UNDERGRADUATE THESIS

Stylometric Features for Multiple Authorship Attribution

Author:
Brian Yu

Presented to the Department of Computer Science and to the Department of Linguistics in partial fulfillment of the requirements for an A.B. degree with Honors

Harvard College

March 2019

Abstract

In computational linguistics, authorship attribution is the task of predicting the author of a document of unknown authorship. This task is generally performed via the analysis of stylometric features — particular characteristics of an author's writing that can be used to identify his or her works in contrast with the works of other authors. A wide variety of different features have been proposed in the literature for this classification task, many of which are based on the analysis of lexical, syntactic, or semantic properties of a text. I propose an extension to existing authorship attribution models aimed at solving the related problem of multiple author attribution, which attempts to perform authorship classification on documents that may be jointly written by multiple authors instead of one, and aims to predict sentence-level and section-level authorship within the document. To do so, I propose a model that first uses a sentence-level Bayesian classifier to predict the most likely author of each sentence in the composite document, and then uses those sentence-level predictions to estimate likely section boundaries between authors. The model is tested against a set of synthesized multi-author documents generated from a corpus of sentences of known authorship from literature. A Hidden Markov Model-based procedure is proposed for estimating section boundaries, and new possible syntax-based feature sets for classifier training — including function word embedded subtrees and a variant of syntactic n-grams — are proposed and demonstrated to improve predictive accuracy when solving multi-author attribution problems.

Acknowledgements

While the authorship of this thesis is attributed to me, there are many people for whom I am extremely grateful, and without whom this thesis would not have been possible.

First, I'd like to thank Professor Stuart Shieber, for advising me throughout this process for more than a year, for consistently providing thoughtful insights and suggestions, and for challenging me to continue to ask and answer new questions as they arose throughout the research process. Additionally, I'd like to thank Professor Kevin Ryan, for advising me during the beginning of this research project, and for helping me to formulate and articulate the problem that I wanted to tackle.

I also owe great thanks to Yonatan Belnikov for assisting me with some of the algorithmic design for this project, to Cristina Aggazzotti for advising me on computational linguistics during the early stages of this project, to Zuzanna Fuchs for inspiring me to pursue linguistics, and to Julia Sturm and Diti Bhadra for their feedback and advice on research in linguistics.

I'd also like to thank Professor David Malan and Professor Stephen Chong, for being incredibly supportive of me and my interests throughout my time as an undergraduate in the computer science department.

Outside of the computer science and linguistics departments, I'd like to thank Professor Leah Price, for her valuable knowledge and insight about the Brontë family and their work, and Professor Michael McCormick, for the assistance he provided to me in exploring the history surrounding issues of multiple or disputed authorship.

Additionally, I'd like to thank Samantha Berman, for being an inspiration and for always asking me thoughtful questions, Francesca Cornero, for being a source of encouragement and motivation, Amy Morrisett, for your kindness towards me through every stage of this process, Athena Braun, for your dependeability and for all the times we worked and struggled together, Truelian Lee, for encouraging me and believing in me since I started working on this project, and all of my other friends who have been incredibly kind and supportive as I've worked on this thesis.

Finally, I'd like to express my deepest gratitude to my parents, for their encouragement and support since the very beginning.

Contents

Chapter 1

Introduction

Between October 1787 and April 1788, Alexander Hamilton, James Madison, and John Jay published The Federalist Papers, a series of 85 documents — originally disseminated anonymously — supporting the ratification of the United States Constitution. Two days prior to his death in 1804, Hamilton compiled a list of the Federalist Papers that he claimed to have written, and provided the list to his lawyer (Adair 1944). The trouble was that this list directly contradicted a list that Madison would publish in 1817, in which Madison claimed to have written twelve of the papers that Hamilton had asserted were his own.

These twelve disputed Federalist Papers became an early notable instance of an authorship attribution problem, for which scholars could not conclusively agree on which of the disputed papers were written by Hamilton, and which were written by Madison. In the mid-20th century, Mosteller and Wallace (1963) approached the problem of attributing authorship to the twelve disputed papers by means of statistical inference, by studying the distribution of a particularly chosen set of 165 words across the disputed documents. By examining differences between the two writers in the frequency of their usage of words from the curated set, Mosteller and Wallace were able to draw conclusions about the general writing tendencies of Hamilton and Madison. Hamilton, for instance, used the word "upon" approximately 18 times more frequently than Madison did. By comparing the distribution of these words in the disputed documents with the distribution of these words in the known documents, Mosteller and Wallace concluded that the twelve disputed Federalist Papers were written by Madison rather than Hamilton.

While advances in statistics, machine learning, and stylometry — the statistical analysis of linguistic style — have improved the effectiveness of authorship attribution methods over time and have allowed such techniques to become more generalizable, it remains an open question as to how best to design a system for the analysis of disputed

documents to enhance the accuracy, performance, and robustness of modern authorship attribution systems.

I aim to explore a particular extension of the authorship attribution problem — namely, the problem of multiple authorship attribution: determining authorship for a document when the document in question may be written not just by a single author, but by multiple individuals in a set of candidate authors.

1.1 Stylometric Features for Authorship Discrimination

Central to the concept of authorship discrimination is the premise that the same idea, expressed by different individuals, can be articulated in distinct ways. Consider the following three sentences:

(1) Even though it was raining this morning, I still went running anyways.

(2) I ran this morning despite the rain.

(3) So I went for a run this morning, but it was raining.

These sentences express the same idea and convey the same core information, but vary in how their meaning is articulated: the sentences differ in length, use of punctuation, sentence structure, verb tense choice, and use of vocabulary, among other potential features. Thus, in addition to the semantic content of the sentence, each author's articulation of the sentence conveys additional information — information about the author's linguistic style. Authorship attribution has its basis in the hypothesis that individuals have natural inclinations towards particular characteristics of style. Those characteristics can then be used as a discriminator to distinguish them from other authors.

Since Mosteller and Wallace's work analyzing the Federalist Papers, additional techniques have emerged for determining authorship of an anonymous document based on an analysis of the stylistic tendencies of the potential authors.

Generally, the authorship attribution process first requires taking a set of documents and extracting stylometric features — particular measurable qualities of the documents that characterize the writing style of the author. The goal of the feature extraction process is to find features that can serve as effective author invariants — features of a text that generally remain consistent between different works written by the same author. In the case of Mosteller and Wallace, the stylometric feature of choice was word frequency: it was hypothesized that Madison and Hamilton's use of particular high-frequency

function words — such as "upon" — would be consistent and distinguishable across different works within the Federalist Papers so as to act as an effective discriminator.

Stamatatos (2009) identified several other measurable stylometric features that have served as effective author invariants in the authorship attribution literature. Among the most common are lexical features that involve analyzing a document's text at the word level. In addition to word frequency, for which the analysis of function words has proven particularly effective, some studies have also used distribution of the length of words, the distribution of the length of sentences, and vocabulary richness (for instance, measuring the number of unique words, or the number of uncommon words below a certain frequency threshold of popularity in the language) as other mechanisms for drawing inferences about authorship.

For most of the history of work in authorship attribution, analyses predominantly emphasized word-level features such as word frequency and vocabulary richness in determining authorship (Kestemont 2014). In particular, prior to Mosteller and Wallace, low-frequency features such as uncommon nouns were often used as discriminators, as they offered a simple way to suggest authorship. The theory was that some individual writers might have inclinations to more frequently use particular words that were comparatively less common across an entire population. Using low-frequency words can be effective in some attribution cases, but reliance upon them proves not particularly robust; Kestemont noted that in cases where a writer may be writing in multiple distinct genres, the particular content-word vocabulary used in one document might drastically differ from the sorts of words used in another document. The result is that an authorship attribution classifier trained on these features might fail to attribute an anonymous document to the correct author because the distribution of the low-frequency nouns in the unlabeled (anonymous) text differs significantly from the distribution in the labeled text.

Mosteller and Wallace's work addressed these concerns by introducing the novel approach of examining high-frequency function words as an author invariant, rather than low-frequency words. While the use of uncommon low-frequency content words is largely under a writer's conscious control and could therefore be altered intentionally and easily by the author, the use of high-frequency function words generally appears to be outside of an author's conscious control (Chung and Pennebaker 2007).

Yet stylometric features need to do more than just serve as author invariants for them to prove effective in solving authorship attribution problems; they also need to be features that vary enough among a population such that they can be used as effective

Document Set	"the federal government"	"the national government"
Undisputed Hamilton	31 (36.5%)	54 (63.5%)
Undisputed Madison	35 (94.6%)	2 (5.4%)
Disputed	10 (100.0%)	0 (0.0%)

TABLE 1: Frequency of the phrases "the federal government" and "the national government" in the Federalist Papers

discriminators. Examining the frequency of the word "the," for instance, is unlikely to prove particularly useful for determining authorship, since the word is used frequently by almost all authors of English. Instead, it is necessary to search for stylometric features whose values differ significantly between different authors.

In part for this reason, word n-grams — sequence of n consecutive words in a document — have proven to be effective features for drawing conclusions about authorship. Processing all trigrams (n-grams where $n = 3$) in Hamilton and Madison's Federalist Papers, for instance, results in a number of trigrams that consistently appear frequently in both sets of documents: "of the State", "the United States", and "of the Union" are among the most popular trigrams for both authors.

Other trigrams, meanwhile, prove better discriminators. Hamilton, for instance, used the trigram "the national government" 54 times across the Federalist Papers known to be his, and used the trigram "the federal government" 31 times. Madison, meanwhile, used "the national government" just twice across all the documents known to be his, and used "the federal government" 35 times. Based on these features alone, then, it might be reasonable to infer that if the disputed papers contain a high relative frequency of the trigram "the national government," that would serve as evidence in favor of Hamilton's authorship; and if the frequency was low, it would serve as evidence in favor of Madison.

And in fact, among the twelve disputed papers, "the federal government" appears ten times, but "the national government" does not appear even once. A summary of this data is depicted in Table 1.

To still use word-level features while adding more information to the model about the structure of an author's sentences, some have taken the approach of first preprocessing each document to compute part-of-speech tags for each lexical entry in the document. With the resulting part-of-speech tags, a new feature set can be extracted: namely, n-grams over part-of-speech tags instead of n-grams over words (Gamon 2004).

Such sets of features offer some insight into the types of phrases that writers tend to use in their writing, even if the particular word choice may differ. Other stylometric features for authorship attribution have been proposed as well.

1.2 Authorship Attribution as a Learning Problem

In the context of machine learning, an authorship attribution problem can be reduced to a classification problem: given a collection of training documents of known authorship and their labels — the authors who wrote each of the documents — the task becomes one of predicting the most likely label for a new document of unknown authorship.

For most approaches to using machine learning algorithms as a means of solving authorship attribution problems, the first step is to take each training document and to convert it into a feature vector of numeric values (El Bouanani and Kassou 2014). This is typically done by processing, for each document, a list of stylometric features, each of which corresponds to one or more dimensions of the vector. The resulting vectors can then be provided to a classifier to be trained on the data.

Naive Bayes classifiers are a simple but effective classifier that have provided highly accurate results for authorship attribution problems (Howedi and Mohd 2014). The classifier takes the set of labeled data as input and uses the data to compute the probability that a particular feature will be present in a document given that the document was written by a particular author. Using Bayes' theorem, these probabilities can then be used to compute the probability that a particular author wrote a given document given the presence or non-presence of a particular feature. Use of this method to predict the overall probability that a document was written by a particular author given the values of the stylometric features associated with the document is "naive" in the sense that it assumes that all feature values are independent, but still tends to produce accurate results.

The difficulty of an authorship attribution problem depends on a number of factors. For one, increasing the number of candidate authors tends to make the problem more difficult, since there are finer-grained classifications that need to be made in order to attribute authorship to a single individual. Moreover, reducing the number of training samples of known authorship that an authorship classifier has access to will also decrease its accuracy. Additionally, if particular writers tend to share a common subject matter on which they write, are from a similar time period, or tend to use similar writing styles, those factors can also increase the difficulty of authorship attribution.

Kim et al. (2011) found that their authorship attribution algorithm was more effective at discriminating between New York Times writers from different sections of the paper — such as one writer from the business section and one writer from the health section — than two writers who wrote for the same section of the newspaper.

As a consequence, authorship attribution experiments will often attempt to choose authors that are close to one another in writing style, genre, and time period to narrow the focus of stylometric analysis as closely as possible to purely stylistic aspects of an individual's writing.

1.3 Syntactic Structure in Stylometric Analysis

In September 2018, the New York Times published an anonymous op-ed from a "senior administration official" working in the Donald Trump administration. The op-ed detailed previously unknown inner workings of the administration, and described a group of senior officials who were working to resist against the President's agenda (Dao 2018). The publication of the essay resulted in widespread speculation as to who had authored the op-ed.

Observant readers noticed that, in the penultimate paragraph of the essay, the anonymous author described the late Senator John McCain as "a lodestar for restoring honor to public life." The word "lodestar" in particular caught the attention of many as a word not typically used in everyday modern English. Some hypothesized that the word "lodestar" could be used as a discriminator of authorship: if, among senior officials in the Trump administration, one official had a tendency to use the word "lodestar" frequently in other known documents and speeches, that could in theory provide reason to believe that they had been the author of the op-ed.

And indeed, one senior official in the Trump administration was found to match the description: Vice President Mike Pence. Pence was found to have used the word "lodestar" numerous times throughout speeches and writings dating back years (Mack 2018).

Pence denied writing the op-ed. Despite this, the "lodestar" controversy raised an interesting set of questions for authorship attribution: was the use of the word itself a giveaway that Pence was indeed the author? Or could it have been that the true author had intentionally used the word "lodestar," knowing full well that the usage would raise suspicion and draw analysts' attention to suspect Pence incorrectly?

In the study of linguistic style, this attempt to obfuscate one's own writing style or imitate another's writing style is known as "adversarial stylometry," which has been studied as a means via which writing can be intentionally modified in such a way so as to prevent attribution via authorship attribution techniques (Brennan, Afroz, and Greenstadt 2011). As a result, a drawback of focusing on word-level features alone for authorship attribution is that their usage is reasonably easily under conscious control if one is attempting to obscure their own writing style, and thus is highly susceptible to adversarial stylometry.

Though most of the literature has focused on word-level lexical features for authorship attribution, other parts of the literature suggest that syntactic features are potentially more valuable feature sets since they carry with them a significant amount of deeper linguistic information about how the writer constructed their writing, and also tend to be less easily consciously manipulated in cases of adversarial stylometry. Syntactic information aims to provide an alternate insight into stylometry that lexical information doesn't capture: namely, that authors can have characteristic types of phrase structures that they tend to use more often than the average writer. Distributions over these syntactic features can thus serve as useful metrics for authorship attribution.

Chaski (2012) provided a potential reason why syntactic features are less common in modern authorship attribution methods, by pointing out that current computational methods have trouble accurately extracting syntactic information — including features like constituency structure — especially when it comes to real-world data where words may be misspelled or grammar may be imperfect.

Still, Feng, Banerjee, and Choi (2012) compared the performance of syntactic stylometric features for the problem of deception detection, and found that using "deep syntax" — such as the use of the rewriting rules produced by a context free grammar parse of a document — resulted in a 14% error reduction when compared to the use of "shallow syntax" features such as part-of-speech tagging.

Gamon (2004) found that while the use of syntactic features independently could produce reasonably accurate results for solving authorship attribution problems, more accurate results could be obtained by using "feature combinators," in which feature vectors consist of the combined results of analyzing multiple different feature sets, including context-free grammar rewriting rules, function word frequency, and binary semantic features.

Still others have taken the approach of designing new syntactic features with the

goal of capturing particular aspects of individual writing style that can be used as discriminators.

Tschuggnall and Specht (2014) proposed a method for using syntax tree analysis in the process of determining authorship. In particular, they extrapolated the lexical idea of using n-grams of words to analyze pq-grams in trees, where p is the number of syntax tree nodes to consider vertically, and q is the number of syntax tree nodes to consider horizontally. By creating an index of all possible pq-grams for a particular input text, Tschuggnall and Specht were able to develop a "syntax tree profile" of a particular author. By computing a syntax tree profile for multiple authors, and then comparing the distribution of pq-grams of an unlabeled input document to the known profiles, they were able to use the syntactic information to draw conclusions about authorship.

Kim et al. (2011) approached the problem of authorship attribution by designing yet another novel syntactic feature — that of the k-embedded-edge subtree — with the intention of capturing syntactic information that reflects writers' natural tendencies to structure their sentences in particular ways. While subtrees of a particular size generally are only able to capture local relationships between nodes in a tree, k-embedded-edge subtrees are able to encode for longer-distance dependencies. In a k-embedded-edge subtree, the subtree is allowed to have at most k embedded edges, where an embedded edge is an edge between two nodes where there is an ancestor-descendant relationship in the original tree, but not a parent-child relationship.

Dependency grammars have also been proposed as an alternative parsing method for sentences. In such grammars, links exist to connect individual words with other semantically related words, without the internal node structure commonly found in phrase structure grammars. Covington (2001) identified that the use of dependency grammars has a number of advantages over constituency grammars: in particular, dependency grammars express the semantic relationship between words in a more direct way that constituency grammars. Covington also presented a parsing algorithm for generating dependency trees for a given sentence, potentially addressing the concerns raised by Chaski and Kim et al. about the inaccuracy in phrase structure parsing. Despite this, Wennberg (2012) compared phrase structure grammars with dependency grammars and evaluated their effectiveness as discriminators of authorship, and found that attribution based on dependency grammars performed significantly less well compared to attribution based on phrase structure grammars.

Still, syntactic features have generally shown a demonstrated ability to serve as effective authorship discriminators, suggesting that an author's natural tendencies in

syntactic structure contain valuable information for performing authorship attribution tasks. In particular, when attempting to solve authorship problems where less data is available, or where some of the constraints of the problem — like the single authorship assumption — are relaxed, syntactic features offer additional data beyond what lexical features offer that have the potential to improve the accuracy and reliability of authorship attribution systems.

Chapter 2

The Multiple Authorship Problem

2.1 The Assumption of Single Authorship

Central to most attempts to solve the authorship attribution problem is the assumption of single authorship: namely, the assumption that each document was written by just a single author. This assumption simplifies the classification process, since the authorship system needs only to predict a single authorship label for a document, and thus needs only to calculate the most likely author from a set of candidate authors.

Relaxing this assumption, however, leads to a related, but more nuanced, question: given a document that may have been jointly written by multiple authors, how can an authorship attribution system determine which authors wrote which sections of the document?

The problem of authorship attribution in a setting where each document belongs to only a single author is one in which features can be extracted from anywhere throughout the document in order to estimate the author that is the most likely author of the document in question. In an authorship attribution system capable of handling multi-author documents, a different approach is necessary.

2.1.1 Historical Problems of Multiple Authorship

This problem of the attribution of documents that may be written by more than one author has a variety of historical instantiations. In the late 8th century, Charlemagne ordered the writing of the *Libri Carolini*, a theological treatise that critiqued the Second Council of Nicaea (Shahan 1908). There is some dispute, however, over who actually wrote the *Libri Carolini*. While most attribute the works to Theodulf of Orléans — a writer and an advisor to Charlemagne — Wallach, Augoustakis, and Wallach (2017) pointed out that there is some evidence that portions of the works were written by

another of Charlemagne's advisors, Alcuin of York. In this case, there are works of known authorship written by both Theodulf and Alcuin, and it is therefore reasonable to imagine that a system capable of handling multi-author documents could analyze the *Libri Carolini* to determine which sections of the document, if any, were written by these two writers.

Another source of authorship controversy lies with the works of Hildegard of Bingen, one of the most notable female scientists and writers of the Middle Ages. Kestemont, Moens, and Deploige (2015) described how Hildegard would generally dictate her works to one of her scribes, and her scribes would handle the actual writing process. However, one of her scribes, Guibert of Gembloux, would often make substantial changes to Hildegard's work in the process of scribing. Historians have thus raised the question of how much of of Hildegard's work is her own, and how much is actually the writing of Guibert. For many of the works Hildegard published while Guibert was her scribe, multiple authorship is therefore quite likely. Kestemont, Moens, and Deploige (2015) performed classical single-author attribution on some of Hildegard's shorter texts, and found that their classifier attributed the texts to Guibert instead of Hildegard, suggesting that Guibert made significant contributions to the work. But single-author classifiers are unable to answer questions about which portions of the works should be attributed to Guibert and which should be attributed to Hildegard: if one knew the boundaries at which point one author's writing ended and another began, it would be possible to segment the work along boundary lines and treat each segmentation as a document of its own. But in this case, and in many cases, the section boundaries are unknown, and conventional single-author classification methods are thus inadequate for solving such problems.

Even in the case of the twelve disputed Federalist Papers that Mosteller and Wallace (1963) concluded were written by Madison, some scholars believe the true answer to the dispute lies with multiple authorship. Rudman (2012) critiqued Mosteller and Wallace's approach and suggested that the twelve disputed papers quite likely were a joint effort, where parts of each paper were written by Hamilton, and parts of each written by Madison. Though Rudman did not attempt to determine which parts of the papers were written by which author, a multi-author attribution system capable of detecting section boundaries and assigning authorship appropriately would be able to make such predictions.

2.2 Multiple Authorship

One could imagine modeling the authorship problem in a number of different ways: in the most general sense, multiple authors may collaborate together on the choice of a particular word, and therefore each word in a document might be attributed to some non-empty subset of the set of authors for the document as a whole. One of the only existing works studying multiple authorship attribution methods is by Althoff, Britz, and Shan (2014), whose authorship model allows for a many-to-many mapping of documents to their possible authors, where each author is the author of some subset of the sentences in the document.

Though in practice, a single sentence might be the work of multiple authors, Althoff, Britz, and Shan made the simplifying assumption that each sentence was written by just a single author; they then endeavored to identify, given a document for which each sentence was written by one of a number of candidate authors, the author associated with each sentence, as well as the overall authors of the entire document. This assumption is useful since it allows the use of sentence-level authorship models that can resemble how conventional document-level authorship models work, and can predict the most likely author of each sentence. An individual sentence, however, contains far less stylometric data than a document composed of many sentences, so sentence-level predictions are less accurate.

To test their model, Althoff, Britz, and Shan generated synthetic documents from a set of single-author papers of known authorship. Each synthetic document was composed of a sequence of paragraphs, where each paragraph consisted of multiple sentences written by a single author. Althoff's model, however, predicted the author of each sentence based solely upon the sentence itself, without consideration for this structure of multi-author documents where a single author is more likely to write multiple sentences in sequence. In practice, this assumption is often a reasonable one: when multiple authors contribute to the writing of a single work, even maintaining the assumption that each sentence is written by a single author, a single author is likely to write multiple consecutive sentences in a "section" of the text. Thus, conditioning on knowing the author of a particular sentence increases the probability that the following sentence was written by that author.

Formally, in a sequence of consecutive sentences $S_1, S_2, ..., S_n$ in a multi-author document, and a function $\text{AUTHOR}(S_j)$ that returns the author of sentence S_j, $P(\text{AUTHOR}(S_j) = a | \text{AUTHOR}(S_{j-1}) = a) > P(\text{AUTHOR}(S_j) = a)$ likely holds.

2.2.1 Defining the Multiple Authorship Attribution Problem

Expanding on the ideas of Althoff, Britz, and Shan (2014), I present a version of the multiple authorship attribution problem, and aim to determine the stylometric features that prove most effective for predicting sentence-level authorship in such a setting. Additionally, I also aim to improve upon sentence-level authorship predictions by aggregating sentence-level results to predict section-level authorship, even when section boundaries of a document are not given as part of the input.

In particular, given a set of documents, I assume each document can be subdivided into "sections," where each section is a sequence of contiguous sentences written by the same author. The training data for this model is the same as the training data in a single-author attribution model, but at the sentence level instead of the document level — the classifier takes as input a set of sentences of known singular authorship for each of the potential candidate authors. The task, then, is to take a document in the form of a sequence of sentences of unknown authorship, where section boundaries are also unknown, and perform the following two prediction tasks:

1. Predict the boundaries between sections: in other words, determine at what points in the document one author stopped writing and another author began writing.

2. Predict, for each of the estimated sections, the most likely author of the section.

Figure 1 gives an example of an abstract document with all of the section and author labels present. The document is subdivided into sections, not necessarily of the same length, for which each section has a single author who is the author of each of the sentences in that section. Note that a single author can be assigned to more than one section: in the example given, Author 2 is the author of both the second section as well as the final section.

This version of the problem presents a number of interesting challenges that are distinct from traditional authorship attribution tasks: in particular, that attributing authorship on a sentence level is more challenging than on a document level, given that there is far more limited data about the unlabeled sentence that a classifier has access to, as compared to an entire unlabeled document.

2.2.2 From Sentence-Level to Section-Level Predictions

The design of a sentence-level classifier is similar in principle to a classifier that attributes authorship to entire documents. The training data takes the form of sample

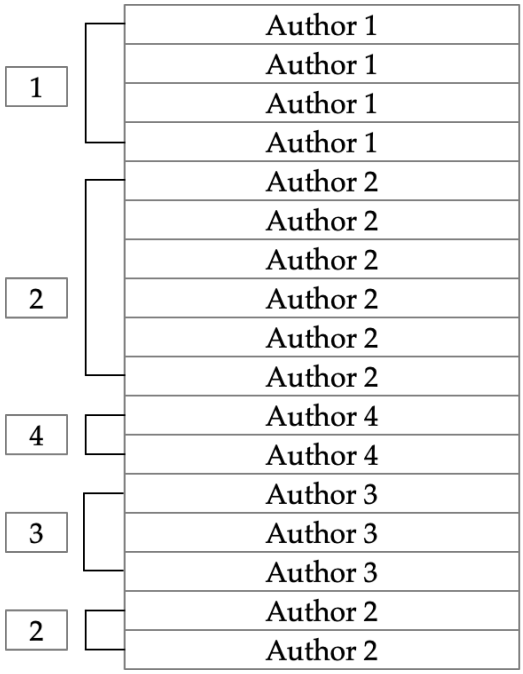

FIGURE 1: A sample model of a document subdivided into sections. Each section consists of a sequence of sentences, all of which are written by the same author. The goal for the authorship attribution system is to predict where the section boundaries are (denoted by the brackets to the left of each sentence), and the author that wrote each section (denoted by the number to the left of each section).

sentences of known authorship, and the classifier's job is to take as input each sentence in the unlabeled document and predict the most likely author for that sentence. Of course, some features that are commonly used in document-level authorship attribution — such as average sentence length — might be less meaningful or less useful in the context of extracting features to characterize an entire sample, and other features — such as average number of word per paragraph — are ill-defined for samples of only a single sentence.

However, treating each sentence as entirely independent fails to take advantage of the information that authors are likely to write multiple sentences in a row in a single section. Should a situation arise where a classifier found that in a sequence of ten consecutive sentences, all but one were attributed to author A_i, and one sentence in the middle was attributed to a different author A_j, it may be more likely that all ten sentences belong to a single section written by A_i rather than that author A_j having a section of length 1 in between two sections by author A_i. Of course, it is also possible that author A_j has an intervening section of length greater than 1, but the classifier incorrectly predicted author A_i on sentences that actually belong to A_j.

Still, it may be possible to take sentence-level predictions and refine them into section-level predictions that are more accurate, by taking advantage of the entire sequence of predictions. Doing so would suggest a two-step process for multiple authorship attribution: computing sentence-level predictions independently, and then combining those predictions to compute section-level predictions.

The prediction algorithm I propose, then, follows those two general steps:

First, the algorithm will compute sentence-level features, in a manner similar to how prior attribution systems have classified entire documents. For each sentence, a fixed set of numeric features is extracted in order to form a fixed-length feature vector representing the sentence in question. This vector is then provided to a classifier, which compares the feature vector against the feature vectors of all of the sentences of known authorship used to train the classifier, and outputs a most likely author for the given sentence.

After the sentence-level predictions are made, the next step is to use those predictions to estimate the likely boundaries between sections, and to estimate the likely authors of each of the sections. Though in theory, the definition of a section given here doesn't preclude a section from just being a single sentence written by an author, in general it is reasonable to assume that most sections will consist of multiple sentences written by the same author.

Importantly, if a sequence of sentences is predicted to be mostly by the same author, but there are a few intervening sentences in between for which the classifier predicted were written by different authors, then it may be reasonable to conclude that rather than switching frequently between authors in short sections, the more likely option may be that the sentence-level classification is inaccurate and that the sequence of sentences belongs to a section written by a single author.

Figure 2 shows an example of this sort of prediction. The high-level intuition for the prediction algorithm works as follows: The sentence-level predictions assign, for each sentence, an author that is the most likely author for the sentence. But of the first seven sentences, five of them are predicted to be by Author 1, and the other two are predicted to be by other authors. So, in the prediction step for the section boundaries, it is reasonable to guess that those sentences form a section by Author 1. Meanwhile, immediately following the first seven sentences are a sequence of sentences that are mostly written by Author 2, with a few intervening sentences that are predicted to be by other authors in between.

Based on that information, it is likely reasonable to draw two conclusions: first, that there is likely a section boundary between authors somewhere around the seven sentence mark; and second, that the second section is likely written by Author 2.

2.2.3 Inferring Authorship from Sentence Adjacency

In addition to the information about an author writing multiple consecutive sentences in the same section, there may be other measurable dependency effects between sentences, in particular when considering adjacent sentences. Samardzhiev, Gargett, and Bollegala (2017) proposed a model of Neural Word Salience scores that could be used to predict the words that appear in a sentence, given the words that appear in the sentence before it or the sentence after it. This suggests a dependency between adjacent sentences that goes beyond the fact that they are written by the same individual.

This also raises an interesting possibility for the problem of multiple authorship attribution, and suggests a potential optimization. Given that there is a measurable relationship between the content and structure of adjacent sentences, it's possible that such a relationship is measurably different between adjacent sentences written by the same author, and adjacent sentences written by different authors. Capturing that relationship may improve the results of section-level authorship predictions: if a classifier predicts that $\text{AUTHOR}(S_j) = A_i$, and a model for assessing the relationship between adjacent

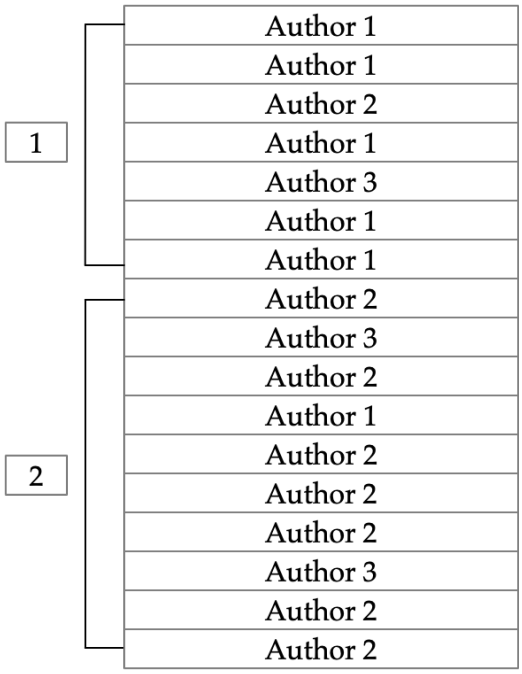

FIGURE 2: An example of sentence-level and section-level predictions. On the sentence level, each sentence is treated independently, and the learning algorithm predicts the most likely author for that particular sentence. On the section level, the algorithm looks for the most likely dividing lines between sections and for the most likely section authors. Here, some of the noise of the sentence-level predictions is reduced.

sentences suggests that S_j and S_{j+1} are like written by the same author, that affects the probability distribution for the prediction of the system's guess for $\text{AUTHOR}(S_{j+1})$.

2.3 Features

In order to represent the sentence-level authorship attribution problem as a learning problem, it is useful to represent each sentence as a numeric feature vector to encapsulate the relevant features of the data. The goal, though, is to identify a feature vector for a sentence that captures not just information about the sentence itself, but in particular captures information about the writing style of the sentence's author. This section explores some commonly used features in authorship attribution tasks.

2.3.1 Word-Level n-grams

Perhaps the simplest of features to use when considering extracting features from a text is word frequency, where the idea is that certain authors likely tend to use certain words more frequently or less frequently than other authors do. But reducing a text down to only the words that compose it removes important stylometric information about the style of the writer: in particular, it ignores everything about the ordering of words. It is reasonable to imagine that some authors tend towards using particular sequences of words more or less frequently than other authors.

To address this, a useful set of features to extract are the set of word n-grams, where an n-gram is a sequence of n contiguous words.

However, if when testing an authorship attribution model, testing and training sentences are both drawn from the same sample text (a novel, for instance), then even if those sentences are distinct, using pure word n-grams as features skews the results to be unnaturally accurate. In particular, because word-level n-grams capture very context-specific words like proper nouns, learning models could focus on those features regardless of the author's actual writing style. For instance, even knowing nothing about the style of J.K. Rowling's writing, it would be quite reasonable to assume that if the word "Hermione" appeared in a feature vector for a sentence taken from fiction writing, J.K. Rowling is a likely choice for the author of the sentence.

Thus, in order to accurately assess whether an authorship attribution model can correctly learn an author's writing style, a different approach is needed.

There are a few solutions to this problem that can be employed to address this issue.

For one, the model can look at only at the distribution of a fixed list of function words, described in subsection 2.3.3. This is similar to the work done to determine the authorship of the Federalist Papers (Mosteller and Wallace 1963): training the model on the frequencies with which various authors use common function words.

Still, there are a number of words that are common words that could likely act as good predictors of authorship, even if the words themselves are not considered function words. So additionally, to avoid over-fitting on the training data, the model can be trained on a pre-processed version of the text: a version of the text where all words other than the approximately 5000 most common words are replaced with the identifier token "UNKNOWN". A sentence like the following, taken from one of James Madison's essays in the Federalist Papers (Madison 1787):

(4) Liberty is to faction what air is to fire, an aliment without which it instantly expires.

would be pre-processed into the following:

(5) UNKNOWN is to UNKNOWN what air is to fire, an UNKNOWN without which it instantly UNKNOWN.

This avoids the proper noun or context-specific noun problem by focusing on a limited pre-selected list of words from which to extract feature information from. This also minimizes the number of needed features, and helps to focus the model on the features that are most likely going to appear frequently across the data set.

2.3.2 Part of Speech n-grams

It is also possible to compute n-grams over tokens that are not individual words, and thereby capture a different part of linguistic style that may not be captured by taking n-grams over words alone.

Similar styles of phrases, though they might differ in the specific words that are used, may have similar parts of speech in common. Another useful feature is thus to look at n-grams over part-of-speech tags, which gives general information about the types of words that are commonly found in sequence in a particular author's writing.

Taking the following sentence again:

(6) Liberty is to faction what air is to fire, an aliment without which it instantly expires.

Running a part-of-speech tagger on the sentence results in the following sequence of parts of speech:

(7) NN VBZ TO NN WP NN VBZ TO NN DT NN IN WDT PRP RB VBD

The parts of speech used here are the part-of-speech tags used by the Penn Treebank Project. Note that even in this sentence, the 4-gram of part of speech tags "NN VBZ TO NN" appears twice, the result of tagging the sequences of words "liberty is to faction" and "air is to fire". Here, NN refers to a singular or mass noun, VBZ refers to a verb in the third person singular present, and TO is a tag reserved for the word "to". The part-of-speech *n*-gram extraction is thus able to identify similar types of phrases, even if the phrases themselves are not identical, and therefore serve as potentially useful discriminating features for documents, especially if those documents do not themselves contain many phrases that are also phrases in the training set.

Part-of-speech tagging and pre-processing to remove rare words can also be combined to form a new sequence: the sequence of common words, with rare words replaced with their corresponding part of speech. The above sentence would then become:

(8) NN is to NN what air is to fire, an NN without which it instantly VBD

2.3.3 Distribution of Function Words

Following the example of Mosteller and Wallace (1963), another common feature set for use in authorship attribution is the distribution of function words, the words that work in English grammar to indicate the relationship between the "content words" that carry more of the lexical meaning. While some tasks in learning — such as topic-based text classification — will frequently ignore function words since they don't contribute much to an understanding of the topic of a document, authorship attribution systems make frequent use of function words as features, since individuals will often have natural tendencies to use certain function words more or less frequently than other individuals in a population, as a matter of their own linguistic style (Stamatatos 2009).

According to Chung and Pennebaker (2007), fewer than 400 function words that make up less than 0.04% of the vocabulary of the average native English speaker end up accounting for more than half of the words that are used in regular speech. As a result of this frequent use, this limited set of words carries a lot of information that can be used to differentiate one author from another. A simple classifier could take each

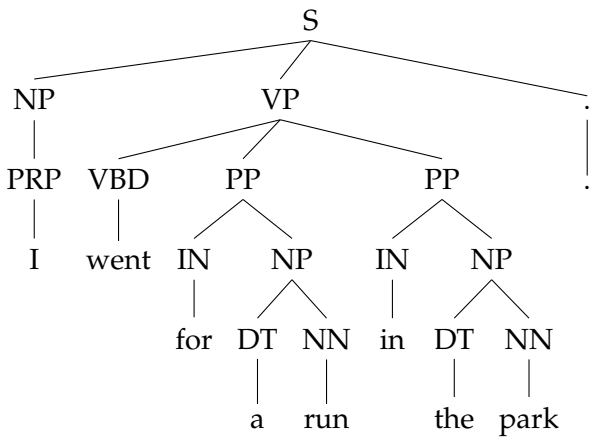

FIGURE 3: An example of a sentence parsed as a context-free grammar, as generated by the Stanford Natural Language Parser.

training document and compute the frequency of each function word, combining the results into an approximately 400-dimensional vector that could then be used to predict future authorship based on the distribution of the frequencies of those same function words.

2.3.4 Syntactic Structure

Among syntax-based features for authorship attribution, one of the most common is to examine the distribution of context-free grammar rewriting rules (Gamon 2004). Given a sentence, modern parsers such as the Stanford Natural Language Parser can generate a probable syntax tree for the sentence (Klein and Manning 2003). Once the syntactic structure is known, authorship attribution systems can look to the distribution of rewriting rules used in the derivation of the sentence, using that distribution as a feature set to characterize a particular author's syntactic tendencies.

Consider the example in Figure 3. The rewrite rule S $\rightarrow$ NP VP . is used once (as the very first derivation in the sentence), but some other rules, including PP $\rightarrow$ IN NP and NP $\rightarrow$ DT NN, are used twice. Gamon (2004) suggested that individual writers often consistently differ from one another with the frequency in which they use various rewriting rules, thus allowing the syntactic structure of a person's language use to serve as a measurable factor in determining authorship.

Chapter 3

Models and Methods

3.1 Data for Multi-Author Attribution

3.1.1 Corpus

The difficulty of the authorship attribution process depends upon the choice of corpus: choosing authors who are very different from one another in time period, genre, and location, for example, will result in greater stylistic differences between authors that are therefore more easily captured. Meanwhile, the number of candidate authors also affects the probability of success — as the number of candidate authors increases, so does the difficulty of the attribution process.

Several authorship attribution studies, include Gamon (2004), therefore have chosen to use the works of the three Brontë sisters: Anne Brontë, Charlotte Brontë, and Emily Brontë. These 19th century English writers shared a common education, time period, and style, and therefore discriminating between the three writers proves to be an interesting challenge — significantly more difficult than differentiating, say, the writings of Mark Twain from the writings of Ernest Hemingway (Hemingway is well known for keeping his sentences short, for instance, while Twain has a sentence in *Adventures of Huckleberry Finn* that contains a sentence upwards of 250 words long).

For this experiment, the following works of the Brontë sisters were used, consistent with the works used in authorship experiments performed by Gamon (2004):

- *Agnes Grey* by Anne Brontë
- *Wildfell Hall* by Anne Brontë
- *Jane Eyre* by Charlotte Brontë
- *The Professor* by Charlotte Brontë

- *Wuthering Heights* by Emily Brontë

3.1.2 Document Synthesis

In order to test the performance of a multi-author authorship attribution system, the system first requires a set of properly labeled multi-author documents upon which to test. The difficulty is that most documents in real-world data sets that are written by multiple authors, whether they be jointly written papers or books that are coauthored, do not have authorship information at the sentence level to determine the author for any particular sentence. Instead, a more practical method for data collection is to synthesize artificial multi-author documents from a corpus of single-author documents.

To synthesize documents of this form, it is possible to simulate the generation of multi-author documents using a discrete-time Markov chain where, in addition to a starting state and terminating state, the model also has one state for each author who is a possible contributor to the document. At every point in time t, if the Markov process is not in a starting or terminating state, then a new sentence written by the author corresponding with the current state of the process is generated. To do so, it is also necessary to have a corpus of sentences of known authorship from which to draw.

When synthesizing a document with n possible authors, the transition probabilities of moving from the start state to any individual author state is $\frac{1}{n}$. From any given author state, there is some (likely high) probability s of remaining in the same state after the next transition — in other words, allowing the author to contribute another sentence as part of the current section of the document. With some low probability t, the model terminates, and that marks the end of the document. The probability of switching from any one author to any other author, then, is $\frac{1-t-s}{n-1}$.

Figure 4 demonstrates such a model for a three-author document generation process. The likely result of running this model with sufficiently high s and sufficiently low t is that a document is generated that contains multiple sections of varying length, each containing sentences written by the same author.

For this experiment, $s = 0.95$ and $t = 0.01$ were chosen as the transition probabilities for staying with the same author and terminating, respectively. Thus, the transition model for this Markov chain is defined as

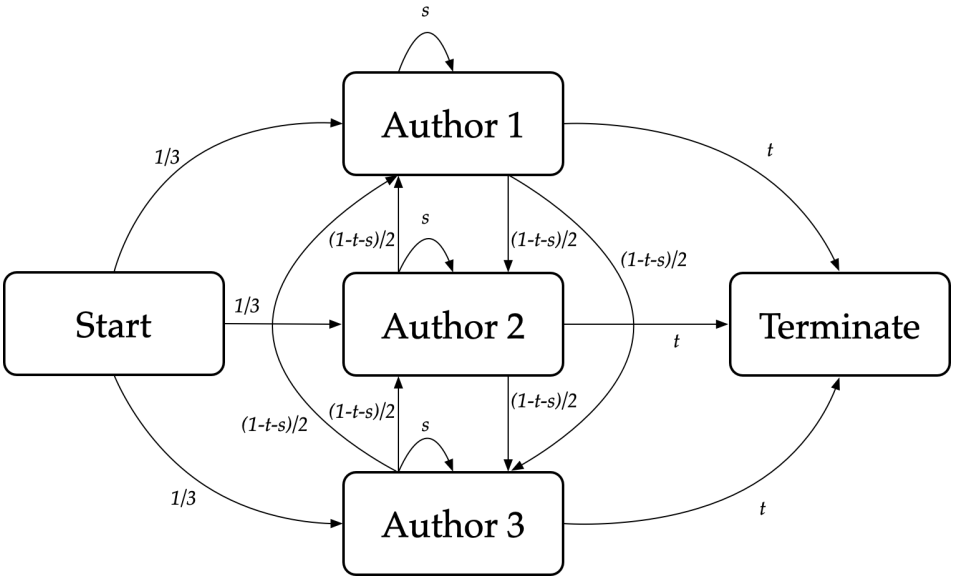

FIGURE 4: An example of a theoretical Markov model that could be used to generate documents from three authors (though the model is easily extended to handle more than three authors). An initial author is selected uniformly at random, and then for each time an author's state is reached, a random sentence from that author is added to the document. With some probability s, the model stays on the current author and the author contributes another sentence sequentially to the document. With some low probability t, the model terminates. Otherwise, the model switches to a new section author.

$$P = \begin{array}{c} \\ Start \\ Anne \\ Charlotte \\ Emily \\ Terminate \end{array} \begin{array}{c} \begin{array}{ccccc} Start & Anne & Charlotte & Emily & Terminate \end{array} \\ \begin{pmatrix} 0 & 0.33 & 0.33 & 0.33 & 0 \\ 0 & 0.95 & 0.02 & 0.02 & 0.01 \\ 0 & 0.02 & 0.95 & 0.02 & 0.01 \\ 0 & 0.02 & 0.02 & 0.95 & 0.01 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Here, P_{ij} is the probability of moving from state i to state j. Once the process reaches the termination state, the document generation stops. To ensure each document is long enough to be interesting to analyze, if a document generation process terminated with fewer than 50 sentences in the document, the document was immediately discarded and a new document was re-generated, so that all documents would have a length greater than some minimum threshold.

Once the documents are generated, the next step is to extract features from sentences in order to perform the sentence-level prediction task.

3.2 Proposed Syntactic Features

Here, a number of novel features are proposed that could be used for attribution authorship.

3.2.1 Combinatory Categorial Grammars

Combinatory categorial grammars are another grammatical formalism for describing the structure of language, which has qualities that make it appealing as a potential candidate for use in authorship attribution problems, but to date has not yet been explored in an authorship attribution context.

In a combinatory categorial grammar, each lexical item is assigned a syntactic category (Steedman 1996), similar in spirit to a part of speech, but containing more information about the word's role in the larger grammatical construction. In particular, while some objects like nouns have syntactic categories that are similar to parts of speech (NP), other words like verbs have syntactic categories that are actually functions over parts of speech, combining with another category to the left or to the right to produce an output category.

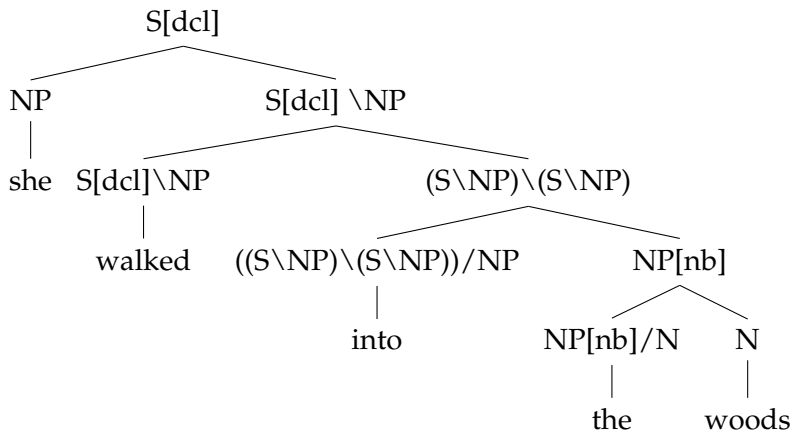

FIGURE 5: A CCG parse of a sentence. Words are labeled with CCG tags. Some tags act as functions that take other items as arguments, and return a new type.

Combinatory categorial grammars appear to have potential for applications to authorship attribution problems for a number of reasons. For one, efficient and accurate parsers exist for taking English sentences and parsing them into their combinatory categorial structure. Moreover, such structure results in tagging information that may contain more syntactic information about words and their role in a sentence than standard part-of-speech tagging would: suggesting that, at minimum, given enough training data, using n-grams over CCG tags could prove a more accurate discriminator than n-grams over part-of-speech tags. Likewise, the actual frequency of combinators themselves and the arguments to which they are applied could prove to be a more syntactically meaningful analog to traditional context-free-grammar rewriting rules.

An example of a CCG parse of a sentence is in Figure 5. In particular, note that associated with each word is a tag that describes the role of the word in the sentence relative to the rest of the structure of the sentence.

Just as n-grams over part-of-speech tags can be extracted from text, n-grams over CCG tags can be extracted as well. Since the result of a CCG parse of a sentence is a tree, rather than a flat list of labels, as is the case with part-of-speech tagging, there are more options available for how to extract CCG information.

For one, it is possible to perform a direct analog of the part-of-speech tagging using CCG tags: take each word's CCG tag as defined by the parser, and use the sequence of tags in sequence as the list from which to extract n-grams.

However, the hierarchical nature of CCG parses offer other information that can be

extracted into numerical features as well. For one, CCG tags compose with one another via combinators: a tag of type NP\NP might combine with a tag of type NP to form a larger phrase structure of type NP, for instance. The frequencies of the combinators, frequencies of the types that combine with one another, and the depths in the trees at which those frequencies occur, can all make for features that allow for prediction of authorship.

3.2.2 Constituency-Preserving Embedded Subtrees

In the spirit of Mosteller and Wallace 1963, it could be potentially useful to add a level of syntactic analysis to the distribution of function words when considering function words as features to use in authorship classification. While Mosteller and Wallace considered words independently of their context, it is not unreasonable to imagine that certain individuals have a tendency to use particular structures of function words more than other individuals.

Kim et al. (2011) introduced the idea of using embedded subtrees as features for authorship classification. In an embedded subtree, a subset of the nodes of a syntax tree are considered as the feature set, and the edges between the nodes can represent any ancestor-descendant relationship from the original tree, not just a parent-child relationship.

Here, I propose the novel feature set of constituency-preserving embedded subtrees over function words, where the subtree contains only leaf nodes that correspond to function words from the original tree. Meanwhile, the constituency-preserving quality is defined as follows: a subtree s is a constituency-preserving embedded subtree over function words of a tree t if, for all constituents of t, the function words dominated by t are also the leaf nodes of a constituent of s. Symmetrically, for any constituent of s, the leaf nodes within it correspond to the function words contained within a constituent of t.

Take the sentence: "She walked into the woods."

Parsed using the Stanford CoreNLP parser, we get tree (a) in Figure 6.

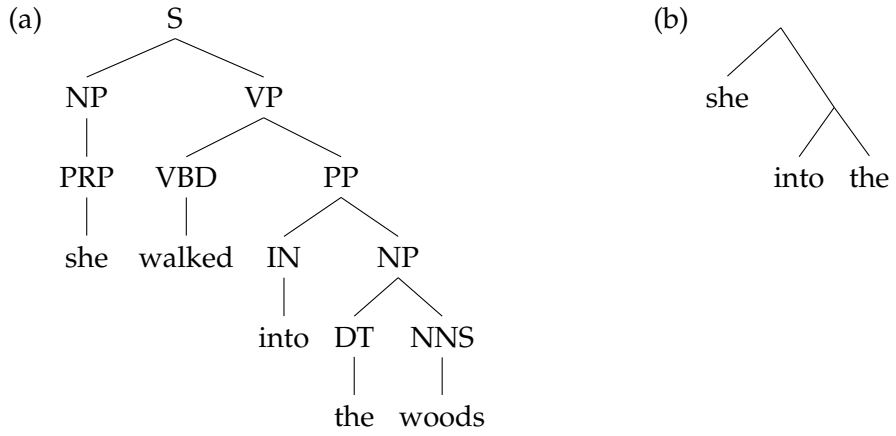

FIGURE 6: Tree (a) shows the constituency parse of the sentence. Tree (b) shows the embedded subtree over function words. Only function words are included in tree (b), and for any constituent in (a), all function words contained within the constituent also form a constituent in (b).

The function words in the sentence are "she," "into," and "the." We can form the embedded subtree over function words by taking embedded edges between function words in the constituency parse and translating them into edges of the embedded subtree. The result is seen in tree (b) in Figure 6.

This subtree represents a number of characteristic qualities of the sentence: the number and use of function words, an approximation of the constituency structure of the sentence, and information about the function words in relation to one another. Authorship attribution can then be performed using the constituents of these subtrees as a feature set.

A node's depth in the tree is not preserved in the transformation from constituency tree to embedded subtree. Note that in Figure 6, "into" and "the" are at different depths in the constituency tree, but at the same depth in the function word embedded subtree.

3.2.3 Lexical Order-Preserving Syntactic n-grams

A number of attempts have been made to capture syntactic information in n-grams for use as features in machine learning systems. Sidorov et al. (2012) described a model for syntactic n-grams for machine learning where, rather than take n-grams over words in the lexical order in which they appear in the original text, syntactic n-grams are extracted by taking words in the order in which they appear in a depth-first traversal of

the sentence's dependency structure tree. Wennberg (2012) attempted to use such dependency grammar relations as features in authorship attribution, but found that using dependency grammar structure as a feature proved less effective than keeping words in lexical order, possibly because the lexical order captures more about an individual writer's stylistic tendencies than the dependency structure does.

However, it is still possible to model n-grams (over words or part-of-speech tags) in such a way that n-grams can capture syntactic structure while also preserving lexical order. Here, I propose the feature of lexical order-preserving syntactic n-grams for authorship attribution (abbreviated later as LOPS n-grams), which use the syntactic structure of the n-gram as a tree in addition to just the lexical order.

For example, consider the following pair of sentences, in particular noting the underlined n-gram for $n = 5$:

(9) She saw the bowl of candy.

(10) It reminded the girl of home.

Representing each of the sentences as a syntax tree yields the trees in Figure 7. The underlined words in each sentence contain the same linear sequence of part-of-speech tags, "VBD DT NN IN NN", but their syntactic structure is different: in the first sentence, the prepositional phrase is contained within the larger noun phrase; in the second sentence, the prepositional phrase and the noun phrase are on the same level of the tree.

Though the standard lexical part-of-speech n-grams would be identical here, the LOPS n-grams for these sentences would be distinct, as contrasted in Figure 8.

As a matter of implementation, these features can be represented in bracket notation as the strings "[[VBD][[[DT][NN]][[IN][NN]]]]" and "[[VBD][[DT][NN]][[IN][NN]]]", respectively.

3.3 Sentence and Section Classification Models

3.3.1 Naive Bayes Classification for Sentence-Level Predictions

For these experiments, I make use of a Naive Bayes classifier, which predicts a label for data based on the conditional probability of the label given some evidence. The classifier is used to predict the most likely author of any given sentence. Under this model, the classifier takes as training input a set of $(label, vector)$ pairs, where the *label*

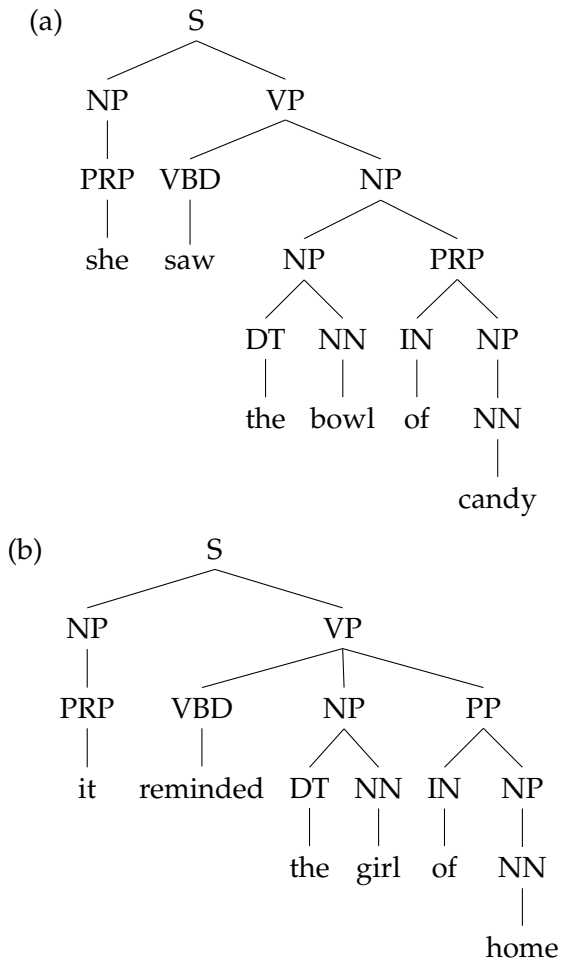

FIGURE 7: In both trees, the sequence of part-of-speech tag sequence "VBD DT NN IN NN" appears, but the syntactic structural relationship between those tags is different.

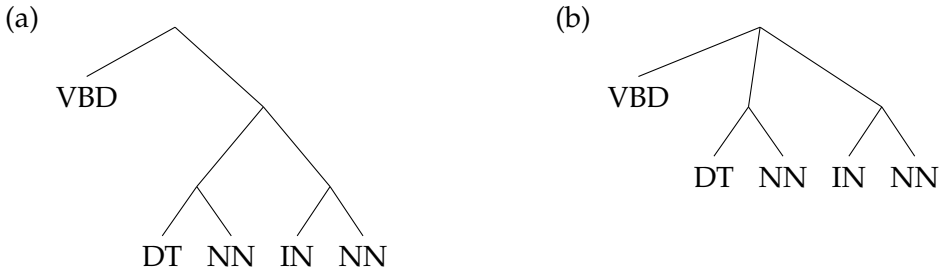

FIGURE 8: In cases where two lexical n-grams may be identical, their syntactic n-gram equivalents may differ.

is the name of the author of the sentence, and *vector* is the numeric feature vector corresponding to that particular sentence. The classifier then, given a new feature vector, attempts to predict the appropriate label via use of Bayes rule: computing the conditional probability that the sentence was written by a particular author, given the values of the features in the feature vector. The most likely author is the one the classifier will estimate.

It is worth noting that the features extracted from each sentence here do not perfectly fit the assumptions of a Naive Bayes Classifier. In particular, the classifier assumes that the values of all of the features are independent of one another, which is most definitely not the case: conditioning on the values of certain features in this model likely affects the distribution of other feature values: the presence of the trigram "better late than", for instance, almost certainly increases the likelihood that the trigram "late than never" will also be present in the document. However, several authorship studies have found that assuming independence of features still results in accurate results for authorship (Howedi and Mohd 2014).

3.3.2 Section Boundary Detection and Authorship Attribution

Once sentence-level authorship predictions are determined, the next step is to use those predictions to infer section boundaries and section-level authorship. Here, a Hidden Markov Model can be used to model the prediction process. The model has a sequence of underlying (hidden) states: one state per sentence, where the state corresponds to the true author of the sentence. The sentences are observed, and the sentence-level classifier predicts an author for each sentence. The result is a prediction, for each observed state, that hopefully correlates with the actual value of the underlying state. Figure 9 depicts this model pictorially.

Formally, we define the model for authorship attribution over N candidate authors as (A, B, π). The transition probabilities are given by A, a $N \times N$ matrix where A_{ij} is the probability of moving from author i to author j between sentences. The emission probabilities are given by B, a $N \times N$ matrix where B_{ij} is the probability of predicting that author j wrote the sentence given that author i wrote the sentence. The initial state distribution is given by π, a vector of length N such that π_i is the probability of the document starting with author i.

If our learning model knew the actual transition probabilities A between states and the emission distribution B for the probability of guessing a particular author given the true author, then we could predict the underlying states via maximum a posteriori

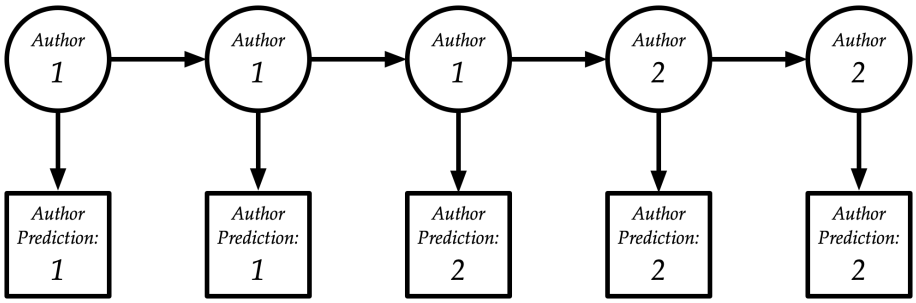

FIGURE 9: Modeling the multi-author attribution problem as a Hidden Markov Model. There is one underlying state per sentence, and the state represents the author of that sentence. The sentence is observed, and each sentence has a prediction for the most likely author of the sentence. This prediction, however, may not always correspond with the true author, as with the third sentence in this example.

estimation or via the Viterbi algorithm, which, given a Hidden Markov Model and a sequence of observed emission states, calculates the most likely sequence of underlying states (Stratonovich 1960). Though the document synthesis process itself required the setting of parameters s and t to determine the transition probabilities, in practice it is not reasonable to assume that for a non-synthesized document one would know the transition probabilities. Thus, it is instead necessary to find a strategy for predicting these unknown parameters.

The Baum-Welch algorithm is well suited for this task. The Baum-Welch algorithm seeks to determine what the underlying transition probabilities in the Hidden Markov Model are based on the observed states. This happens through an iterative estimation process: the algorithm begins by choosing transition probabilities at random, and then uses those probabilities to estimate the probability of being in a given state at each point in the process via the forward-backward algorithm. Then, the algorithm re-estimates the transition probabilities based on the frequency of transitions in the estimated underlying states. This process repeats until the transition probabilities converge, at which point there is a final prediction for the transition probabilities and, by extension, a prediction of the most likely underlying states.

3.3.3 Sentence Adjacency and the Markov Assumption

The Hidden Markov Model maintains the Markov assumption that the probability of an observed state depends only on the underlying state, and is independent of the state or the emission that came before it. In practice in a multiple authorship attribution

setting, however, this assumption likely does not hold. Work by Samardzhiev, Gargett, and Bollegala (2017) suggests that given a sentence, it is possible to predict the words that appear in an adjacent sentence. Hypothesizing that this relationship is stronger in adjacent sentences that are written by the same author than in adjacent sentences written by different authors, this information can be used to potentially improve the accuracy of section-level predictions.

Utilizing this adjacency effect requires a modification to the forward-backward algorithm for the computation of the probabilities of seeing the observed state sequences. Under the traditional forward-backward algorithm, if the predicted sequence of authors is $Y_1, Y_2, ..., Y_T$ and the underlying sequence of authors is $X_1, X_2, ..., X_T$, the algorithm computes $\alpha_i(t) = P(Y_1 = y_1, ..., Y_t = y_t, X_t = i | A, B, \pi)$ and $\beta_i(t) = P(Y_{t+1} = y_{t+1}, ..., Y_T = y_T | X_t = i, A, B, \pi)$. These values would be recursively computed as:

$$\begin{aligned}
\alpha_i(1) &= \pi_i B_{i,y_1} \\
\alpha_i(t+1) &= B_{i,y_{t+1}} \sum_{j=1}^{N} \alpha_j(t) A_{j,i} \\
\beta_i(T) &= 1 \\
\beta_i(t) &= \sum_{j=1}^{N} \beta_j(t+1) A_{i,j} B_{j,y_{t+1}}
\end{aligned}$$

The forward-backward algorithm as stated is not able to take into account this adjacency effect, since it assumes that each observation depends only on the hidden state at that time point and not on other observations. If the model could compute a function $\textsc{Adjacent}_i(y_a, y_b)$ that returned the probability that adjacent sentences y_a and y_b were written by the same author i, then it would be possible to use the $\textsc{Adjacent}_i$ function in place of the transition model A in the forward-backward algorithm for modeling the likelihood of staying with the same author from one sentence to the next. This results in a slight modification to the recursive equations for α and β:

$$\begin{aligned}
\alpha_i(t+1) &= B_{i,y_{t+1}} \sum_{j=1}^{N} \begin{cases} \alpha_j(t) \textsc{Adjacent}_i(y_t, y_{t+1}) & \text{if } i = j \\ \alpha_j(t) A_{j,i} & \text{if } i \neq j \end{cases} \\
\beta_i(t) &= \sum_{j=1}^{N} \begin{cases} \beta_j(t+1) \textsc{Adjacent}_i(y_t, y_{t+1}) B_{j,y_{t+1}} & \text{if } i = j \\ \beta_j(t+1) A_{i,j} B_{j,y_{t+1}} & \text{if } i \neq j \end{cases}
\end{aligned}$$

It may well be the case that, for a sufficiently accurate model for the definition of ADJACENT_i, using the modified forward-backward algorithm when predicting the most likely underlying sequence of authors will result in a more accurate prediction.

3.4 Procedure

3.4.1 Single Author Attribution

A set of single authorship classification tests was performed first, to verify that the authorship classifier was able to classify documents, before attempting to classify individual sentences within those documents.

The full text of five Brontë books selected for this experiment were downloaded from Project Gutenberg as text files. After the header and footer information of each text was manually removed, the Python `nltk` library was used to tokenize each book into a list of sentences. In total, there were 6,651 sentences for Anne Brontë, 12,302 sentences for Charlotte Brontë, and 6,814 sentences for Emily Brontë. The sentences were then randomly assigned to ten distinct separations, such that ten trials could be run, each on a disjoint set of sentences.

For each separation, 30 training documents were generated: 10 for each of Anne Brontë, Charlotte Brontë, and Emily Brontë. Additionally, 50 testing documents for each author were generated, for which authorship labels were not provided to the classifier. The document generation process for synthesizing a document by a particular author involved repeatedly choosing a sentence by that author at random and without replacement, for 20 iterations. Thus, each document contained 20 randomly chosen sentences from the author.

For each document, the following features were computed:

- Non-Redacted n-grams, $n \leq 5$: These are word n-grams over the unmodified text of the original document. This includes unigrams, bigrams, as well as other n-grams for $n = 3, 4, 5$. An n-gram needed to appear at least 5 times across the data set for the separation to be included as a feature. The feature value itself, for each n-gram, is the ratio of the number of times the particular n-gram shows up in the document compared to the total number of n-grams in the document.

- Redacted n-grams, $n \leq 5$: This feature is the same as the non-redacted version, except a preprocessing step took place before n-grams were computed. The Corpus of Contemporary American English publishes a list of the top 5000 most frequently used words, and any words not found in the list were replaced in a redaction process with the token UNKNOWN. The redacted text was then processed by computing n-grams for $n \leq 5$.

- Redacted LOPS n-grams, $n \leq 5$: This feature extracts n-grams from the redacted text, as with the standard n-grams, but also preserves the syntactic structure of each n-gram, such that two n-grams that contain the same sequence of words but those words obey different constituency structures are treated as distinct features.

- POS n-grams, $n \leq 5$: These are n-grams over the part-of-speech tags of the document. First, part-of-speech tagging was performed on each document by `nltk`. The resulting sequence of part-of-speech tags was used to compute n-grams for $n \leq 5$.

- LOPS POS n-grams, $n \leq 5$: These are n-grams over the part-of-speech tags of the document that also preserve the syntactic structure of the n-gram as part of the feature.

- CCG n-grams, $n \leq 5$: These are n-grams over the CCG tags of the sentence. First, the `depccg` parser was used to take each document and parse it using a CCG (Yoshikawa, Noji, and Matsumoto 2017). The CCG tags for each token were then extracted from left to right into a sequence. This feature represents n-grams over the resulting sequence.

- Function Words: O'Shea, Bandar, and Crockett (2012) compiled a list of 264 function words, which was later expanded into a list of 277 English function words. The expanded list of 277 function words was used to compute, for each of the function words, the proportion of words in the entire document that are the function word in question.

- CPFWS: This feature represents constituency-preserving function word subtrees. Here, the expanded list of 277 function words from O'Shea, Bandar, and Crockett (2012) is also used, but the feature in question is the subtree of the original syntax tree of the sentence (as generated by the Stanford CoreNLP constituency parser) that contains only the function words, with non-function word nodes removed.

For each feature, a Naive Bayes classifier was trained on each of the training documents in the separation — for each document, both the author of the document and the result of the feature vector computation were given to the classifier. Subsequently, the classifier was provided with each of the unlabeled testing documents, and the classifier predicted the most likely author given the training data. The accuracy of each trial was measured by taking the percentage of documents that were correctly labeled. For each feature, ten trials were performed across each of the separations.

3.4.2 Multiple Author Attribution

Multiple authorship attribution was tested using the same set of original corpus texts. Again, each document was tokenized into sentences using `nltk`, and all sentences were randomly split into 10 disjoint separations.

For each separation, a randomly selected 80% of the sentences were used as training data. The remaining 20% of sentences were used to generate documents. The generation of documents used the Markov model depicted in Figure 4, where after each sentence, the model would select another random sentence by that same author with probability $s = 0.95$, and terminate with probability $t = 0.01$, assuming the document had reached the threshold length of 50 sentences. Finally, with probability 0.04, the model would uniformly at random choose a different author to switch to.

Each separation was tested with a number of feature sets, both independently and in combination. The feature sets tested were the following:

- Non-Redacted n-grams, $n \leq 3$ and $n \leq 5$
- Non-Redacted LOPS n-grams, $n \leq 3$ and $n \leq 5$
- Redacted n-grams, $n \leq 3$ and $n \leq 5$
- Redacted LOPS n-grams, $n \leq 3$ and $n \leq 5$
- POS n-grams, $n \leq 3$ and $n \leq 5$
- LOPS POS n-grams, $n \leq 3$ and $n \leq 5$
- CCG n-grams, $n \leq 3$ and $n \leq 5$
- Function Words
- CPFWS

- Redacted n-grams and Redacted LOPS n-grams, $n \leq 3$ and $n \leq 5$
- Redacted n-grams and Function Words, $n \leq 5$
- Redacted n-grams and CPFWS, $n \leq 5$
- Redacted LOPS n-grams and Function Words, $n \leq 5$
- Redacted LOPS n-grams and CPFWS, $n \leq 5$
- POS n-grams and LOPS POS n-grams, $n \leq 5$
- POS n-grams and CCG n-grams, $n \leq 5$

For each feature set, a Naive Bayes classifier was trained on each of the training sentences. For feature sets that included a combination of features, feature vectors were obtained by concatenating feature vectors for individual features. Once the classifier was fitted, it was used to predict, at a sentence-level, the most likely author for each sentence for each of the 10 testing documents.

The result of this process was an overall sentence-level prediction for the author of each sentence in the document. The next step was to use this data to predict the actual section boundaries: this was done by running the Baum-Welch algorithm until convergence on an initially random set of parameters and treating the authorship process as a Hidden Markov Model. The results were then evaluated in two ways: the proportion of sentence-level predictions that were correct based upon the results of the classifier, and the proportion of sentence-level predictions that were correct after the results were adjusted to account for authorship boundary detection.

3.4.3 Adjacency Effect Authorship Attribution

Finally, a series of separate tests was performed to assess whether modeling sentence adjacency could improve the accuracy of authorship predictions in a multi-author setting. In the original multi-author test, documents were synthesized by randomly choosing sentences from an author corresponding to the current state of a Markov chain. For this test, documents required that within a section of a document, adjacent sentences in the synthesized document were also adjacent sentences in the original text from the corpus.

First, each original corpus text was segmented into sections of 4 to 10 consecutive sentences. The sections were distributed into 10 disjoint separations. Within each separation, for each feature, 80% of the sections were used as training data and the remaining 20% were used as testing data. Ten testing documents were generated by iteratively selecting a section at random to add to each document, until all testing sections had been utilized.

The Naive Bayes classifier was trained on all of the training sentences. In addition, a logistic regression model was used to compute a function ADJACENT_i to estimate, given two adjacent sentences, the probability that they were written by the same author. Since this model may differ based on the choice of author, a separate ADJACENT_i was computed for each author. Three different choices of independent variables were used to create three different models for ADJACENT_i and compare their performances:

- Unigrams: The first test computed, given two sentences, the ratio of distinct unigrams the two sentences had in common to the number of total distinct unigrams in the two sentences. This is based on the hypothesis that adjacent sentences written by the same author are more likely to have more unigrams in common.

- Unigrams and Bigrams: The second test computed, given two sentences, the ratio of distinct bigrams the two sentences had in common to the number of total distinct bigrams. It also included the unigram ratio as a second independent variable.

- Oracle: For a point of comparison, the third test used an oracle in place of ADJACENT_i, where the function is defined to have $\text{ADJACENT}_i(S_j, S_k) = 1$ when $\text{AUTHOR}(S_j) = \text{AUTHOR}(S_k)$ and is defined to be 0 otherwise. This model knows with certainty whether two adjacent sentences are written by the same author, and can thus be used to test to estimate how much error is due to an insufficiently accurate adjacency model.

All of the adjacent sentences for a particular author were used as positive examples used in the linear regression, and an equal number of pairs of sentences where the second sentence was by a different author chosen at random were used as negative examples for the regression.

Sentence-level predictions were generated for each sentence in the testing documents using the Bayes classifier, and the Baum-Welch algorithm was then run twice on the resulting sequence: once using the standard implementation of the algorithm,

and once using the modification to the Baum-Welch algorithm as described in subsection 3.3.3. The result was two distinct sequences of section-level authorship predictions, one taking into account the adjacency model and one that did not take into account the adjacency model. The percentage of sentences correctly attributed was measured for each result as an evaluation metric.

This process was repeated twice for two different choices of features: redacted n-grams for $n \leq 3$, and POS n-grams for $n \leq 3$. Each choice of feature was tested using all three adjacency models: unigrams, unigrams and bigrams, and the oracle. For each choice of feature and adjacency model, and for each of the 10 disjoint sets of data, 10 trials were run on 10 documents each.

Chapter 4

Results and Discussion

4.1 Results of Single Authorship Attribution

Of all the feature sets tested, the non-redacted n-grams were most effective at attributing authorship, correctly predicting the document's author in 99.4% of cases. This feature set, along with all of the other n-gram feature sets used for single author attribution, considered n-grams for $1 \leq n \leq 5$. However, the non-redacted n-gram features take into account rare words that may be document-specific — if an uncommon word appeared repeatedly throughout a document, then the non-redacted n-grams would capture that information and be able to use it to attribute authorship, in effect overfitting on the training data provided.

Thus, a more accurate test of capturing linguistic style is to ignore especially rare words and only focus on common words. The redacted n-gram test did that by replacing any of the words not found in the top 5,000 English words with the token UNKNOWN. All of the original corpus texts have more than 7,500 distinct words, and all but *Agnes Grey* have more than 10,000 distinct words. Training the classifier on the redacted n-grams resulted in an average authorship attribution accuracy of 96.3%, a lower ($p < 0.01$) average attribution accuracy compared to the non-redacted n-grams, which was to be expected. Using the LOPS syntactic n-grams resulted in a slightly higher accuracy, though not significantly so.

The authorship attribution system was also tested on feature sets that did not include any of the words. The part-of-speech n-grams, for instance, considered only n-grams over the part-of-speech tags of each of the words, rather than the words themselves. This has two potential advantages: first, fitting more on truly stylistic aspects of an author's writing as opposed to words that might just be more common in the corpus of data used; and second, identifying similar structures of phrases that might be

composed of different words. However, the part-of-speech n-grams also have the disadvantage of containing less information: while the part-of-speech tags can distinguish between singular nouns and plural nouns, they can't distinguish any particular singular noun from another, for instance. The result is that the accuracy, while still much better than random, is lower than that of the n-grams over words in the document. In this experiment, the part-of-speech n-grams were accurate 75.8% of the time. The corresponding syntactic part-of-speech n-grams were slightly, but not significantly, less accurate.

Similar in spirit to the part-of-speech n-grams were the CCG n-grams, which attempted to use n-grams over CCG tags as a measure of linguistic style. The CCG tags predicted authorship of documents with an average accuracy of 40.7% — significantly better than would be expected from random chance ($p < 0.001$), but also significantly worse than part-of-speech tags provided ($p < 0.001$).

Additionally, two feature sets were tested that utilized function words as their basis for attributing authorship. Using purely the distribution of 277 function words as the choice of feature set, function words alone were able to predict authorship in 42.5% of documents. Using constituency-preserving function word subtrees, taking the same set of 277 function words and using their syntactic structure as a feature set, the accuracy of document attribution was 46.3%.

The results of the single-author attribution tests are summarized in Table 2.

4.2 Results of Multiple Authorship Attribution

For each of the feature sets for assessing the effectiveness of multiple author attribution, two accuracy values were computed. First, the sentence prediction accuracy number represents the percentage of sentences that were correctly attributed to their author by the sentence-level classifier. Second, the composite prediction accuracy reflects the number of sentences correctly attributed to their author after running the sentence-level predictions through the Baum-Welch algorithm to estimate the parameters of the authorship model and predict the most likely sequence of underlying authors of the entire document taken as a composite whole.

As was the case with single author attribution, the most accurate features for multiple author attribution were the non-redacted n-grams. These were tested when computing n-grams using $n \leq 5$, and also when using only unigrams, bigrams, and trigrams; for each choice of n, the feature set was tested with standard lexical n-grams and also

Feature Set	Average Accuracy	Standard Deviation
Non-Redacted n-grams	0.9940	0.0049
Redacted n-grams	0.9627	0.0299
LOPS Redacted n-grams	0.9660	0.0192
POS n-grams	0.7580	0.0515
LOPS POS n-grams	0.7240	0.0511
CCG n-grams	0.4067	0.0346
Function Words	0.4247	0.0322
CPFWS	0.4633	0.0497

TABLE 2: Accuracy of various feature sets in authorship attribution, for traditional single author attribution. Documents of twenty consecutive sentences were extracted from the corpus, and the model was trained on 10 training documents per author. Each testing sample contained 150 documents, and the results above are the average of 10 trials with disjoint sets of data. For all n-grams, $n = 1, 2, 3, 4, 5$ were used.

LOPS n-grams. Non-redacted LOPS n-grams where $n \leq 5$ had the highest sentence accuracy, correctly identifying the author for 68.7% of sentences; after computing composite predictions, the prediction accuracy increased to 80.8% on average. The best composite accuracy, though, was achieved by the non-syntactic version of the same feature set, which initially attributed 66.2% of sentences correctly for sentence-level predictions alone, but attributed 86.5% of sentences correctly on average after taking composite-level predictions.

Of these four versions of the non-redacted n-grams, there was a statistically significant higher sentence accuracy with the syntactic n-grams where $n \leq 5$ ($p < 0.05$) compared to the sentence-level accuracy of the non-syntactic n-grams where $n \leq 3$.

As expected, redacted n-grams that replace uncommon words with the UNKNOWN token tend to perform worse (in the 50% to 57% sentence accuracy range as compared to the 62% to 68% sentence accuracy range). Again, however, the LOPS syntactic n-grams over the redacted text for $n \leq 5$ performed significantly better at sentence-level accuracy than the non-syntactic equivalent feature, correctly predicting 56.0% of sentences on average compared to 50.1% of sentences ($p < 0.05$).

In all of the tests with both non-redacted n-grams and redacted n-grams, the composite accuracy after running the Baum-Welch algorithm was higher on average than

the sentence-level predictions, but with a much higher variance. In many cases, predicting the parameters and then predicting the most likely authors resulted in accuracy comparable to that of single-author attribution: in one trial, 99.3% of sentences ended up being attributed correctly, and accuracies over 90% were not uncommon. But also common were trials where the model would converge on the wrong parameters, and actually score worse than the sentence-level predictions.

Unlike the case in single-author attribution, using n-grams over part-of-speech tags did not perform significantly differently from redacted n-grams for $n \leq 3$: part-of-speech features correctly predicted 53.5% of sentence authorships at the sentence-level, and correctly predicted 62.0% of sentence authorships after running the Baum-Welch algorithm. Using syntactic POS-grams did not perform significantly better or worse.

As was the case with single authorship attribution, CCG n-grams performed significantly worse than all other features, averaging 38.8% accurate in sentence-level predictions. Function words features alone correctly attributed 44.2% of sentences on average, and function word embedded subtrees correctly attributed 46.0% of sentences on average.

For feature sets that tended to be less accurate — with sentence-level prediction accuracies of less than 0.5, for instance — drawing composite-level predictions were generally not significantly better, and in some cases were worse. It seems reasonable, then, that a sufficient threshold of sentence-level predictive accuracy is required for the maximum likelihood estimation to prove effective, since otherwise the high frequency of incorrect sentence predictions is likely to hinder the process of converging on the true underlying sequence of authors.

A summary of the results of authorship attribution on multi-author documents is shown in Table 3.

In addition to testing features individually, several feature combinators were also used to test the effect of using multiple features simultaneously in predicting authorship. Adding redacted syntactic n-grams to redacted non-syntactic n-grams for $n \leq 5$ resulted in a slight but significant improvement from 50.9% of sentences accurate to 54.7% of sentences accurate ($p < 0.05$). In other cases, however, the feature combinators did not perform significantly better on sentence-level predictions than the most accurate of the features in isolation. When considering composite-level predictions after running Baum Welch, the combination of redacted LOPS n-grams and function word subtrees predicted 69.7% of sentence authors correctly on average, significantly better than both redacted LOPS n-grams alone ($p < 0.05$) and better than the combination of

	Sentence Predictions		**Composite Predictions**	
Feature Set	**Accuracy**	**Std. Dev**	**Accuracy**	**Std. Dev**
Non-Redacted n-grams, $n \leq 3$	0.6245	0.0528	0.7101	0.1596
Non-Redacted n-grams, $n \leq 5$	0.6510	0.0595	0.8651	0.0891
Non-Redacted LOPS n-grams, $n \leq 3$	0.6616	0.0623	0.7093	0.1474
Non-Redacted LOPS n-grams, $n \leq 5$	0.6864	0.0561	0.8084	0.1925
Redacted n-grams, $n \leq 3$	0.5570	0.0431	0.5593	0.1351
Redacted n-grams, $n \leq 5$	0.5086	0.0522	0.5751	0.1143
Redacted LOPS n-grams, $n \leq 3$	0.5419	0.0450	0.5838	0.1348
Redacted LOPS n-grams, $n \leq 5$	0.5601	0.0598	0.5610	0.1383
POS n-grams, $n \leq 3$	0.5355	0.0585	0.6201	0.1805
POS n-grams, $n \leq 5$	0.5352	0.0783	0.6153	0.1786
LOPS POS n-grams, $n \leq 3$	0.5300	0.0502	0.6224	0.1665
LOPS POS n-grams, $n \leq 5$	0.4828	0.0918	0.4234	0.1446
CCG n-grams, $n \leq 3$	0.3880	0.1322	0.4087	0.1970
CCG n-grams, $n \leq 5$	0.3584	0.0608	0.3512	0.0627
Function Words	0.4421	0.1197	0.4949	0.1399
CPFWS	0.4601	0.0732	0.4458	0.0722

TABLE 3: Accuracy of various feature sets in authorship attribution, for multiple author attribution. The results above are the average of 10 trials with disjoint sets of data.

redacted LOPS n-grams and function words ($p < 0.05$).

A summary of the results of authorship attribution on multi-author documents when using a variety of different feature combinators is shown in Table 4.

4.3 Results of Accounting for Adjacent Sentences

When the adjacency model was used to modify the Baum-Welch algorithm to take advantage of the dependence of adjacent sentences, the accuracy for both word adjacency models and for the n-gram word and bigram adjacency model was not significantly different than the accuracy from using the standard Baum-Welch algorithm. Moreover, for the word and bigram adjacency model for part-of-speech n-grams, the adjacency model actually performed significantly worse ($p < 0.05$).

	Sentence Predictions		**Composite Predictions**	
Feature Set	**Accuracy**	**Std. Dev**	**Accuracy**	**Std. Dev**
Redacted n-gram $n \leq 3$ Redacted LOPS n-gram $n \leq 3$	0.5235	0.0540	0.5917	0.1484
Redacted n-gram $n \leq 5$ Redacted LOPS n-gram $n \leq 5$	0.5469	0.0332	0.6188	0.1723
Redacted n-gram $n \leq 5$ Function Words	0.5561	0.0370	0.6021	0.2132
Redacted n-gram $n \leq 5$ CPFWS	0.5212	0.0379	0.5992	0.1399
Redacted LOPS n-gram $n \leq 5$ Function Words	0.5339	0.0672	0.5346	0.1749
Redacted LOPS n-gram $n \leq 5$ CPFWS	0.5407	0.0472	0.6967	0.1458
POS n-gram $n \leq 5$ LOPS POS n-gram $n \leq 5$	0.4730	0.0804	0.4272	0.1576
POS n-gram $n \leq 5$ CCG n-gram $n \leq 5$	0.4993	0.0612	0.5679	0.1748

TABLE 4: Accuracy of various combinations of feature sets in authorship attribution, for multiple author attribution. The results above are the average of 10 trials with disjoint sets of data.

	Composite Predictions		Adjacency Predictions	
Feature Set	**Accuracy**	**Std. Dev**	**Accuracy**	**Std. Dev**
Redacted n-gram $n \leq 3$ Word Adjacency Model	0.6459	0.0642	0.6310	0.0547
Redacted n-gram $n \leq 3$ Word and Bigram Adjacency Model	0.6301	0.0633	0.6004	0.0586
Redacted n-gram $n \leq 3$ Oracle Adjacency Model	0.6270	0.0481	0.7138	0.0407
POS-gram $n \leq 3$ Word Adjacency Model	0.6513	0.0749	0.6494	0.0495
POS-gram $n \leq 3$ Word and Bigram Adjacency Model	0.6872	0.0458	0.6335	0.0629
POS-gram $n \leq 3$ Oracle Adjacency Model	0.4897	0.0720	0.5685	0.0577

TABLE 5: Accuracy of multiple authorship predictions for standard Baum-Welch section boundary prediction, as well as taking advantage of the adjacency model to detect when two adjacent sentences are written by the same author.

However, when the adjacency model was replaced with an oracle that could perfectly predict whether two adjacent sentences were written by the same author, the accuracy of the n-gram based authorship attribution improved from 62.7% of sentences correctly attributed to 71.4%, a significant difference ($p < 0.001$).

A full summary of these results is shown in Table 5.

This suggests that, for a sufficiently accurate model of adjacent sentences, such as the one proposed in Samardzhiev, Gargett, and Bollegala (2017), taking advantage of the dependence of adjacent sentences can help to improve the accuracy of authorship attribution in a multiple author setting, but that simple adjacency models that examine just the number of words two sentences have in common are insufficiently accurate for such tasks.

Chapter 5

Extensions and Future Work

5.1 Features and Models

A number of features tested took advantage of the syntactic structure of the authors' use of language, but there are likely other stylometric features — both syntactic and non-syntactic — that are worth testing to see how they perform in a multiple authorship setting. Stamatatos (2009) suggested features including vocabulary richness, context free grammar rewrite rules, and semantic dependencies as features that have shown a degree of effectiveness for attributing authorship in single-author cases, and it is worth investigating how those features compare in effectiveness with the features tested here, particularly in multiple authorship settings.

Areas for additional research also include testing these models on texts from different domains: I focus predominantly on literature, but authorship attribution also has applications to historical documents, journalism, forensic linguistics, and other fields in which identifying the author of a text may be valuable. Experimenting on documents of other domains, or even other works of literature by other authors, is another area for further research.

Potential further investigations also include improving the process of detecting section boundaries between authors based on sentence-level predictions. The current method of Baum-Welch expectation maximization results in improved authorship attribution accuracy on average, but comes at the cost of high variance. The model frequently would settle on a sequence that was worse than the original prediction, potentially a result of the algorithm settling at a local maximum instead of finding the global maximum.

There were also a few cases where the modified Baum-Welch algorithm appeared to hang during the process of estimating parameters. In these cases, the trial was terminated and restarted, but those cases suggest that the modified Baum-Welch algorithm

may not have a guarantee of convergence. Refining the modified algorithm to better handle models of adjacent sentences is therefore also a potential area for further exploration.

5.2 Evaluation Metrics

In order to evaluate the performance of the authorship attribution system, there must be a way to evaluate the segment predictions. The evaluation metric employed in this experiments was to, after performing the section-level predictions via application of the Baum-Welch algorithm, compute the number of sentences which were correctly attributed to the right author out of the total number of sentences in the document.

This strategy, however, does not quite capture the accuracy of the segmentation. In particular, this evaluation metric over-punishes boundary errors that are close to the actual boundaries, and under-punishes boundary errors that are further away. A different evaluation metric that is able to capture this distinction may offer more insight into which feature sets are actually better able to predict section boundaries and section authorship.

One strategy is to employ an algorithm proposed by Hearst (1994), in which the accuracy of a segmentation is measured along two axes: precision (the number of boundaries selected that are really boundaries) and recall (the number of true boundaries that are found), within a certain margin of error (tested at both no error, and at an error of one paragraph too early or too late). This accounts for the sort of discrepancy in segmentation described in Figure 10. However, Hearst's evaluation metric was designed only to test the accuracy of drawing the boundaries, and not for the assignment of the sections between those boundaries to a label. Potential areas for further work would include modifying the metric to handle assessing the accuracy of a labeled section, and exploring other metrics that could be used to evaluate the effectiveness of section-level authorship predictions.

5.3 Alternative Approaches to Modeling Multiple Authorship

One could also imagine an entirely different approach to solving the multiple authorship problem, by inverting the order of the steps that are taken in approach. Whereas I chose the route of performing the sentence-level prediction step first, and then using the results of those predictions to compute sections boundaries and section-level

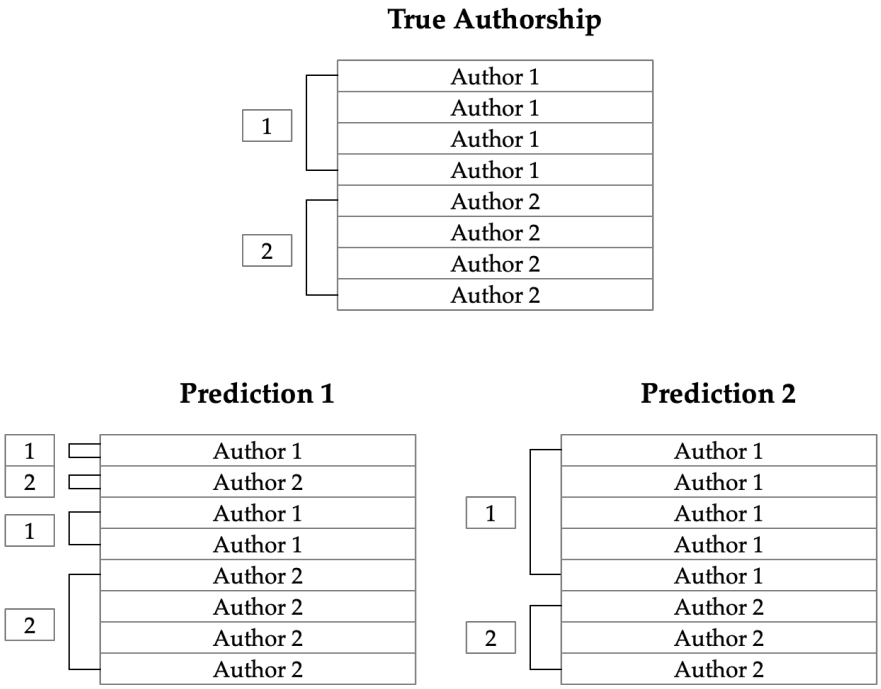

FIGURE 10: Though both Prediction 1 and Prediction 2 guess the same number of sentences correctly (7 out of the 8), Prediction 2 should generally be considered the better prediction.

authorship predictions, it is also reasonable to imagine an approach in which the first step of the process is to identify likely section boundaries, and then to use more conventional authorship attribution methods to determine the likely author for each of the sections. Hearst (1994) has proposed the TextTiling algorithm for partitioning a document into sections based upon the topic structure of the document overall. Though Hearst's algorithm itself doesn't necessarily map directly to the authorship problem, since in a real-world setting, multiple authors may collaborate on a section of a document on the same topic, it is conceivable that a variant of the algorithm could also be used to determine likely authorship boundaries. The result would then be a set of predicted single-author sections, on which standard single-author attribution systems could be used.

It would also be interesting and practical to approach the problem by testing real-world data. To do so would require getting access to a labeled data set of documents for which each sentence has an author label. It may be possible that data can be extracted from online collaborative writing platforms — such as Google Docs — to determine such sentence labels, but such processes would likely require a different prediction model since there is no guarantee on such platforms that each sentence will be written by a single author only.

Chapter 6

Conclusion

The goal of this authorship attribution system has been to establish a model for predicting authorship of multi-author documents, and present a variety of stylometric features — some drawn from the literature, and some novel — for use in authorship classification.

First, a formulation of the multiple authorship attribution problem — which has very rarely been explored in the authorship attribution literature — is presented. The formulation of the problem is based on the assumption of single-author sentences, and the conjecture that in multi-author documents, single authors are likely to write multiple sentences in sequence. This section-based approach allowed for a novel formulation of the process of performing authorship attribution in a multiple author setting: by attempting to predict boundaries between sections, and predicting the most likely author for each of the sections. A number of instances throughout history where multiple authorship of this form or of a similar form are explored, for which a multiple authorship attribution model may prove to be useful.

In addition, a model is presented for moving from sentence-level predictions to section-level predictions in a multi-author setting. In particular, a Hidden Markov Model-based approach is proposed, where the true author of each sentence is treated as an unknown hidden state, and the observed sentences and their corresponding predicted authors are treated as the observed emission states. From there, the Baum-Welch algorithm can be applied to refine the original sentence-level prediction to produce a new prediction based on estimating the most likely sequence of underlying states. In additional, a proposal is made for modifying the Baum-Welch algorithm for taking into account the dependency that adjacent sentences have on one another.

A number of novel features are proposed to tackle the multiple authorship attribution problem. These features include the use of a combinatory categorial grammar

parser to extract n-grams over CCG tags, the extraction of constituency-preserving function word subtrees to capture a sentence's use of various different function words while also capturing the syntactic relationship in the usage of function words, and a variant on syntactic n-grams that are able to capture both the syntactic structure of a sequence of tokens while also preserving the lexical order of words from the original sentence.

Experimental results suggest that multiple authorship attribution can predict individual sentence authorship between the three Brontë sisters correctly up to 68.7% of the time on average, and up to 86.5% of the time after running the Baum-Welch algorithm to predict the most likely sequence of underlying authors. Even ignoring the actual words used and instead using part-of-speech tags to avoid over-fitting on the training data, the most likely sequence of underlying authors can be computed to predict sentences with 62.2% accuracy on average. Additionally, in certain cases, adding syntactic structure to n-grams of text and using the constituency-preserving function word subtrees instead of just function words alone were shown to offer statistically significant improvements in prediction accuracy.

Bibliography

Adair, Douglass (1944). "The Authorship of the Disputed Federalist Papers". *The William and Mary Quarterly* 1.2, pp. 97–122.

Althoff, Tim, Denny Britz, and Zifei Shan (2014). "Authorship Attribution in Multi-author Documents". *Department of Computer Science, Stanford University*.

Brennan, Michael, Sadia Afroz, and Rachel Greenstadt (2011). "Adversarial Stylometry: Circumventing Authorship Recognition to Preserve Privacy and Anonymity". *ACM Transactions on Information and System Security* 15.3, pp. 1–22.

Chaski, Carole E. (2012). "Author Identification in the Forensic Setting". *Oxford Handbook of Language and Law*, pp. 489–503.

Chung, Cindy K. and James W. Pennebaker (2007). "The Psychological Functions of Function Words". *Social Communication*, pp. 343–359.

Covington, Michael A. (2001). "A Fundamental Algorithm for Dependency Parsing". *Proceedings of the 39th Annual ACM Southeast Conference*.

Dao, James (2018). "How the Anonymous Op-Ed Came to Be". *The New York Times*.

El Bouanani, Sara El Manar and Ismail Kassou (2014). "Authorship Analysis Studies: A Survey". *International Journal of Computer Applications* 86.12, pp. 22–29.

Feng, Song, Ritwik Banerjee, and Yejin Choi (2012). "Syntactic Stylometry for Deception Detection". *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 171–175.

Gamon, Michael (2004). "Linguistic correlates of style: authorship classification with deep linguistic analysis features".

Hearst, Marti A. (1994). "Multi-Paragraph Segmentation of Expository Text". *ACM*.

Howedi, Fatma and Masnizah Mohd (2014). "Text Classification for Authorship Attribution Using Naive Bayes Classifier with Limited Training Data". *IISTE Computer Engineering and Intelligent Systems* 5.4, pp. 48–56.

Kestemont, Mike (2014). "Function Words in Authorship Attribution From Black Magic to Theory?" *Proceedings of the 3rd Workshop on Computational Linguistics for Literature*, pp. 59–66.

Kestemont, Mike, Sara Moens, and Jeroen Deploige (2015). "Collaborative authorship in the twelfth century: A stylometric study of Hildegard of Bingen and Guibert of Gembloux". *Literary and Linguistic Computing* 30 (2), pp. 119–224.

Kim, Sangkyum et al. (2011). "Authorship Classification: A Discriminative Syntactic Tree Mining Approach". *Special Interest Group on Information Retrieval*.

Klein, Dan and Christopher D. Manning (2003). "Accurate Unlexicalized Parsing". *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 423–430.

Mack, Justin L. (2018). "Mike Pence, lodestar and the New York Times op-ed: What we know". *Indy Star*.

Madison, James (1787). "The Same Subject Continued: The Union as a Safeguard Against Domestic Faction and Insurrection". *The Federalist Papers* 10.

Mosteller, Frederick and David L. Wallace (1963). "Inference in an Authorship Problem". *Journal of the American Statistical Association* 58.302, pp. 275–309.

O'Shea, James, Zuhair Bandar, and Keeley Crockett (2012). "A Multi-Classifier Approach to Dialogue Act Classification Using Function Words". *Transactions on Collective Computational Intelligence* 7.

Rudman, Joseph (2012). "The Twelve Disputed 'Federalist' Papers: A Case for Collaboration". *Digital Humanities 2012*.

Samardzhiev, Krasen, Andrew Gargett, and Danushka Bollegala (2017). "Learning Neural Word Salience Scores".

Shahan, Thomas (1908). "Caroline Books (Libri Carolini)". *The Catholic Encyclopedia* 3.

Sidorov, Grigori et al. (2012). "Syntactic N-grams as Machine Learning Features for Natural Language Processing". *Mexican International Conference on Artificial Intelligence*.

Stamatatos, Efstathios (2009). "A Survey of Modern Authorship Attribution Methods". *Journal of the American Society for Information Science and Technology* 60.3, pp. 538–556.

Steedman, Mark (1996). "A Very Short Introduction to CCG".

Stratonovich, R.L. (1960). "Conditional Markov Processes". *Theory of Probability and its Applications* 5.2, pp. 156–178.

Tschuggnall, Michael and Gunther Specht (2014). "Enhancing Authorship Attribution By Utilizing Syntax Tree Profiles". *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 195–199.

Wallach, Luitpold, Antony Augoustakis, and Barbara Wallach (2017). "Alcuin's Authorship of the Libri Carolini: Theodulfian Fictions and Elective Affinities". *Illinois Classical Studies* 42.2, pp. 279–317.

Wennberg, Victor (2012). "A Stuctural Approach to Authorship Attribution using Dependency Grammars".

Yoshikawa, Masashi, Hiroshi Noji, and Yuji Matsumoto (2017). "A* CCG Parsing with a Supertag and Dependency Factored Model". *Proc. ACL*.