paper for calculating the standard error between to dependent proportions. The equation is also used in [2] concerning election margins of error and confidence intervals.

A JGAAP Canonicizers

- Normalize ASCII

  Normalizes the document to printable ASCII, removing non-ASCII characters such as ♣, Kanji characters, etc....

- Normalize Whitespace

  Changes length of all white spaces to 1. Any sequence of whitespaces including newline, tab, and space, will become a single space in the processed document.

- Punctuation Separator

  If any punctuation (defined as a non-word non-whitespace character) is next to any non-whitespace character, adds a space between them.

- Strip AlphaNumeric

  Strips all non-punctuation from the document.

- Strip Null Characters

  Strips all null characters from the document.

- Strip Numbers

  Replace all numbers with "0".

- Strip Punctuation

  Strips any punctuation from the document.

- Unify Case

  Converts all characters in the document into lower case using Character.toLowerCase().

B JGAAP Event Sets

- 2–3 Letter Words

  This event set is all words with $2 \leq \text{length} \leq 3$.

- 2–4 Letter Words

  This event set is all words with $2 \leq \text{length} \leq 4$.

- 3–4 Letter Words

  This event set is all words with $3 \leq \text{length} \leq 4$.

- Appending Multiple EventDrivers

  Appends two or more underlying EventSets (parameterized as underlying events, a comma-separated list of EventDrivers) into one EventSet.

- Binned Frequencies

  Discretized (by truncation) ELP lexical frequencies. Default truncation length = 3.

- Binned naming times

  Discretized (by truncation) ELP naming latencies. Default truncation length = 2.

- Binned Reaction Times

  Discretized (by truncation) ELP lexical decision latencies. Default truncation length = 2.

- Black-List

  Filters all Event strings against named file and removes named events. Compare to WhiteListEventSet, which removes all BUT named events

- Character BiGrams

  Extracts consecutive groups of 2 character as features.

- Character NGrams

  Extracts consecutive groups of N characters as features with default N = 2.

- Character TetraGrams

  Extracts consecutive groups of 4 character as features.

- Character TriGrams

  Extracts consecutive groups of 3 character as features.

- Characters

  This event set is all individual characters, as determined by the preprocessing applied in the previous stage.

- Dis Legomena

  Finds words that are used exactly twice in a document.

- Coarse POS Tagger

  A simplification of the normal part of speech tagger, neutralizing minor variations such as plural inflection; for example, all noun types (proper/common, singular/plural) are grouped.

- Generic Event N-gram

  This event set is N-grams (parameterized as N) of an underlying event model (parameterized as underlyingevents). Default value of N = 2 and default underlyingevents is Words.

- Generic Tumbling Event N-Gram

  This event set is N-grams (parameterized as N) of an underlying event model (parameterized as underlyingevents). It differs from NGramEventDriver in that it is a 'tumbling' window instead of a 'sliding' one; if the current event is ABCDEF, the next one will not start with B but may start with C, D, or even the next symbol after F. The amount of tumbling is set by the 'tumbleLength' parameter. Default value of N = 2, default underlyingevents is Words, and default tumbleLength = 2.

- Hapax Legomena

  Finds words that are used exactly once in a document.

- Hapax/Dis Legomena

  Finds words that are used exactly once or twice in a document.

- Lexical Decision Reaction Times

  Reaction times taken from English Lexicon Project. Converts each word (using table) to the time it takes to perform lexical decision on that word in the ELP database. Obviously English-only, and obviously incomplete; words that are not in the database are silently removed.

- Lexical Frequencies

  Corpus frequencies taken from English Lexicon Project. Converts each word (using table) to the (log-scaled) frequency in which that word appears in the general purpose HAL corpus as recorded in the ELP database. Obviously English-only, and obviously incomplete; words that are not in the database are silently removed.

- M–N letter words

  This event set is all “words" (NaiveWordEventDriver) with M $\leq$ length $\leq$ N (M and N being parameters “M" and “N" respectively). M and N default to 2 and 3 respectively.

- MW Function Words

  Uses function words as defined by Mosteller-Wallace in their Federalist papers study.

- Naming Reaction Times

  Naming times taken from English Lexicon Project. Converts each word (using table) to the time it takes to name that word in the ELP database. Obviously English-only, and obviously incomplete; words that are not in the database are silently removed.

- Numeric Transformation Events

  Transforms Event strings for other Event Strings in generation. Creates an EventSet using an underlying EventDriver, then reads in a file containing /from/to/ substitutions pairs. Can be used, for example, for normalization, stemming, and so

forth. Default underlying EventDriver = Words and default substitution file = null (no substitutions).

- POS

  Extracts the distribution of parts of speech in the document.

- POS BiGrams

  Extracts consecutive sequences of 2 parts of speech as features.

- POS DecaGrams

  Extracts consecutive sequences of 10 parts of speech as features.

- POS DodecaGrams

  Extracts consecutive sequences of 12 parts of speech as features.

- POS EnneaGrams

  Extracts consecutive sequences of 9 parts of speech as features.

- POS HendecaGrams

  Extracts consecutive sequences of 11 parts of speech as features.

- POS HeptaGrams

  Extracts consecutive sequences of 7 parts of speech as features.

- POS HexaGrams

  Extracts consecutive sequences of 6 parts of speech as features.

- POS NGrams

  Extracts consecutive sequences of N parts of speech as features. Default value of N = 2.

- POS OctaGrams

  Extracts consecutive sequences of 8 parts of speech as features.

- POS PentaGrams

  Extracts consecutive sequences of 5 parts of speech as features.

- POS TetraGrams

  Extracts consecutive sequences of 4 parts of speech as features.

- POS TriGrams

  Extracts consecutive sequences of 3 parts of speech as features.

- POS TriskaidecaGrams

  Extracts consecutive sequences of 13 parts of speech as features.

- Rare Words

  This event set is all events occurring only once of an underlying event model * (parameterized as underlyingevents)

- Sentence Length

  Extracts the number of words in each sentence as features.

- Suffices

  Calculates N (parameter) character suffix of Events, useful for extracting English suffixes like "-tion" or "-er" or "-est." Of course, it also works on other languages.

- Syllable Transitions

  Extracts syllable bigrams as features. (Suggested by Richard Forsyth, David I Holmes, and Emily K Tse, in 1998 tech report "Cicero, Sigonio and Burrows: Investigating the Authenticity of the 'Consolatio' ")

- Syllables Per Word

  This event set is the number of syllables in a given word, defined (naively) by the number of vowel clusters. This will not work well for words like "react" or "safes," but should be a decent approximation.

- Transformation Events

  Transforms Event strings for other Event Strings in generation. Creates an EventSet using an underlying EventDriver, then reads in a file containing /from/to/ substitutions pairs. Can be used, for example, for normalization, stemming, and so

forth. Default underlying EventDriver = Words and default substitution file = null (no substitutions).

- Truncated Events

  Truncates Events to shorter strings – i.e. "hello" becomes "he" Useful for binning NumericEventSets among other things Default EventSet = Words and default truncation length = 2.

- Vowel 2–3 letter Words

  Extract vowel-initial words with between 2 and 3 letters as features

- Vowel 2–4 letter Words

  Extract vowel-initial words with between 2 and 4 letters as features

- Vowel 3–4 letter Words

  Extract vowel-initial words with between 3 and 4 letters as features

- Vowel M–N letter Words

  Extract vowel-initial words with between 2 and 3 letters as features

- Vowel-initial words

  This event set is all "words" (NaiveWordEventDriver) beginning with vowels "aeiouAEIOU"; extension may be necessary to include non-English vowels or characters with diacritical marks like Danish ædigraph or German ö

- White-List

  Filters all Event strings against named file and removes unlisted events. Compare to BlackListEventSet, which removes listed events

- Word BiGrams

  Extracts all 2 word sequences as features.

- Word Length

  Extract number of characters in each word as features.

- Word NGrams

  Extracts all sequences of N words as features, with default N = 2.

- Word Stems

  Applies Porter's stemming algorithm (the Porter Stemmer) to produce just the stems of the underlying words or word sets. Stems, e.g. : farms, farmed, farming should all become just "farm". Porter's algorithm is freely licensed for all uses.

- Word Stems w/ Irregular

  Uses word stems with an expanded list of stem, such as "was" to "be" or "geese" to "goose"

- Word TetraGrams

  Extracts all 4 word sequences as features.

- Word TriGrams

  Extracts all 3 word sequences as features.

- Words

  Extract whitespace-separated words (including punctuation) as features.

C JGAAP Analysis Methods

- Canberra Distance

  Canberra distance, defined as $D(x,y) = \sum |\frac{(x_i - y_i)}{(x_i + y_i)}|$. This is a distance for Nearest Neighbor algorithms, based on (Wilson & Martinez 1997, JAIR).

- Cosine Distance

  Cosine Distance or normalized dot product. This is another distance for Nearest Neighbor algorithms. Defined as $D(x,y) = |\frac{\sum x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}} - 1|$.

- Cross Entropy Divergence

  Cross-entropy "divergence" for Nearest Neighbor. Calculated by $\sum (x_i)(-\log(y_i))$

- Histogram Distance

  Histogram distance using L_2 metric, defined as $D(x,y) = \sum (x_i - y_i)^2$. This is yet another distance for Nearest Neighbor algorithms

- Intersection Distance

  Given two event type-sets, A,B, calculates $1 - ||A \cap B||/||A \cup B||$.

- KeseljWeighted Distance

  Histogram distance as weighted by Keselj (2003). N.b. this was the AAAC 2004 winner when used with "common N-grams". Defined as $D(x,y) = \sum \frac{(x_i - y_i)^2}{(x_i + y_i)^2}$

- Kullback Leibler Distance

  Kullback-Leibler divergence, to be treated as yet another distance for nearest-neighbor algorithms. This is technically a divergence instead of a "distance" as it is noncommutative. Defined as $D(x||y) = \sum \log(\frac{x_i}{y_i})x_i$

- LZW Divergence

  LZWDistance, : conditional LZW distance, basically $LZW(ab) - LZW(b)$. This is yet another distance for Nearest Neighbor algorithms.

- Manhattan Distance

Histogram distance using L_1 metric, defined as $D(x, y) = \sum |x_i - y_i|$. This is yet another distance for Nearest Neighbor algorithms

- Nominal KS Distance

  Nominal Kolmogorov-Smirnov distance for Nearest Neighbor algorithm. Defined as $D(x, y) = \frac{\sum |x_i - y_i|}{2}$