C-SALT: CONVERSATIONAL STYLE ATTRIBUTION GIVEN LEGISLATIVE TRANSCRIPTIONS

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Garrett Summers

June 2016

COMMITTEE MEMBERSHIP

TITLE:	C-SALT: Conversational Style Attribution given Legislative Transcriptions
AUTHOR:	Garrett Summers
DATE SUBMITTED:	June 2016
COMMITTEE CHAIR:	Professor Foaad Khosmood, Ph.D. Assistant Professor of Computer Science
COMMITTEE MEMBER:	Professor Alexander Dekhtyar, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Professor Franz Kurfess, Ph.D. Professor of Computer Science

ABSTRACT

C-SALT: Conversational Style Attribution given Legislative Transcriptions

Garrett Summers

Common authorship attribution is well described by various authors summed up in Jacques Savoy's work. Namely, authorship attribution is the process "whereby the author of a given text must be determined based on text samples written by known authors [48]." The field of authorship attribution has been explored in various contexts. Most of these works have been done on the authors written text. This work seeks to approach a similar field to authorship attribution. We seek to attribute not a given author to a work based on style, but a style itself that is used by a group of people. Our work classifies an author into a category based off the spoken dialogue they have said, not text they have written down. Using this system, we differentiate California State Legislators from other entities in a hearing. This is done using audio transcripts of the hearing in question. As this is not Authorship Attribution, the work can better be described as "Conversational Style Attribution". Used as a tool in speaker identification classifiers, we were able to increase the accuracy of audio recognition by 50.9%, and facial recognition by 51.6%. These results show that our research into Conversational Style Attribution provides a significant benefit to the speaker identification process.

ACKNOWLEDGMENTS

Thanks to:

- I give my thanks to my committee members for taking the time to review my work.
- I would like to thank Foaad Khosmood for advising me throughout my research, and for sticking with me through the various experiments we tried that didn't pan out.
- Finally, I would like to thank my parents and brother for supporting me throughout my entire school career, and for encouraging me to keep working through difficult challenges.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Overview

Information about occurrences in our world can be recorded in many different ways. A person can transcribe what is occurring onto paper. Pictures can be taken of a scene or an entire event can be filmed. But will all relevant information be retained through this recording of data? At the time of recording, what information is relevant is unknown. The relative time of the event could be lost, or possibly the location it occurred at. The amount of information that the world has available is immense, and extracting as much data as we can out of what is available can be a challenge.

There are many different kinds of meta-data that can be taken from any specific event. This thesis focuses on being able to recover meta-data that is unavailable due to the nature of how an event was recorded. In particular, we try to gain knowledge of the identity of a speaker from a video recording. There are several ways that a speaker might be identified through such a medium. The first that comes to mind for most people is either voice or facial recognition. There is a third type, however, which is normally not thought of, but can possibly help to improve the first two methods classification ability. Namely, the comparison of the words, or textual vocabulary, a speaker uses. We will focus on this third category of classification. Given the textual transcription of recorded audio from a video, we attempt to classify an unidentified speaker into a specific category. In terms of methodology, we adopt a stylistic approach to identification as opposed to a heavily statistical method. David Holmes states, "we may define style as a set of measurable patterns which may be unique to an author." [17] His works discuss using the number of different words in a text, the

richness of the vocabulary, and use of "filler" words as stylistic identifiers.

Moshe et al. describes the simplest type of authorship attribution as, "the one in which we are given a small, closed set of candidate authors and are asked to attribute an anonymous text to one of them." [26] Many times, there are lots of text associated with each author. This type of attribution can be seen in various works such as Khosmood's work in neural network authorship attribution [23]. If there are only a few possible people that could be in a given recording, then the analysis would be much simpler. What happens when the domain of people is increased? When there are possibly hundreds of people that could be present in an audio recording, how can we identify them or classify them? Luyckx and Daelemans explores this problem on written word authorship attribution [32]. This work, C-SALT, seeks to take the ideas of authorship attribution and apply them to a new category. Namely, the classification of speaking style of a given group. By preforming classifications this way, C-SALT can function as a tool for simplifying the process of classifying oral datasets by lessening the number of authors that need be considered.

1.2 Purpose

The research being done here is done in conjunction with a project called Digital Democracy. The goal of Digital Democracy is to provide transparency for the state government and provide access of that information to the general public, found at https://digitaldemocracy.org. To do this, they provide a searchable database of legislative information, including topics such as bills, legislators, and hearings. The project is also doing more research into information that can be extracted from legislative transcripts, such as how a legislators arguments may be affected by who is giving them money.

We can help the project achieve this goal by providing additional data that they

may not have otherwise been able to obtain through their traditional methods. Working with them also provides us with another benefit. Digital Democracy has been working on transcribing state legislator video recordings, providing access to an ample supply of test and training data on hundreds of different people and legislative hearings. All material we have gathered was obtained with permission from the Institute for Advanced Technology and Public Policy (IATPP), who run the Digital Democracy project.

Throughout this work, we use recordings of the California State Legislature as the base data we are trying to gain additional, new, previously unavailable information on. The speakers we are trying to identify are the legislators involved with government hearings that have been recorded over the last two years. As hearings do not consist of only legislators, the presence of people such as the general public and lobbyists creates a large pool of individuals. This increases the possible sources of error and adds to classification complexity. This research seeks to filter out these non-legislators to help reduce identification issues during classification. We show that as a result of this work, legislator identification will be significantly improved during classification. This is seen through the textual classifier functioning as a tool working in conjunction with facial and audio recognition to achieve an overall higher accuracy of individual speaker identification than could otherwise be achieved.

The rest of this document is organized into several sections. First, we discuss the related work this topic is involved with in chapter 3. In chapter 3, we delve into the background of the work itself. Chapter 4 discussed the goals of the research. In chapter 5 we discuss the experiment that was done in this work, followed by the results in chapter 6, and finally the conclusion in chapter 7.

Chapter 2

RELATED WORKS

This section examines what authorship attribution is, what work that has been done in the field previously, and the less common area of spoken authorship attribution that this work focuses on.

2.1 Written Authorship Attribution

The main idea behind any type of Authorship Attribution(AA) is to take features from a given text and use them to distinguish between several authors. Each author that is being evaluated has text samples that are know to have originated from them. These are what the test data in question can be compared against [47].

Savoy specifies this type of classification as, "a focus on the closed-class attribution method in which the real author is one of several possible candidates." More simply, we know the possible pool of candidates the author can be found in, and have the data collected for each of these authors to compare them against. The standard type of authorship attribution (text written by the creator) normally involves the creator of a written work. Most of the previous research done in the field has been done on written works as such.

Stamatatos' work investigating authorship attribution traces writing style attribution back to the 19th century [49]. He found the most influential work to be that of Mosteller and Wallace on the authorship of *The Federalist Papers*. Their method used a Bayesian statistical analysis of the frequencies of a small set of common words. The work marked one of the first non-traditional authorship attribution studies (compared to traditional human expert-based methods) [35]. This started the wide use

of stylometry as a way to classify writing style. Authorship attribution continues to be useful today in other written forms such as the digital medium. Chaski's work describes the use of a stylisic method for identifying people who have committed a digital crime when authorities would not otherwise be able to prove who was actually doing the typing on a specific computer [8]. Though this work will not be using the typical stylometry featurization on one specific author like the previous works, it will still explore stylometry on certain selected features in the classification process.

Depending on the amount of data available, authorship attribution works tend to get fairly accurate results. Khosmood's work, done in 2005, received up to 99% accuracy on a test corpus [23]. In fact, the five main test sets that he examined all were able to achieve results of 98% and up. Each of these datasets, however, had fairly large numbers of documents to work with and an average number of authors. This confirms what Luyckx and Daelemans discuss in their work on traditional authorship attribution [32].

Many works follow the trends that were found in Luyckx and Daelemans work. As the number of authors goes up, the accuracy of the classifier goes down. This is also true given a limited amount of training data for the given authors. This is an issue that our work will need to address to a minor degree. It is less problematic in our case than in standard authorship attribution because we focus more on categorization of each speaker, rather than the identity of the speakers themselves.

2.2 Spoken Authorship Attribution

While the general notion of authorship attribution is focused on an author's written text, this is not the only medium an author can use. With this in mind, "authorship attribution" may not be the right term to use for this work. As Khosmood describes, "source" attribution is a more accurate description of what this work is doing [23].

With speech-to-text technology, and general language transcription, the spoken word is a source that can be easily examined. The topic area could even be further extended to encompass works like paintings and sculptures. For the purpose of this work, however, we will stick with using the spoken word as the chosen source for the attribution process.

As shown by the work done by Juola and Sofko on improving authorship attribution technology, spoken word author classification does exist, but is not the most prevalent idea in the current field [21]. They discuss a competition in which there are 13 different problems in the field that need to be addressed. Only one of these falls into the field of spoken word authorship attribution - "transcripts of unrestricted speech gathered during committee meetings." Cristani et al take cues from spoken word tendencies in their work [11]. They focus on authorship attribution for chat client messages. While these aren't words that have been spoken and transcribed, their work describes chat messages as "[sharing] many aspects with spoken conversations." Their work focuses more on aspects of a conversation, such as turn-taking, but it does start to explore the area of the spoken medium which AA approaches have not taken it into account until now.

The typical method used, as we have discussed, for speaker recognition is to use the audio of the person talking. Features are extracted on characteristics in the speakers audio recording, not from the words themselves. Speaker recognition can be used for in a number of ways including identification purposes for security access or simply identifying someone in a recording (like we are researching). The work of Joseph Campbell shows the considerations of the first type, while his other work in conjunction with W.M. Campbell focuses on using a support vector machine algorithm for speaker recognition [6][7]. This algorithm is the primary focus of the voice recognition we are using in the Digital Democracy project as well.

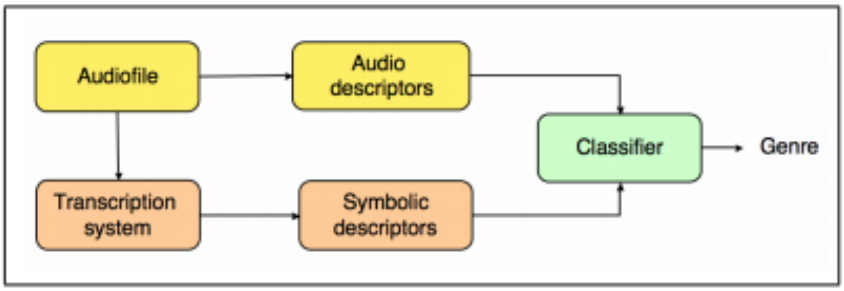

Figure 2.1: The classification process of Lidy et al. [29]

A focus on using audio in speaker recognition seems logical; a person's voice is going to be more unique than a few words they may have said. The research this paper does focuses on text as a way to increase the accuracy of voice (and facial) recognition for this reason. It is not intended as the primary source of classification, but more of an unexplored feature that can be used to improve accuracy. Instead of classifying the speakers, we will be classifying the style of a group speakers.

2.3 Style In Transcription-Based Classification

Classification of speakers based on transcribed data is be base of what this work tries to achieve. One major inspiration to classify the style of each category we are grouping people into came from outside the "author/speaker" field. Namely, research done on classifying the genre of music has shown style classification to be a valid approach. Lidy et al. show a similar classification approach to what we are trying to achieve [29].

At first, they try to classify genre using audio features, but having reached a ceiling to what they could achieve, move on to including features from something similar to text. In addition to the audio features, they used a musical transcription system to turn the audio data into symbols, and extract features from the result. Their process, shown in figure 2.1, combines all thee features into one classifier.

	Live	Replay	Others	CM
Speakers	Announcer	Announcer, Commentator	Announcer, Commentator, Reporter, etc.	Others
Length of Sentences	Short	Long	cannot be determined	cannot be determined
# of Sentences	A few	Many	cannot be determined	cannot be determined
Situational Phrases	highly likely	less likely	probably	rarely

Figure 2.2: Features used by Nitta and Babaguch [37]

Category classification based on style has also been accomplished in relation to sports as well. In particular, Nitta and Babaguch classify videos of sports broadcasts using the closed-caption text [37]. They attempt to classify each segment of scenes in the video as "Live" , "Replay", "Others" - like Report or Studio - or "Commercial Message". In terms of classification features, they used information about what type of person was talking in the video, as well as textual features such as sentence length, number of sentences, and certain situational phases as seen in figure 2.2.

They were able to achieve an average recall of 87% and a precision of 76%. This is similar to our work in that we are trying to classify who is speaking into a category of legislator or non legislator. We also have to deal with a problem similar to what this work had, errors in the transcription of audio to text. While some of the difficulty in Nitta and Babahuch's work comes from more classification categories, it has much less people that speak and contribute to the style seen in each segment. Our work takes the opposite approach, having only two categories, but many different individuals contributing their own unique style attributes to their group.

Our work can been seen as being most similar to that of Tambouratzis et al [50]. Both works focus on style classification from transcriptions in a legislative setting and speaker recognition. Several differences in the works are notable: we focus on the California State Legislature while their work examines Greek Parliament, the transcriptions we use are created from videos while theirs were written by a secretary during the session, and we focus on categorizing the speakers into groups while they

classify the text into different registers of speech.

It is important to note the method of information retrieval used in each work. As the secretary was present and recording the speech as it was said, the text being used in the research should be much more accurate than what we are able to achieve off a recoding. On top of that, the secretary is often provided with written copies of a speech a member of the parliament has given, leaving less room for error in transcription. Its also possible, however, that the speeches were "sanitized" and may not accurately represent what was said. Regardless of quality, the fact remains that both works take information recorded from oral speech. This is key in that this area is not well explored by many other works.

The work of Tambouratzis et al. focused on two different issues with their corpus. First, they focus on discriminating between registers in the Greek language itself. The registers make up the three parts of their data: a historical register, a fictional register, and the parliament register we are interested in. They take many different features from each register such as parts of speech and number of sentences to create a classifier that can distinguish between these registers. This is the style classification section of their work. They then tried to preform speaker identification on the individuals in their parliament corpus, and were successful. This information is promising for our work. While their parliament dataset only consisted of five people and large amounts of data per person, Tambouratzis et al. show that style classification in relation to speaker recognition is viable.

Chapter 3

BACKGROUND

This section describes much of the technical background related to authorship attribution and the techniques used in this work.

3.1 Technical Terms

This work makes use of some technical terms that may seem straightforward, but for clarity we describe exactly what their meaning is when referencing them here.

First is the term "utterance". The standard dictionary definition is "a spoken word, statement, or vocal sound." An utterance is defines as a vocal statement varying in length from a single word to several sentences, said by a speaker. The dataset that we use is made up entirely of utterances taken from hearings. We also use it in reference to transcribed, textual versions of the statement.

The term "feature" is used often in relation to different classifications and utterances. A feature of an utterance represents a piece of information that was extracted from the text to be used in classification. This could just be a word or it could be the entire length of the text. Most references to features represent the general category of information that is being extracted. This means that "using unigrams as a feature" is stating we will use all the unigrams taken from the data as many individual features in the classification process.

Features	
Lexical	Token-based (word length, sentence length, etc.) Vocabulary richness Word frequencies Word *n*-grams Errors
Character	Character types (letters, digits, etc.) Character *n*-grams (fixed length) Character *n*-grams (variable length) Compression methods
Syntactic	Part-of-speech (POS) Chunks Sentence and phrase structure Rewrite rules frequencies Errors
Semantic	Synonyms Semantic dependencies
	Functional
Application-specific	Structural Content-specific Language-specific

Figure 3.1: Typical stylistic features seen in authorship attribution [49].

3.2 Stylistics

In his work, Verdonk says that, "Stylistics is concerned with the study of style in language [53]." This then begs the question, what exactly is style? How can it be used to define the shape or design of something, or how something is done? Andreas Jucker defines style as what Chomsky calls "performance" and de Saussure calls "parole" [20]. Jucker states that style is a "comparative concept". In the context of this research, Holmes' work sums up the basics of Stylistics and what style is. Namely, "the stylometrist [therefore] looks for a unit of counting which translates accurately the 'style' of the text, where we may define 'style' as a set of measurable patterns which may be unique to an author [17][18]." He expresses the notion that characteristics like the number of words in the sentence and the number of different words used in the text can be analyzed as features of an authors work. He notes that even the typically unnoticed "filler words" from mainstream vocabulary can be used as features.

This idea can be further extended in a spoken word analysis to speech disfluencies - words such as "um" or "uh". Holmes further looks into characteristics like word length, sentence length, part of speech distribution, and even syllables. The work of Stamatatos surveys modern authorship attribution, and discusses several different stylistic features like these. He lays them out into several categories as shown in figure 3.1 [49]. The figure shows many different areas that can be considered for stylistic features, with this work focusing primarily on the lexical and application specific categories, with the main type of feature seen in the other categories mixed in.

- unigram (1-gram):

a	swimmer	likes	swimming	thus	he	swims

- bigram (2-gram):

a swimmer	swimmer likes	likes swimming	swimming thus	...

- trigram (3-gram):

a swimmer likes	swimmer likes swimming	likes swimming thus	...

Figure 3.2: Example of a sentence being split into different level ngrams.

3.3 N-grams

One major type of feature that can be used under the category of stylistics is n-gram extraction. N-grams are chunks of either words or letters of a given length. A unigram is chunk of data being examined that looks at one entity on its own. Bigrams look at one entity, and the second entity after that, trigrams three entities in a row, and so on. Figure 3.2 shows an example of a sentence split up into such n-grams. By splitting sentences into different levels of complexity, we can see if specific words, phrases, or word orderings are common for a given person or group. N-grams have been used in many works, [49][24][32] are examples of authorship attribution and remain a great way to classify characteristics of human speech.

3.4 Preprocessing

Several techniques can be used to increase the accuracy of the spoken language attribution process. These preprocessing techniques involve changing the text itself to be more uniform across all of the data, providing values that can be better compared to each other. Figure 3.3 shows the order that the preprocessing steps occur in. Keep in mind that not all steps in of the process may be applied in a given experiment.

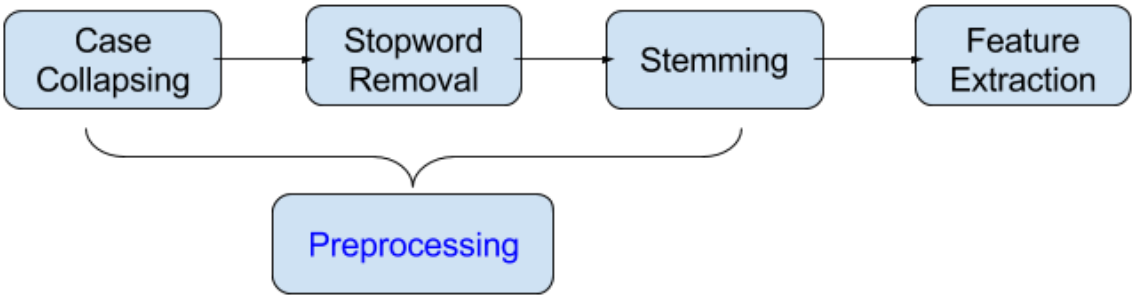

Figure 3.3: The order of preprocessing steps before feature extraction

3.4.1 Stemming

Mayfield and McNamee define stemming as, "an approximation to lemmatization in which morphological variants of a word are reduced to a single form [33]." In layman's terms, the word being stemmed is converted to a more basic form by removing suffixes such as "ing". Figure 3.4 shows an example of a sentence that has undergone the stemming process. The process makes the data more uniform in that the main root of every word is what is compared to other words, regardless of verb tense or plurality.

There are several different types of stemming algorithms available. We originally considered choosing between three of the more common stemming algorithms: the widely used Porter stemming algorithm, the Snowball algorithm (which is an improvement over the Porter algorithm) and the Lancaster algorithm which more aggressively concatenates words over the Snowball algorithm [54][38]. After initial testing, the Snowball stemming algorithm was chosen to be used for the testing of this work as it preformed better than the Porter stemming algorithm, and doesn't cut down the words as much as Lancaster stemming.

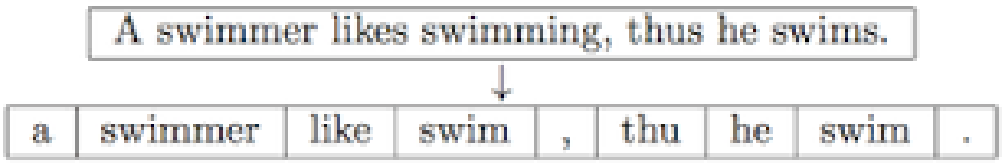

Figure 3.4: An example of a sentence that has undergone stemming

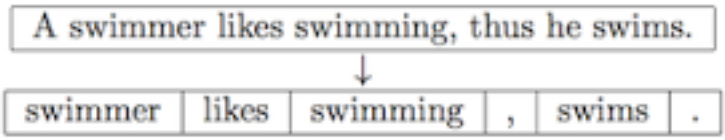

Figure 3.5: An example of a sentence that has undergone stopword removal

3.4.2 Stopword Removal

Put simply, stopword removal is the process of taking out words that are considered to be common across the data. This typically includes words like "a" and "the". These words typically do not add anything to the sentence, and mostly just provide structure. As these words are common, the idea is that they would provide no benefit for comparison purposes. If anything, they would just add computation time to the process. While this is generally the case, these words could still be important. Only test really would show one way or the other. The words that we chose to remove are taken from the default English stopword list used by Ranks NL, a cite that focuses on making tools for search engine optimizations [13].

3.4.3 Case Collapsing

The case of a word is often dependent on where it is located in the sentence as well as if it is in relation to a name or place. To a classifier, the word "The" and the word "the" are different. We consider these as the same, so we go through and change the case of every letter in the data to lowercase. Words with different cases are then seen as the same.

3.5 Surrounding Utterances

Because we take our data from speech of many gathered people in one setting, we can examine the changes in who is speaking as a feature. It is possible that there is a connection to non-legislators turn taking when speaking with a non-legislator, or something similar. Therefore, we choose to examine the utterances that are said before and after the utterance we are extracting features from and include the uni/bi/trigrams of these utterances as features for the current utterance (though they are tagged to differentiate them from the ngrams of the current utterance). This feature is something that is much more available in a spoken medium, as a written medium more often consists of just one author.

3.6 Diarization

"Audio diarization is the process of annotating an input audio channel with information that attributes (possibly overlapping) temporal regions of signal energy to their specific sources. These sources can include particular speakers, music, background noise sources, and other signal source/channel characteristics"[44][51]. This quote from Reynolds and Carrasquillo does a good job of defining audio diarization in general. This definition focuses on the broad scope of what diaraztion can do though. It can help detect the presence of speech, determine the sex of a speaker, and segment audio into parts given the different speakers. For our purposes, it is simpler to say that audio diarization is used to separate different speakers voices in a given recording and group all the things said by each individual speaker together. Figure 3.6 shows an example of how diarization can be used to split one continuous audio stream into chunks split by speaker. These chunks are then transcribed and represented as different utterances. With these groupings of voices, we can run our analysis knowing all

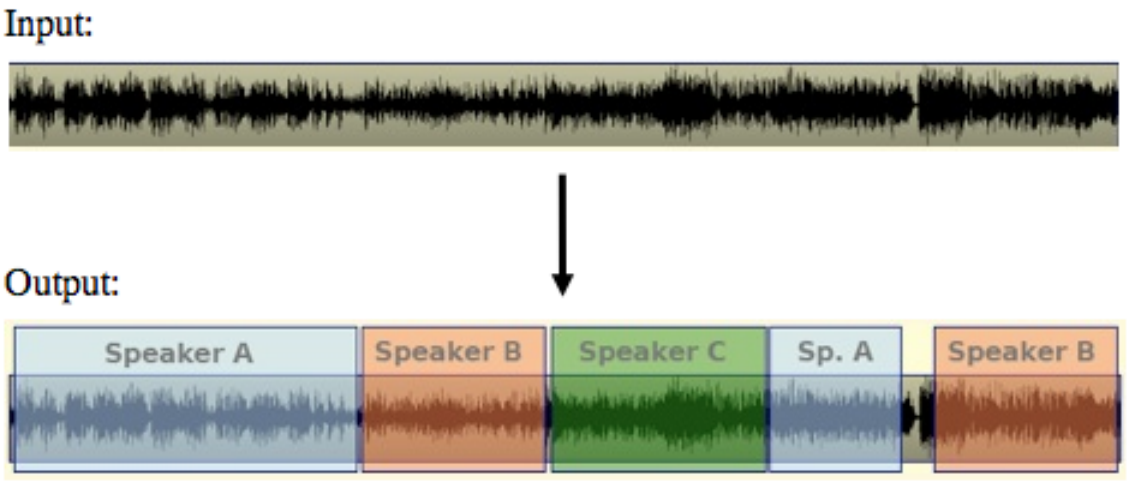

Figure 3.6: An example of how a recording can be split into several speakers through diarization.

of the utterances a speaker has said, even though we may not know who that speaker is.

3.7 Classification

In the terms of this work, classification can be defined as "assigning a label to data based off characteristics of the data itself"[25]. Kohavi and Provost describe a classifier as, "a mapping from unlabeled instances to (discrete) classes." Throughout this section, we discuss classification through the process of supervised machine leaning. We further discuss the models of the algorithms we employ, the type of term frequency weighting we apply for several classifiers, and how entropy relates to the classification process.

3.7.1 Supervised Learning

Supervised machine learning makes use of features that have known labels associated with them [27]. These labels are used to train a classifier and predict the result

of another utterance being tested. A test set of utterances is typically taken from the original set of data. This is in contrast with unsupervised learning in which the data has no labels. Supervised learning is mostly used when trying to determine if data is similar to previously known data, where unsupervised learning is focused on discovering unknown information about items.

3.7.2 Information Retrieval Models

Boolean Model: Boolean retrieval systems are popular for various reasons, high standards of performance being one of them [46]. As Bordogna and Pasl cite in their work, the Boolean retrieval model is still the basis of the majority of commercial information retrieval systems, even with its well known problems [5]. It is the simplest information retrieval model, and uses keyword weights of either 0 or 1 [30]. In relation to our work, this model's main detractor lies in the fact that all terms carry equal importance regardless of their frequency. With the amount of data we have available, however, the high performance may be worth its use if the accuracy in retrieval is still sufficiently high.

Bag-Of-Words Model: The main idea of the bag-of-words model is to quantize the key ideas of a subject into visual words, then represent each image by a histogram of the visual words [56]. This turns the problem into a text classification problem, which in our case is the problem to begin with. The main difference in a bag-of-words model from the boolean model is that the number of occurrences of a word can be taken into account. This is because the representation of a bag of words is normally a vector based off the word frequencies [30]. In our classifiers, we specify the weighting of these frequencies using TF-IDF, which is described below.

Vector Space Model: The basic idea of the vector space model is that the information retrieval objects are seen as elements of a vector space [42]. Terms, documents,

$$enthropy(D) = -\sum_{i=1}^{k} \frac{1}{k} \cdot \log_2\left(\frac{1}{k}\right) = -\log_2\left(\frac{1}{k}\right) \cdot \sum_{i=1}^{k} \frac{1}{k} = \log_2 k$$

Figure 3.7: Entropy calculation

concepts, and queries are all considered as vectors. In the classifiers we use that fall under the vector space model, the features that make up an utterance are what make up the dimensions of the vector using weighting schemes such as TF-IDF [30]. These vectors can then be compared through various different mathematical means, typically cosine similarity, to see how similar they are.

TF-IDF Weighting: TF-IDF stands for Term Frequency Inverse Document Frequency. It is, not surprisingly, a combination of term frequency, which is the standard number of times a term is seen, and inverse document frequency which relates terms to the number of total documents [30]. It calculates values for each word in a document via an inverse proportion of the frequency of the word in a document to the percentage of documents the word is in [43]. In our work, a document would be a text utterance. Words that have a high TF-IDF imply that there is a strong relationship with the document they appear in. On that same idea, if the word is seen less frequently in all the documents, it is more important per occurrence.

Information Gain: Information gain is important in its relation to a disparity in entropy between states [30]. When classifying, we want to select the attribute that splits the dataset D into the most distinct subsets. The entropy can be calculated as shown in Figure 3.7 if each class label has the same probability of occurring being log_2k. If we are given a dataset, the information gain of the dataset after being split using feature Ai given its domain across a vector can be defined as shown in Figure 3.8. While information gain isn't largely used in this work, the idea of differences

$$enthropy_{A_i}(D) = \sum_{j=1}^{s} \frac{|D_j|}{|D|} \cdot enthropy(D_j),$$

Figure 3.8: Information Gain Calculation

in entropy underlies some of the classifiers we use. The decision tree algorithm and, naturally, the maximum entropy classifiers make use of differences in the calculated entropy of the given features.

3.8 Text Classifiers

Here we discuss the several different classifiers that we examine. Each approaches classification in a different way, and we seek to see which works best for spoken word authorship attribution.

3.8.1 Bernoulli Naive Bayes

The Naive Bayes classifier is almost a base standard for various machine learning tasks, including text classification. Zhang and Li describe a Naive Bayes text classifier as, "Bayes Theorem with a conditional Independence assumption that all variables A1...An in a given category C are conditional independent with each others given C [55]." To put it more simply, it is a prediction function that tries to correctly determine the author 'C' given previous data 'A'.

A key thing to note is the "independence" assumption. Namely, "that the probability of each word occurring in a document is independent of the occurrence of other words in a document [34]." Another way to state this is that each feature used in the classifier is treated equally. Additionally, we use a binary independence model in our implementation [28]. This means that each feature in our classifier is put into one of two categories. The feature either is, or is not, related to the class in question.

With that in mind, the information retrieval model associated with this classifier is the Boolean model.

Depending on how many features correctly correspond to the respective author, the probability that they are the author of the source can be calculated. Naive Bayes is a relatively successful classifier, and is popular due to its computational efficiency and good predictive performance [9]. It is very sensitive to feature selection though, so the features used with this type of classifier are extremely important.

3.8.2 Multinomial Naive Bayes

Multinomial Naive Bayes is a different variation of the Bernoulli classifier described in the last section. It is able to capture the word frequency information from the data being examined [34]. Because of this, its information retrieval model changes to a bag-of-words. If certain words are showing up many times, then this classifier can take that into account, where the Bernoulli classifier would only recognize the word as a feature once (but it would still be relevant in various ngrams). This key difference warrants the exploration of the Multinomial classifier as a speaker classification option.

3.8.3 Decision Tree

Safavian and Landgrebe's work describes the process of the Decision Tree classifier as breaking down complex decisions into a combination of multiple simpler decisions [45]. They are an attractive option in that complex decisions can be approximated by combining various simple decisions along the tree. While running the entire tree could take a very long time, and possibly cause overfitting (making the model for the classifier fit too closely to the training data [22]), the classification can be cut off early and the decision made based off the decisions that have already been made. This process in known as pruning. The tree is normally built from the top down,

with the most relevant nodes - those with the highest information gain - at the top [2]. They usually use a breadth or depth first search during tree creation.

3.8.4 Maximum Entropy

The Maximum Entropy (Maxent) classifier is more probability based, and takes its constraints from the training data features and outcomes [10]. The probability distribution that has these constraints, and makes as few assumptions as possible, is the one with the highest entropy. The algorithm can then use this distribution to make decisions one way or another. Unlike Naive Bayes, the algorithm doesn't assume that the features it is given are conditionally independent [39]. It makes use of sparse vectors of 0's and 1's following a bag-of-words model. It has been shown to sometimes, but not always, outperform Naive Bayes classifiers in standard text classification, so it is worth examining.

3.8.5 Support Vector Machines

We have already minimally discussed support vector machines (SVMs) [14], but what does this algorithm actually entail? Hearst et al. describe it as "a linear algorithm in a high-dimensional space...[that] does not involve any computations in that high-dimensional space [16]." Their work describes the basic idea of SVMs as the mapping of data onto a "feature space" through a series of dot products. The mathematics behind SVMs can be a bit complicated, so we will look more into its relations to authorship attribution. Campbell et al. comments on SVMs powerful ability in pattern matching [7].

As authorship attribution focuses on identifying patterns in an author's work, it makes sense to explore its uses in the field as Cambell et als. work and other works have. Diederich is one such work that states the simplest way to think of SVMs

Figure 3.9: An example of a sentence undergoing entity recognition.

[12]. Namely, it creates a hyperplane that separates positive examples from negative examples. When an utterance's features are mapped, its position in n dimensional space is located in regards to this hyperplane to determine if it matches with the positive or negative examples. Pang's work cites research done by Joachims in 1998 that found SVMs to be highly effective at text categorization, generally outperforming Naive Bayes [39]. In terms of information retrieval models, it is, of course, vector space based. As the algorithm has preformed well in various experiments into authorship attribution, and Digital Democracy's own voice recognition, it is worth examining in this new setting of text categorization of speakers.

3.9 Entity Recognition

Another modification we make to the text of each author's transcription involves entity recognition or "Named Entity Recognition and Classification" [36]. Many modern entity recognition techniques involve some sort of machine learning to identify the entity and references to that entity. An entity itself is a word that represents something like a person, company, location, or time. The name "Bill" or "Walnut Park", would be examples of such entities. Figure 3.9 shows an example of a sentence undergoing entity recognition. We use AlchemyAPI for our entity recognition [52]. AlchemyAPI provides many different text analytics services such as sentiment analysis

and keyword extraction. In terms of entity recognition, however, we use it to get the entity itself, as well as what type of entity it is. These types could include "Person", "StateOrCountry", and "Quantity". To help make the dataset more uniform, we take the original entity out of the text being analyzed and replace it with the type of category it belongs to.

3.10 Natural Language Processing Modules

Throughout this work's development, several different natural language processing (NLP) modules were used and are detailed below. Both were used for their implementation of natural language processing algorithms, data preprocessing functionality, and compliance with the python Programming language.

3.10.1 Natural Language Toolkit

The Natural Language Toolkit (NLTK) is an easy to use modual that was originally designed to help with learning about NLP. The early work of Bird and Loper, and Bird's later republication, comment on NLTK [31][4]. It was created with many requirements in mind, but of most import for this work is it's simplicity, good documentation, and consistency. It provides the ability to tokenize and stem text utterances in one line of code as opposed to having to implement the entire algorithm ourselves. NLTK also comes with the ability to interface into Scikit-learn [40], allowing for the use of more complex algorithms with specific run time parameters. For the use of algorithms, a simple pipeline object can be created to set parameters that interface into Scikit-learn. An example of this is show for Multinomial Naive Bayes in figure 3.10

```
pipeline = Pipeline([('tfidf', TfidfTransformer()),
                     ('nb', MultinomialNB())])
classif = SklearnClassifier(pipeline)
```

Figure 3.10: Snippet of a pipeline used to interface Scikit-learn from NLTK

3.10.2 Scikit-learn

Scikit-learn's more complex algorithms require more math, which requires it to interface with Numpy and Scipy (other python modules that handle math) [40]. While Scikit-learn is well documented, it is a bit harder to use than NLTK. With this complexity comes the ability to do many more things such as regression and clustering. For our purposes, the simplicity of NLTK's wrapper for its use is why the two make a great pair for providing many tools needed in natural language research.

3.11 Digital Democracy

Here we discuss the key elements of a project that this work is heavily related to, and the overall big picture of how this research can be used in the immediate future.

3.11.1 Background

The research done for this work is heavily tied into a non-profit project called Digital Democracy. The project is a product of the Institute for Advanced Technology and Public Policy at Cal Poly San Luis Obispo. At its core, Digital Democracy is an online platform the provides a searchable database of state legislative committee hearings. Currently, California is the primary state with accessible information, but other states are currently being incorporated into the project as well.

While the Digital Democracy platform provides a searchable database for users, all of this information stems from the work done in transcribing legislative hearing

videos. These transcriptions are something that were not easily available to the public, or didn't even exist prior to the projects creation. The project has many challenges, with the three major technological problems currently being researched being transcription, speaker identification, and video indexing. This is where the research of this work comes into play. It directly addresses speaker identification, while also helping with the issue of transcription.

Speaker identification and accurate transcriptions are problems that are tied together. If there is not enough information to determine the identity of a speaker, then there is no way to have a completely accurate transcription. Even with information about who the speaker most likely is, there is still room for human error. This is where the idea of Voice-Face-Text recognition (VFT) comes in. The thought process behind VFT is to use all possible information we can take from a video about a speakers identity and use it to correctly identify them. How a speaker looks is key into facial recognition, but the speaker is not always in view of the camera when they are talking.

The way the speaker sounds is normally unique to a degree, but imperfect audio and a large number of competing different voices to analyze make this difficult as well. Finally, there is the actual words that the speaker says. While the vocabulary and speaking style of a person may not be unique, it will differ enough to make some distinctions between one person and another. By combining these three classification methods, each of the weaknesses of the individual classifiers reinforces the others, creating one overall stronger, more accurate classification process.

3.11.2 The VFT Process

Figure 3.11 shows the general overview of how the VFT process functions. When used as a tool for Digital Democracy, all of the classifiers are trained with all the

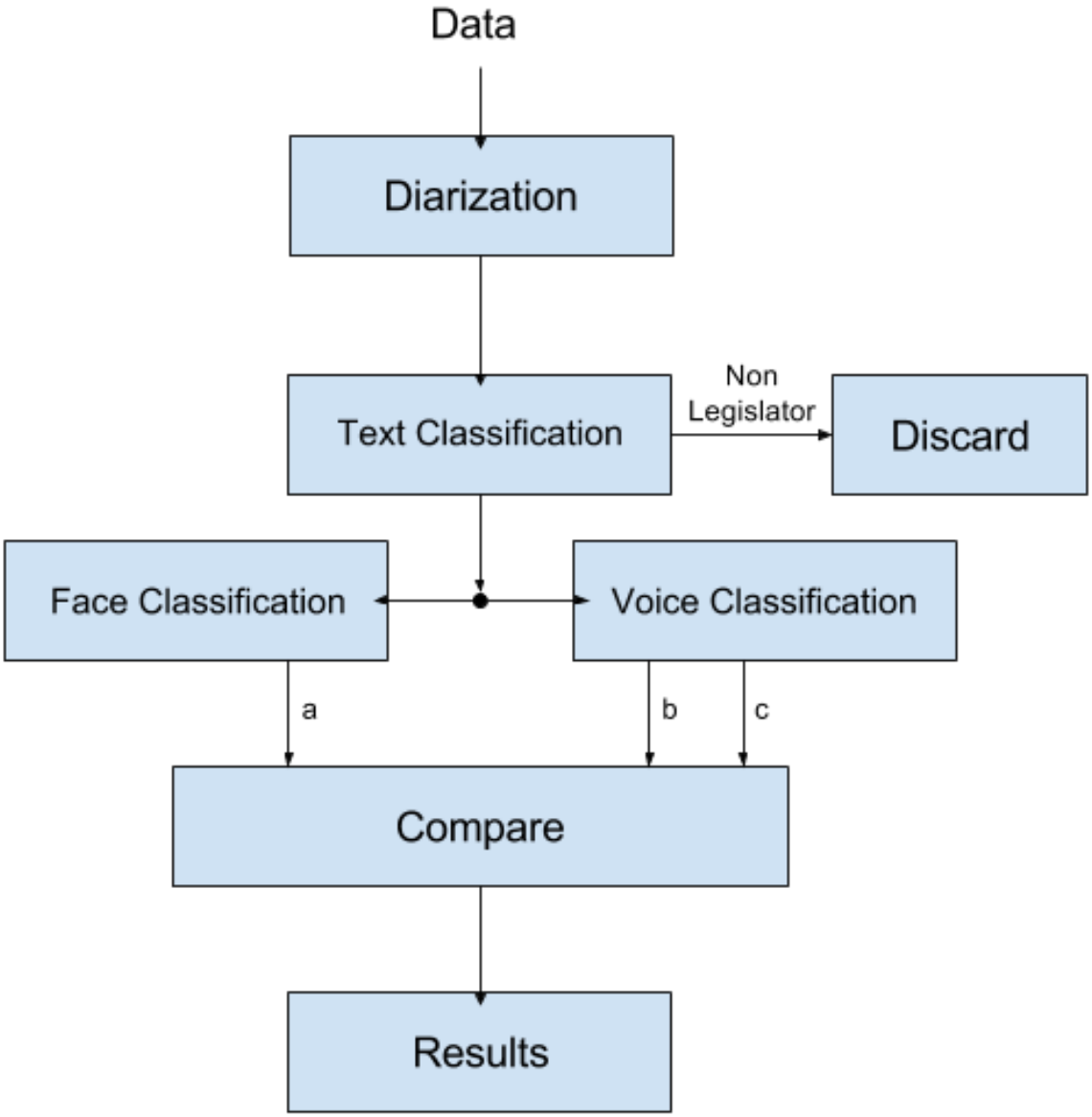

Figure 3.11: The general flow of data through the VFT process

data currently available, (all of the hearings that have been thus far transcribed and have the speaker associated with the given text). A new hearing being processed is represented as the "Data" portion of figure 3.11. The data is first diarized to group the text utterances together by speaker. Assuming that the classifiers have already been trained, the data is then passed into the text classifier. There, the utterances are either tagged as being said by a legislator, or as a person who is not a legislator. Those that fall into the second category are discarded, while the believed legislator utterances are passed on to the face and voice classifiers.

Facial recognition is preformed with one classifier, and determines who it believes spoke each utterance. The voice classification (for the case of Digital Democracy at least) has been constructed to use two classifiers. One uses support vector machines as the primary means of classification, while the other uses a layered neural network. Both classifiers make their decisions as to the speakers identity independently, and forward their results. All three results are then compared, and the speaker chosen based off the combined results.

This speaker identification process is based off of the transcriptions that are already available through Digital Democracy's efforts, but it can also improve the transcriptions moving forward. As no commercially available solutions are good enough to produce quality transcriptions, Digital Democracy often relies on a combination of machine and human transcription. Previously, a human would have to identify a speaker based off their best judgement. With the completion of VFT, however, we can integrate the classification process into the transcription pipeline. This would attempt to determine the identity of a speaker, and present the transcriber with the best candidates for the speaker of a given text utterance. By using the VFT classification as a tool for transcription, we can help reduce the human error involved with transcription, as well as being able to determine the identity of a speaker that was otherwise impossible to definitively determine beforehand.

Chapter 4

RESEARCH GOALS

4.1 Contribution to Speaker Recognition

This work seeks to answer several questions in regards to spoken word style attribution. We seek to answer one main question from our research. We provide an explanation of the thought process relating to the work, and our hypothesis relating to the results.

4.1.1 How much can style attribution through text classification improve speaker recognition?

Facial and audio classifications are what attempt to identify who a given speaker is, and are what we are trying to improve on. By filtering out the non-legislators, whose identities are less important, we can lessen the amount of data these classifiers need to process as well as decrease the amount of people they need to attempt to classify. We examine how much of a difference the addition of text classification makes in the overall process. We also explore various different aspects of the classification process such as the features, datasets, and algorithms used in the research.

Supposition: Given various types of classifiers, datasets, and features that preform well in a natural language setting, we attempt to determine various insights into style attribution. Different algorithms may have different benefits.

Some are fast but do not typically have high accuracy, while others take much longer and may only slightly improve accuracy over another method. The algorithms used also focus on different information models such as boolean, bag-of-words, and vector space. Given that style attribution has not been thoroughly explored, one

model may work better than another.

We tune the algorithms being used with various features. Some may preform well, while others decrease classification performance. We also seek to increase the recall of the C-SALT without lowering the overall accuracy significantly. With this in mind, we try to observe which features best help accomplish this.

As we have discussed, different amounts of data can have a big effect on the accuracy of classification. Given various datasets with variations in size, number of legislators speaking, and "noise" (the amount which non-legislators say things that are similar to a legislators speech and vice versa), we examine the classifiers ability to differentiate each group correctly.

All of these characteristics build the foundations of this research. The overall problem, however, is the ability to correctly recognise the identity of a given speaker. Face and audio recognition do not have an efficient means of differentiating legislators from the public on their own. We believe, however, that text classification can attempt to preform this role instead.

Hypothesis: In terms of algorithms, a simple approach such as Naive Bayes may be the most efficient way to achieve high results. Alternatively, Support Vector Machines have worked well in written settings of individual authors, and may preform well in this setting too. Given various features, the accuracy - and recall - of C-SALT will also be higher given the addition and filtration of said features.

C-SALT training on these features will be informative on each category of classification. While varying degrees of data availability may lead to lower accuracy, the classifier should still preform respectably given various datasets we examine. Overall, the addition of text classification will significant increase the accuracy of both facial and audio classification by decreasing the overall sample size they need to examine. This in turn will drastically improve speaker identification as a whole.

Chapter 5

THE EXPERIMENT

This section will address the background of the experiment being executed, the datasets involved with the experiment and the process in which it was ran, as well as the overall results that the experiment yielded.

5.1 Project Data

This section discusses where we procured the data used throughout the project as well as identified issues that come with it.

5.1.1 Data Quality

The data sources used in this work are taken from recordings of the California State Legislature. As these videos are not normally transcribed, transcriptions were required to be created. The Digital Democracy project has provided this service for us through their own work. This starts by first running the audio through a commercial machine transcription tool, followed by a human verification of the transcription. With that said, the transcriptions that they provided are not perfect, and occasionally will have mistakes from either the machine or human work.

There are times when a speaker may be attributed to an incorrect utterance, (the very problem this research is seeking to help solve). Possible spelling mistakes may be present, as well as the lack of a word or two the transcriber missed. These are sources of error that will have to be dealt with as they are common in real life scenarios, and there is no way for us to fix the issue using this dataset. Of particular note for this research is the punctuation and the removal of disfluencies. As the

punctuation is subjective to the transcriber, this may no longer be a useful feature. As for the removal of disfluencies, it makes sense that they would not be included. It is unfortunate for this work, however, that we don't have that information available.

5.1.2 Data Sources

The simplest and most common attribution typically is done with a small, closed set of authors [26][32]. As Koppel et al. describes, it is usually the case that there are copious quantities of text by each author available. Also, the text being classified is normally fairly long as well. These assumptions, while still providing relevant information, are not the typical case. There could be a large number of authors that need to be compared against, or a small amount of information that you can work with [26]. Luyckx and Daelemans describe how many researchers use over 10,000 words per author, on very few (with the bare minimum of two) authors. In most cases, this simple type of authorship attribution can be accomplished with accuracies over 95% [32]. This is not a big surprise considering how much training data is used on such a small amount of people. This work takes a hint from Luyckx and Daelemans' work, and looks at a range of different authors and data. This can be seen from comparing a somewhat small set of data to an extremely large set of data. Each dataset will be further discussed in a later section.

5.2 Text in VFT

As we have discussed, the voice/face/text identification process is a multi-step process with text classification being the first executed. It is not meant to preform the main classification of, "who is the person that said this utterance." It is a tool to aid the face and voice classifiers. All three of the classifiers involved focus on being able to identify the legislators that are on the committee of the hearing being analyzed. As

the members of the committee typically discuss the most import issues, represent the people that have voted for them, and make the policy that affects the citizens of California, knowing what statements they have made and what opinions they express is essential to both the public as well as a political advocate. It is also convenient to focus on the legislators of a committee for a given hearing because hearings typically only have one committee present at a time.

Separating the legislators from everyone else is beneficial in that it simplifies the data that the other classifiers receive. By omitting everyone but the legislators, the face and voice classifications have less entities that they have to compare against. Facial recognition, for example, no longer needs to examine faces of people that have been determined to be a non-legislator. This saves computation time, and reduces the possibility of a false classification, (given the text classifier is correct). With that, the total number of faces in the hearing that are being used in the classification process is also lessened, making the face classifier more accurate and faster in that is has less total faces it has to compare against.

This similarly carries over to voice recognition. We begin with fewer voices that need to be considered in the first place, and fewer voices the classifier needs to compare against. Without the added layer of category classification above face and voice, the classifiers would need to handle the problem on their own. Both do not have an easy means of determining this, meaning that they would need to attempt to identify every person that speaks in the hearing. While knowing that random members of the public, such as John Smith may say, "I support this bill" could be useful, knowing that the committee chair of the hearing said, "I support this bill" is much more useful and impactful to the hearings outcomes. Confusing John Smith and the committee chair, however, is a gross error, and is a problem that has already been seen in the current transcripts of some hearings.

To put the importance of the text classification layer into perspective, the dataset that is being used to test the full VFT classification process is composed of almost half legislator and half non-legislator utterances. That means that if the text classifier worked perfectly, the remaining face and voice classifiers would have half as much data that they have to processes and attempt to identify.

Separating the legislators from everyone else speaking involves several considerations. The overarching question that needs to be answered is "who do we consider a legislator in a given hearing." This seems rather straightforward on the surface, it would seem to be only the elected legislators. From a classification standpoint, this question is much more difficult. Text classification separates people into categories based off of the words they use when speaking. Basically, we are trying to group authors into a category based off their speaking style.

There are many people in hearings other than legislators, however, who use the same words as legislators and talk in a similar fashion - they have many similar styles. While only some members of the public fall into this category, the main issue is the use of similar styles by lobbyist, legislative staff, and though it may not seem obvious, the secretary taking notes. These people are not legislators, and should not be classified as such. As we will address further on, text classification needs to look into this issue lest it be left as a source of error.

5.3 Text Classification

This section discusses the implementation of the text classifier and the datasets that are examined. As text classification on the spoken work has not been thoroughly explored, this work surveys the effects of different features used with several algorithms.

5.3.1 Process Overview

Figure 5.1 shows the way data is handled by the text classifier. First, all data that we are processing is taken in and split up into two separate sets. The first set is the training set. It is used to train the classifier as to what features relate to legislators and non-legislators. The training set makes up 4/5th of the data we have available, and is represented by the thin arrows in Figure 5.1.

After the data has been split, the training set is sent through preprocessing (though this step is intentionally not done for the first featureset we examine), and then feature extraction. After all of the features from every training utterance have been related to their respective classification category, the classifier starts to handle the last 1/5th of the data. This is what we consider to be the test data and can be seen by the thick arrows in figure 5.1.

The test data initially goes through the same process as the training data of preprocessing and feature extraction. We do not process the identity of the utterance as in the training set, however. If we simply knew the identity of the speaker that we are training on, the classifier could cheat and get perfect classification. With this in mind, the identity of the utterance that we have for test data is just used to check if our classification was correct. The steps of feature extraction and classification are each done utterance by utterance and grouped by diarization.

After feature extraction is complete, the trained classifier is used to provide a classification for the utterance being examined, shown by the double arrows. When all classifications are made for a given speaker diarization, the classifier preforms a vote. The specifics of this voting process are described in a later section, but the general idea is that the previous classification of certain utterances may be changed depending on the vote. After the vote, the utterances that are being examined are given their final classification and are then checked for correctness.

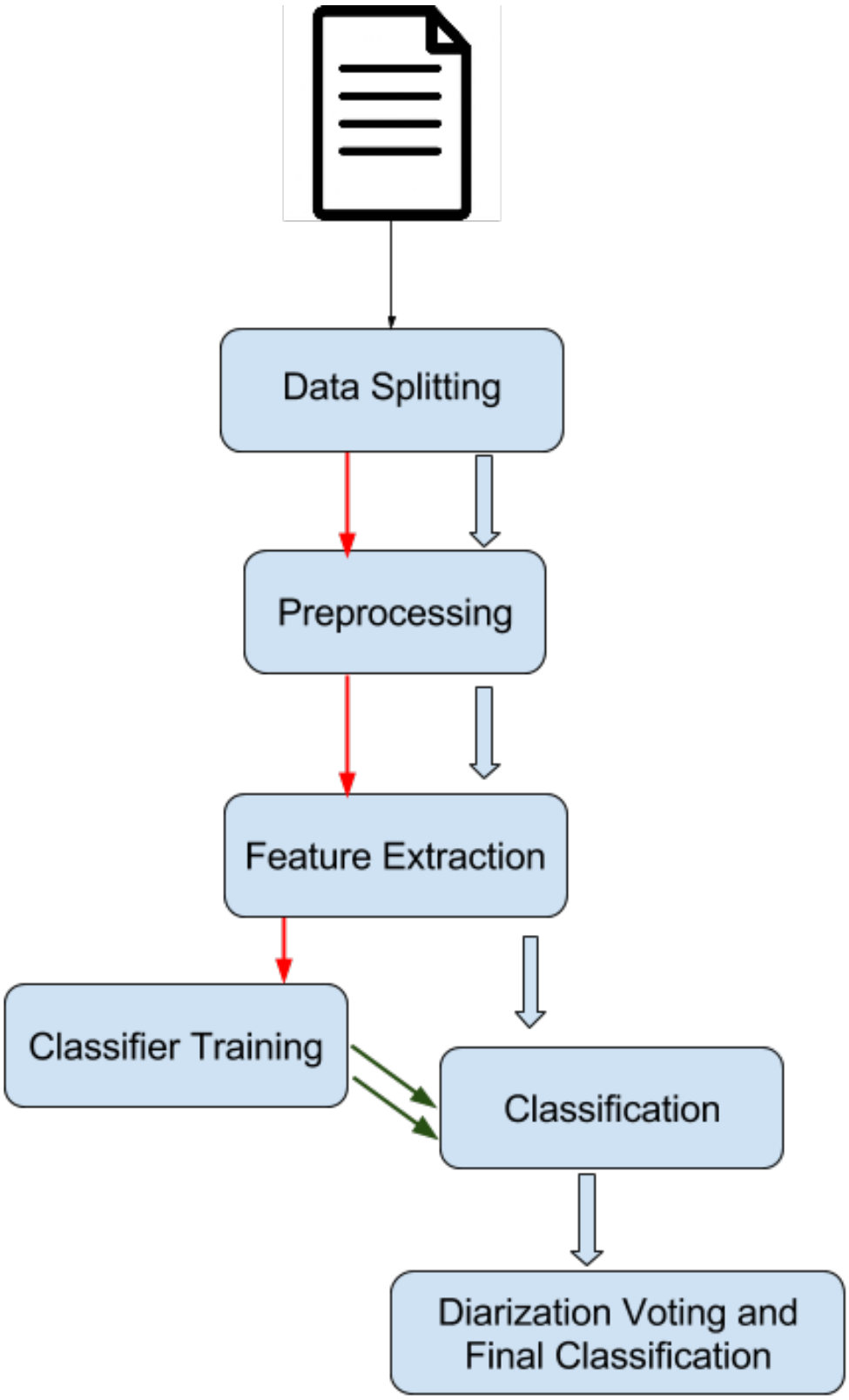

Figure 5.1: The process flow of the text classification process

5.3.2 Dataset Overview

Table 5.1 outlines the five algorithms and five datasets that were examined throughout this experiment. The main differences in the algorithms have been previously explained, so we will now examine the datasets we are testing with. The datasets contain all of the information available to the text classifier. This data is separated by hearing into training and test sets. Once again, all of the hearings are parsed, randomized, and then split so that four-fifths of the hearings are in the training set, and one-fifth comprises the testing set.

One Hearing: The first dataset consists of all the hearings available from one specific committee. While this dataset is one of the smaller sets, it is interesting because most legislators that are present in one hearing will often attend the others. As the committee is made up of the same legislators, for each meeting the people on the committee will make up the majority of the legislator utterances in the test set. This is significant due to the nature of of style in the classifier. Because the same style of speaking will be seen repeatedly, it is possible that the accuracy of the classifier will increase. The fact that the dataset is small, however, means there is less information to train the classifier with. This generally lowers the accuracy seen and makes the classifier particularly vulnerable to training on irregular utterances that may have been spoken.

Vaccine Bill SB277: The second dataset used was created as a "noisy" compilation of utterances. It is comprised of all the utterances in relation to Bill SB277. This was an extremely controversial bill that involved children being required to be vaccinated against certain diseases in order to be enrolled at their school. This dataset is considered "noisy" in that there was a large amount of discussion on the topic, many members of the public spoke, and the words that the members of the public used were often similar to words that legislators were using. With this in mind, it would make

correctly classifying a legislator more difficult. It is also an interesting set of data to examine due to the nature of the controversy involved and the diverse opinions.

Small, Large, And Everything Datasets: The last three datasets were picked due to the nature of their size. The small hearing set is made of ten randomly selected hearings and is meant to examine the problems involved with such a size. While it is a bit bigger than the one hearing dataset, the hearings are not all comprised of one committee, thus missing out on some possible stylistic similarities. The second set composed of a large number of hearings is the dataset that was originally created to test the VFT classification process. It has a large amount of hearings, 61 in total, and is representative of what can be seen in actual use. That brings us to the third dataset. Namely, this is all the current utterances that are in the Digital Democracy database. In total, there are over 300,000 utterances, making it ten times bigger than the large hearing dataset. This test is basically a realistic classification that Digital Democracy can expect to see when classifying new incoming data. The fact that it is so big means that there is a lot of data to train with, but also means that it takes much more time to preform said training and subsequent classification. In real use, the training would only need to be done once every few weeks with the new information that is available, and only a few hearings would be considered the "test set". This means that the time to run the classifier is not a major consideration, but is worth noting when comparing the performance of different algorithms.

Dataset Error: Some interesting challenges involving all the datasets are possible sources of error. Because all of these datasets have been human transcribed at the core, error is inevitable. These errors will hopefully be addressed better once VFT is implemented in assisting transcription, but are something that simply have to be dealt with for now. One major source of error was actually found in the large hearing dataset when researching on improving the text classifier. In particular, there are numerous cases when an important legislator is talking, but the identity associated

Table 5.1: The algorithm and dataset used

Algorithm	Dataset
Bernoulli Naive Bayes	One Committee
Multinomial Naive Bayes	Noisy
Decision Tree	Small Number of Hearings
Maximum Entropy	Large Number of Hearings
Support Vector Machines	Everything in the Digital Democracy Database

with their transcription is classified as a member of the public who spoke previously in the hearing. This is not just an error that appeared in the dataset we are using, but an error on the live website for Digital Democracy. An error such as this causes stylistic features of a legislator to be attributed to members of the public. This error was found not only once, but twice in the dataset. To make it worse, the legislator that was mislabeled was clearly the Committee Chair - arguably the most important legislator in the hearing.

These were errors that were noticed in the attempt to improve the text classifier, meaning there are quite possibly more instances of this that are unknown in the dataset. The opposite issue (members of the public having been identified as legislators) is also largely possible, but was not observed first hand. Figure 5.2 shows an example of one of these errors that was found in the dataset. These utterances were all grouped together by diarization (and since we assume perfect diarization, this means they were in fact all labeled as the same person) in the test set of an early experiment using the text classifier. Every utterance above the dashed line was classified as a non-legislator utterance, and they were all correct in that attribution. Every utterance below the dashed line was classified as a legislator utterance, and they were technically correct as well. The issue is that everything above the line was said by, and labeled as, Rebecca Lee, a member of the public. Everything below the

line was labeled as Rebecca Lee in the Digital Democracy database, but was actually said by the Committee Chair.

Another source of error that was found in the process of improving the text classifier was the existence of an "unknown speaker" identity in the Digital Democracy database. This is the identity that the human transcriber is supposed to label an utterance with if they are unsure who said it. This means, however, that legislators and non-legislators are likely to be mixed into the same identity, once again mixing the stylisic features of the two categories. Fortunately, the classifier can completely disregard this category, leaving it out of training. Unfortunately, we have been informed that there are possibly other smaller cases of exactly this same issue, simply with a different name. As we were not able to identify all of these, they are another source of error that could be possible in a dataset. As they would have a very small amount of utterances associated with them, however, their impact should be very minimal.

5.3.3 Features

Possibly the most important knowledge this research looks to unveil revolves around the interaction of different featuresets in relation to textual classification of the spoken word. Table 5.2 describes the list of featuresets that we examined throughout the research. Individually, each feature can only preform so well, so we try to show how the addition of each additional feature impacts the classification. The features at the start of the table are the most basic features, and more advanced and specific features are added as the list moves downwards.

N-gram Usage: Looking at the first featureset, we only use data that has not been manipulated in any way, and just take each word in the utterance as it is for the purpose of classification. This provides a good base to start off with and provides a

- chair members, rebecca lee, on behalf of the office of rate payer advocates. first, we are supportive of energy efficiency measures, inclusive of military applications. and we initially had submitted an opposed unless amended position.
- but on upon review of the committee's suggested amendments, we do believe that this, that the amendments would address our concerns. and we actually have been having very good conversation with the author's office on clarifying.
- and our, our only issue we were concerned with really was just a narrow issue about a provision that could complicate things and could have some unintended consequence. and so i think this is, with the committee's amendments, would be removing our opposition. thank you.
- chair members, rebecca lee, on behalf of the office of rate payer advocates. we did submit an opposed unless amended position letter. however, upon review of the committee's suggested amendments, we would be removing our opposition because we do not take issue with enrolling customers up to 20 years but we do believe that the bill credit could, for reasons already stated by matt earlier, that could be subject to change.
- and we do believe that as the bill moves forward that it is important to make sure that the participating customers are aware that the bill credit could be subject to change within reasonable bounds, subject to speci, pre, prespecified reasons.
- so, therefore we're happy to continue working with the author's office as this bill moves forward. thank you.
- chair members, rebecca lee on behalf of the office of ratepayer advocates, we are in strong support of this measure.
- we do believe that given that the ratepayer's currently fund more than a billion a year to support energy efficiency programs, that it is important to dedicate as, a portion, of, of those funding tour activities to facilitate longer-term, strategic, comprehensive initiatives to bring about business and consumer behavior, and saving energy.

- so, we look forward to helping, working with the author's office as this bill moves forward. thank you.
- and, and it has come to my attention there's opposition to the bill, so we'll hear from the opposition. you want to come forward?
- thank you very much.
- anyone else out there?
- very well. anyone else? do we have any questions or comment from committee members? senator hill?
- we have a motion. senator, would you like to close?
- okay, very well. we have a motion. clerk please and that's do pass to approps. clerk, please call the role.

Figure 5.2: A trace of utterance classifications with mislabeled identity

Table 5.2: Feature sets

1) Plain Word Unigrams
2) Word Unigrams w/ preprocessing
3) Word Uni/Bi/Trigrams w/ preprocessing
4) Word Uni/Bi/Trigrams w/ preprocessing, stemming, stopword removal
5) Word Uni/Bi/Trigrams w/ preprocessing and letter level Uni/Bi/Trigrams
6) Word Uni/Bi/Trigrams w/ preprocessing and using previous/next utterance
7) Word Uni/Bi/Trigrams w/ preprocessing, selected features, and utterance length
8) Word Uni/Bi/Trigrams w/ preprocessing, selected features, and uniform entity recognition
9) Word Uni/Bi/Trigrams w/ preprocessing, selected features, and category entity recognition

comparison for the second test case. Namely, we use the same unigram based features, but run the data through preprocessing. This includes changing the case of letters to all lowercase, tokenizing the utterances as opposed to splitting the utterances on spaces. While these changes may seem small, it is possible that they make a significant difference because the words being used as features are much more comparable. After looking at just the unigrams of the utterances, we move on to include the bigram and trigrams. These features are useful in picking up certain phrases of words that one group of people might say as opposed to another.

These are the components that provide us with the base features we use across the other features we examine. The features following these have varying effects on the accuracy of the classifier. Some actually hurt accuracy, and so they were not further tested with. Others would increase the accuracy, and so they were continued to be used in the examination.

Stemming/Stopword removal and Letter Level Ngrams: The presence of stemming

and stopword removal are the next things that were tested. Being standard tools in any text based knowledge discovery process, seeing their impact on the accuracy of the classify in a spoken word setting is interesting to us. Addition of unigrams, bigrams, and trigrams are then explored, but on a letter level instead of a word level. While adding these features increased the computation time of the classifier more than any other feature tested, their use is another common tool that is often helpful when examining the text an author has written. Seeing how they preform in regards to a group classification and on spoken language is, once again, interesting to examine.

Surrounding Utterances: For experiment six, we look at the dataset from a different angle. As we know there are several people talking in the hearing we examined, so it is possible that the dialogue itself can be useful as a feature. Consider a back and forth conversation between two individuals, the words that one person says might be telling as to what category type the other person would fall under. This leads us to include the unigrams, bigrams, and trigrams of the previous and next utterances, (with regard to the utterance we are examining) in the hearing as possible features in the classifier. A possible issue with this feature, however, is that the type of person speaking before or after the current person will not be consistent across the data.

Length and Content Specific Features: Experiment seven adds two more types of features to the mix. The length of the utterance is mostly self-explanatory, and could be useful if there is a significant disparity in the amount the legislators talk in comparison to everyone else. The "selected features" feature, however, needs some explaining. Basically, we have found certain aspects of legislator speech that we can leverage, as well as certain problems in the classifier that we can fix. One such example is the fact that the classifier training consistently thinks the one word utterance "aye." is something that a non-legislator says. This makes sense, as it recognizes

that the secretary repeats back the votes of the legislators and says "aye" a lot. This utterance on its own, however, is clearly something that would almost always be said by a legislator, so we can specifically look for it, and always denote it as such. Upon examining the data, we also observed that legislators are several times more likely to ask a question than the public. With this in mind, we can add the presence of a question mark in the utterance as another feature to the classifier. One thing to note is that question marks are only present if the transcriber of the hearing added the punctuation of the question correctly, so not all questions will have correct punctuation. This weakens the effect of such a feature, but it still is useful to explore even if it only helps an ok amount.

Entity Recognition: The features used in sets eight and nine are very similar. Both use entity recognition on the utterance that is being observed. The difference between them is how they handle the entities. In eight, if an entity was found in the utterance, then it is replaced with "__Entity__". This means that all entities in an utterance are treated equally regardless of whether it is a person or a country. Set nine is where the type of entity comes into play. It replaces entities with the category that was associated with them. Instead of everything being replaced with "__Entity__", a persons name would be changed to "__Person__" and a country would be replaced with "__StateOrCountry__". The minute differences examine two different aspects of the data. They are used to determine if the sheer presence of entities is what matters, or if the subject matter of the utterance is of more importance.

5.3.4 Additional Processing

After the classifier has taken the featureset for the utterance and made a classification, there is one more modification that can be made before diarization voting. This process involves taking the utterance that was just classified and running it through

two additional separate classifiers. These classifiers take the same idea as the initial classifier, except they constrain the non-legislator category significantly. The first classifier tries to separate a secretary from a legislator. As the secretary says many things that are similar to a legislator, they are a common source of error. The second additional classifier if focused on lobbyists for the same reason.

These classification categories have been separated from others and used because they are groups which have many utterances that could be misclassified. There are other groups that could also fall into these categories, but we do not look into them as they have much fewer utterances to consider in the first place. Each additional classifier takes a simple Bernoulli Naive Bayes approach as we want them to preform quickly. If the original classifier was using the Bernoulli algorithm, the fastest of all the algorithms we are testing, these additional classifiers would cause the time it takes to preform the entire classification to be almost three times as long. As the amount of data that we are trying to correct is worth examining, but not overly abundant, it makes sense to use the fastest algorithm we have available as long as it still preforms reasonable well.

Each classifier only trains on utterances that were said by a legislator or a lobbyist or secretary. This maximizes the stylistic tendencies seen on the lobbyist and secretary side because the other non-legislator speaking characteristics have been removed. If enough of the utterances in a given diarization group are classified as a lobbyist or a secretary, then the utterances for the diarization group are considered to have been said by a non-legislator. In the results section of this work, we examine the effect of these two classifiers after we investigate the effect of each featureset. We attempted this is a process ran after feature classification occurred, and then also attempted to see if it could be integrated as a feature.

↓ actual \ predicted→	negative	positive
negative	a	b
positive	c	d

Accuracy $(a+d)/(a+b+c+d)$.

True positive rate (Recall, Sensitivity) $d/(c+d)$.

True negative rate (Specificity) $a/(a+b)$.

Precision $d/(b+d)$.

False positive rate $b/(a+b)$.

False negative rate $c/(c+d)$.

Figure 5.3: The confusion matrix and standard related terms by Kohavi and Provost [25]

5.3.5 Evaluation Metrics

This section describes the different metrics that are used in evaluating the text classifier across the various datasets. While other metrics are available such as f-measure, false positive rates, false negative rates, and so on, these were what were most insightful to the work.

Confusion Matrix: Kohavi and Provost detail the idea of a confusion matrix [25]. They state that is is, "a matrix showing the predicted and actual classifications." They expound upon this with the chart and formulas shown in figure 5.3. Of this chart there are several values to be defined: value a represents a "True Negative" result as both the predicted and actual classification were negative, value b represents a "False Negative" result of giving a positive classification when the real classification was negative, value c represents the "False Positive" result by classifying as negative when it was positive, and value d represents the "True Positive" result where both the actual and predicted classifications were positive. In the case of this work, a negative classification equates to a non-legislator while a positive classification equates to a legislator. Various rates are then presented in the chart, but the three that we will focus on are the Accuracy, Precision, and Recall metrics.

Accuracy: Accuracy is the simplest metric for testing the classifier. It can be defined as, "the rate of correct (incorrect) predictions made by the model over a data set [25]." Basically, this metric is calculated by taking the total number of things correctly classified and dividing by the total number of classifications made.

Precision: Precision can be defined as, "the proportion of predicted positive cases that are correctly real positives [41]." This is shown in figure 5.3 as the True Positives/(True Positives + False Negatives) - everything we said that was a legislator divided by everything we said was a legislator regardless of correctness. This measure goes down as the classifier says more and more non-legislators are legislators.

Recall: Recall can be defined as, "the proportion of real positive cases that are correctly predicted positive [41]." In areas such as information retrieval, this metric is not considered as important in precision, but in the case of this work, it is considered more important. While is is important that we try not to classify non-legislators incorrectly, it is more important to correctly identify a legislator. As detailed in figure 5.3, it can be defined as True Positives/(True Positives + False Positives) - everything that we said was a legislator divided by everything that actually was a legislator.

5.3.6 Use of Diarization

Benefit of Diarization: The presence of the diarization process being run on the utterances of our data lets us make some interesting improvements to the text classifier, as well as providing room for selective decisions about where we want the classifiers accuracy to shine. The text classifier functions by taking each transcribed utterance and extracting the features for just that instance. Those features are then used on the trained classifier, and the text is tagged as either being said by a legislator or not. In some cases, there are many features that can be extracted from the text. There

are many other cases, however, of a person only saying short phrases of words such as "thank you". This leaves little information that can be extracted and attributed one way or another. This is where the use of diarization comes in.

While the "thank you" utterance will be classified however the classifier sees fit, we can examine the other utterances that are associated with it through diarization and see how they were classified. We can then make a decision about the entire set of related utterances by using the individual classifications as members of a voting process. This voting process helps to achieve an overall higher classification for several reasons. The first main benefit is the ability to deal with classifications from utterances such as the "thank you example". This is a short piece of text that is commonly said by common people and legislators alike.

If an utterance such as this is classified as a non-legislator utterance, but all the other classifications associated with the diarization are thought to be from a legislator, this utterance classification can be changed and the error rate seen from these types of phrases lessened. This voting can also help deal with phrases that were not common to how an individual group normally talks. It makes sense that occasionally a legislator may say something very "non-legislator". If the classifications associated with their entire set of diarized utterances is primarily legislator based, however, we can still correctly classify it. We note that this work assumes diarization is 100% accurate. In real use, this diarization would not be perfect, and the accuracy of the classifier would decrease. As the diarization currently being used by Digital democracy is above 90%, however, we did not explore the effect of its error in this work.

Diarization Voting: This ability to vote on an entire set of utterances leaves room for us to tweak the classifier towards our research preferences. Consider how a voting process is normally done on a binary decision. Normally, one would think that a majority vote, at least fifty percent of the votes, would decide the winner one way or

another. This was what we originally attempted when examining the large hearing dataset. It actually helped a substantial amount, but made us notice an interesting trait about the classifier.

With this form of voting, the true negative rate of the classifier, or the number of non-legislators that were correctly being identified, was extremely high. In fact, it was on average at 99%. This means that almost all of the people that were not legislators were being identified very well. This is great, except when you consider that the lack of perfect accuracy comes from legislators being classified incorrectly. It also is an interesting commentary about the way legislators speak. It suggests that many of the things that legislators say are in line with what a normal member of the public would say. This is not always true of course, because then a classifier to distinguish the two would be useless.

The goal of VFT in relation to the Digital Democracy project is to correctly identify the legislators involved in a hearing. This current classification, however, completely removed some legislators from the process before they could even enter the main part of the classification. This is the worst possible scenario, as discarding their data immediately makes their classification incorrect for the entire VFT process.

While a high accuracy overall is still desired, we would ideally have the opposite case than what was previously observed. Namely, all of the legislators being correctly identified - meaning the classifier has a high recall value - even at the cost of some non-legislators being incorrectly classified - lowering the precision. Re-examining the voting process we have the ability to do just that. By changing the threshold at which the vote swings, we can increase the recall of the classifier at the cost of a lower rate of correct non-legislator classifications.

Table 5.3 shows the effect of the different voting thresholds that we examined using the Bernoulli Naive Bayes classifier with featureset nine from 5.2. The table shows

Table 5.3: Voting Threshold Comparison

	Small			Large		
Threshold	Accuracy	Recall	Precision	Accuracy	Recall	Precison
None	81.2%	66.9%	90.5%	80.2%	60.3%	91.9%
1/2	86.0%	77.9%	89.1%	80.3%	68.4%	92.2%
1/3	87.3%	78.5%	89.1%	85.4%	77.2%	92.2%
1/4	88.6%	84.1%	90.2%	88.4%	83.8%	91.9%
1/5	86.0%	82.0%	87.9%	89.5%	90.0%	88.6%
1/6	90.0%	90.4%	87.3%	91.9%	94.9%	89.9%
1/7	87.8%	90.0%	85.8%	90.2%	93.0%	87.2%
1/10	90.0%	98.9%	81.3%	86.6%	97.5%	78.8%
Any	66.4%	98.7%	53.4%	75.5%	98.4%	68.2%

the effect voting had on the small number of hearings seen to the left, and the large number of hearings on the right. As you can see, the voting change from no voting to at least a majority vote had a decent impact on the recall, but the accuracy only increases a small amount on the larger dataset. As we change the voting threshold in favor of legislator classification - making the number of required votes be a lower and lower fraction of the total grouped utterances - the accuracy of the classifier goes up as well as the recall rate. The large dataset shows the data trend clearly, with the smaller dataset not being as clear-cut. This makes sense in that the smaller dataset is more susceptible to error and has less training data to begin with.

As the threshold changes, it affects the precision of the classifier more and more due to the increased non-legislators being classified incorrectly. At the threshold of 1/6, the average accuracy peaks and starts declining again. For the purposes of the research this work is doing, this provides a good place to hold the voting threshold. In terms of the use of the VFT process, this threshold could be further increased

to really max out the recall of the classifier. A threshold of 1/10 seems to provide a decent overall classifier accuracy while having a recall rate much closer to 100%. This work chose to stick with the 1/6th voting threshold in its experiments for the comparisons of all of the featuresets, datasets and algorithms in the following section.

Chapter 6

RESULTS

This section discussed the results that were found in examining the various discussed featuresets and algorithms. It also looks at the benefits of additional processing, the overall best configuration of features, the impact text has in the VFT process, and finally the answers to the research question we previously proposed.

6.1 Featureset Evaluations

For every dataset that we examined, there is an accompanying data table and range chart. The data table is laid out with the examined feature set on the y-axis, and the type of algorithm used on the x-axis. Within each cell are three numbers. The first is the average accuracy that was seen using that classifier, the second the average recall, and the third the average precision. These averages were taken across several different runs of the classifier while randomly choosing which hearings were used in training and testing.

For these experiments, there was no particular settings that needed to be specified for Bernoulli or Multinomial Naive Bayes. The Maximum Entropy Algorithm was found to converge fairly quickly, and was set to run through 20 iterations. After examining the various types of Support Vector Machines, we used the Linear SVM variant. Finally, the Decision Tree Algorithm was set to a minimum entropy cutoff of .2 and a depth cutoff of 20 (except for the full database dataset which needed to to changed to 10 for time constraints).

The range chart associated with each dataset tries to give an example of the spread a given algorithm had for that dataset. With a different range for each individual

Table 6.1: Results of the experiments for the One Committee Dataset

	Bernoulli NB	Multinomial NB	Max Entropy	SVMs	Decision Tree
One	0.90 / 0.93 / 0.87	0.69 / 0.74 / 0.81	0.60 / 0.53 / 0.88	0.50 / 1.00 / 0.46	0.58 / 1.00 / 0.56
Two	0.88 / 0.90 / 0.90	0.50 / 0.76 / 0.61	0.79 / 0.74 / 0.91	0.40 / 1.00 / 0.37	0.54 / 1.00 / 0.53
Three	0.89 / 0.99 / 0.82	0.66 / 0.74 / 0.77	0.92 / 0.92 / 0.91	0.82 / 0.90 / 0.84	0.58 / 0.93 / 0.56
Four	0.83 / 1.00 / 0.76	0.69 / 0.78 / 0.78	0.86 / 0.82 / 0.92	0.56 / 1.00 / 0.53	0.66 / 0.86 / 0.70
Five	0.76 / 0.68 / 0.88	0.56 / 0.29 / 0.93	0.49 / 0.34 / 0.94	0.61 / 0.98 / 0.58	0.34 / 0.98 / 0.34
Six	0.82 / 0.98 / 0.78	0.56 / 0.47 / 0.81	0.66 / 0.53 / 0.95	0.64 / 0.97 / 0.64	0.84 / 0.96 / 0.84
Seven	0.89 / 0.99 / 0.82	0.58 / 0.73 / 0.72	0.86 / 0.83 / 0.94	0.66 / 0.94 / 0.65	0.45 / 0.97 / 0.45
Eight	0.88 / 0.99 / 0.81	0.66 / 0.82 / 0.72	0.87 / 0.93 / 0.82	0.67 / 0.94 / 0.65	0.85 / 0.94 / 0.85
Nine	0.87 / 0.99 / 0.81	0.64 / 0.85 / 0.69	0.63 / 0.58 / 0.88	0.56 / 0.96 / 0.55	0.69 / 0.99 / 0.68

run of a featureset, the best way to display the overall range of the classification algorithm was to average the lowest value from each featureset experiment of an algorithm together, and similarly an average of the highest values. These ranges are important because they depict how realistic the average accuracy is. If a given range is small, then the accuracy we are seeing is consistent given various testing data, and is much more useful. A large range means that the accuracy varies a lot depending on what kind of test data we see. Some may work very well, but others may work very poorly. Keep in mind that the highest value in the spread may be higher than the values seen in the results table as the results table numbers are an average taken from the numbers used to make the spread.

6.1.1 One Committee Dataset

This dataset is actually very small compared to the other datasets. With only 1756 utterances to work with, there is not a large amount of information for the classifiers to train on. This no doubt affected the accuracy, but the range of results shown by Figure 6.1 in particular shows the impact of such a small amount of training data. As

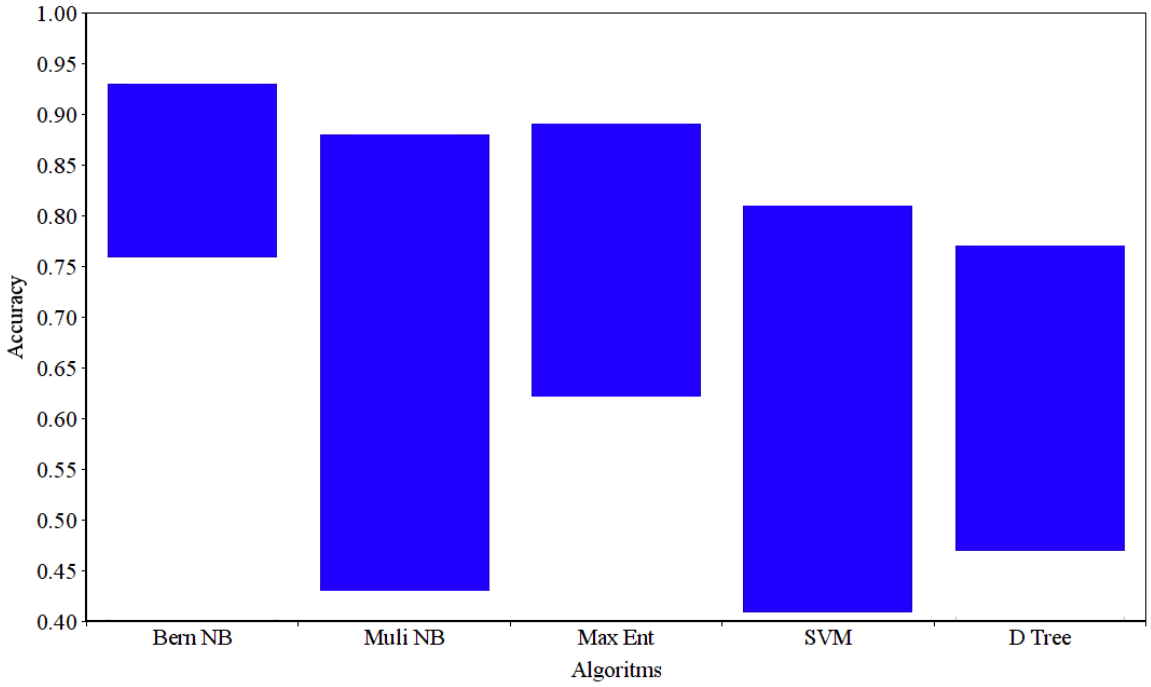

Figure 6.1: The spread seen for the algorithms of the One Committee Dataset

about 1/5ths of this is used as testing, "bad utterances" are particularly troublesome and hard for the classifiers to overcome. The number of legislator utterances is also not even with the number of non-legislator utterances. With only 667 compared to 1089 utterances, the classifier will not be able to identify what represents a legislator as well as it would a non-legislator. All of this together provided another "noisy" dataset as we have denoted the vaccine bill to be.

Algorithm: This dataset has many interesting quirks spread across its experiments as shown in table 6.1. The best accuracy was achieved by the Maximum Entropy Algorithm using Unigrams, Bigrams, and Trigrams (featureset 3). It didn't have a particularly high recall, however, with a score of 92%. Multinomial Naive Bayes, SVMs, and the Decision Tree classifiers performed poorly across the board. Considering the limitations we discussed about this dataset, this is undestandable. Bernolli Naive Bayes was the one algorithm that truly performed well. In particular,

Table 6.2: Results of the experiments for the Vaccine Dataset

	Bernoulli NB	Multinomial NB	Max Entropy	SVMs	Decision Tree
One	0.80 / 1.00 / 0.74	0.82 / 1.00 / 0.77	0.86 / 1.00 / 0.80	0.84 / 1.00 / 0.75	0.81 / 1.00 / 0.68
Two	0.77 / 1.00 / 0.71	0.82 / 1.00 / 0.75	0.75 / 1.00 / 0.68	0.86 / 1.00 / 0.80	0.74 / 1.00 / 0.69
Three	0.84 / 1.00 / 0.78	0.84 / 1.00 / 0.78	0.83 / 1.00 / 0.78	0.81 / 1.00 / 0.75	0.81 / 1.00 / 0.73
Four	0.82 / 1.00 / 0.75	0.78 / 1.00 / 0.72	0.79 / 1.00 / 0.72	0.79 / 1.00 / 0.71	0.79 / 1.00 / 0.69
Five	0.81 / 1.00 / 0.73	0.77 / 0.93 / 0.77	0.87 / 1.00 / 0.72	0.81 / 0.99 / 0.77	0.82 / 0.99 / 0.73
Six	0.79 / 1.00 / 0.72	0.78 / 1.00 / 0.70	0.86 / 1.00 / 0.84	0.79 / 1.00 / 0.67	0.71 / 1.00 / 0.64
Seven	0.78 / 1.00 / 0.71	0.83 / 0.99 / 0.75	0.75 / 1.00 / 0.71	0.80 / 1.00 / 0.70	0.80 / 1.00 / 0.71
Eight	0.81 / 1.00 / 0.72	0.83 / 1.00 / 0.80	0.80 / 1.00 / 0.77	0.86 / 1.00 / 0.81	0.82 / 1.00 / 0.71
Nine	0.85 / 1.00 / 0.81	0.85 / 1.00 / 0.79	0.86 / 1.00 / 0.77	0.78 / 1.00 / 0.71	0.87 / 1.00 / 0.78

consider featureset three and seven which had the same averages across the board. While their accuracy is 3% lower than the high Maximum Entropy value we saw, the recall has an extremely high 99%. As recall is an important consideration in our research, these experiments could be considered the best for this dataset.

Spread: The spread for all the algorithms, shown in Figure 6.1, is fairly large for almost every dataset. The Bernoulli Naive Bayes algorithm had the highest values achieved as well as the tightest range. Maximum Entropy is then next best, which supports what we examined in the results earlier. While most of the algorithms would seem to be fairly unreliable, the Bernoulli spread shows that even with this small size of dataset and unforeseen noise, a decent classification is still achievable.

6.1.2 Vaccine Dataset

Considered to be the designated "noisy" dataset, the Vaccine dataset turned out to be classified pretty evenly across all the algorithms examined. The dataset had a total of 8571 utterances, and a fairly even split of legislator and non-legislator utterances (4234 vs 4337 respectively). This is still considered a fairly small dataset, but is much

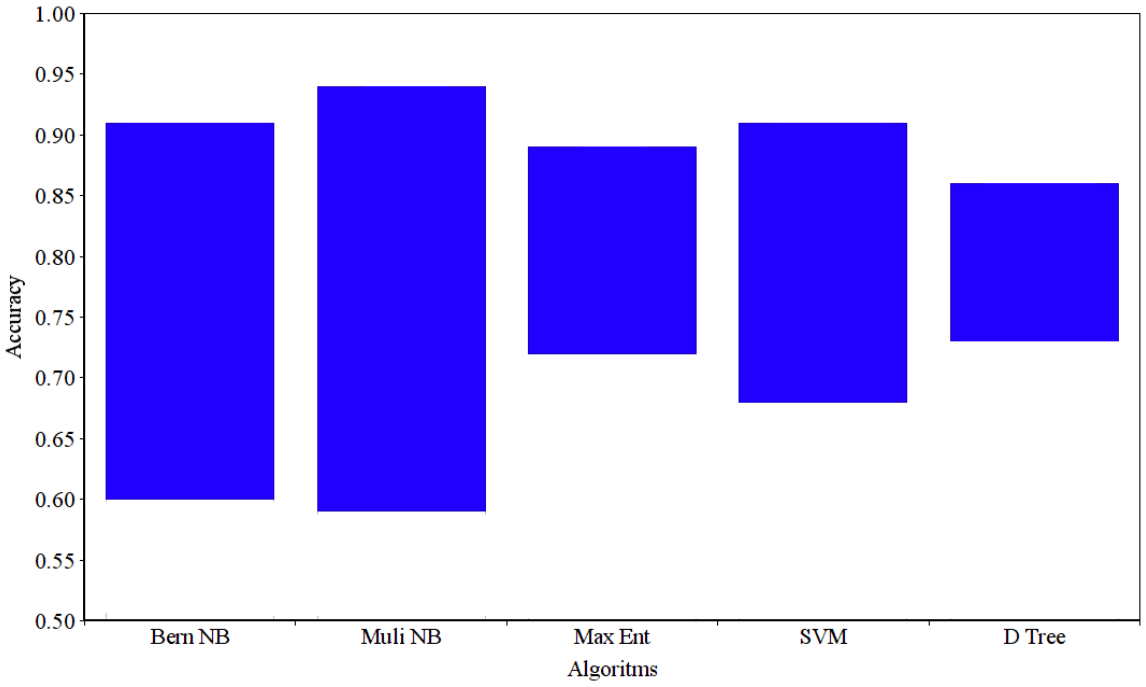

Figure 6.2: The spread seen for the algorithms of the Vaccine Dataset

bigger than the One Committee dataset.

Algorithm: In terms of the best algorithm, it is hard to determine which one performed best off accuracy alone. Referencing table 6.2, the Decision Tree classifier generally performs worse than the other algorithms, but has the highest overall accuracy out of everything in featureset experiment nine. Otherwise, every algorithm has some featureset experiment with 85-86% accuracy. Featureset nine seems to be the best when considering all algorithms.

The noise of this dataset can clearly be seen in the recall of almost every featureset experiment. Almost every single one has 100% recall. As non-legislators were expected to talk like legislators in this dataset, this occurrence makes perfect sense. While we want 100% recall, we don't want to have just classified everything as a legislator without due reason. This makes us examine the precision of the experiments. Many of the precision values are in the 70's and reach up to the low 80's in the best

case. These values are reasonable enough considering our diarization voting scheme. One experiment that stands out above the rest is the Maximum Entropy algorithm paired with featurset 6 - the main feature being the use of the utterances before and after the current utterance. With an 84% precision rate, the use of this feature could indeed be useful when examining noisy datasets. It is hard to say for certain, however, as it is only seen in one of the five algorithms. With no clear best algorithm, we examine the spread of the values seen for each.

Spread: Figure 6.2 clearly breaks the tie of which algorithm is best. The Maximum Entropy Classifier out performs the other classifiers in this dataset because of its consistency. While the Decision Tree Classifier has a smaller overall spread, the lowest value in the spread is comparable with the lowest value Maxent observes. Because the lowest values are almost the same, the bigger spread is better - the algorithm can reach higher overall accuracies.

SVMs could be argued to be better than Decision Trees as well. While it has a higher maximum possible accuracy than Maxent or Decision Trees, it also has a lower minimum accuracy. This difference is enough to make is seem slightly worse than Maxent, but almost on equal footing with Decision Trees. Both Naive Bayes classifiers come in last place. This suggests that they are not as efficient at handling a particularly noisy dataset. These findings are reasonable when you consider the nature of Naive Bayes. These algorithms focus on the presence of given features for differentiation, while algorithms such as Maximum Entropy focus and the differences in the features themselves.

6.1.3 Ten Hearings Dataset

This dataset is relatively small with no particular attributes that should influence accuracy, recall, or precision one way or another. With 7513 total utterances, it is

Table 6.3: Results of the experiments for the Ten Hearing Dataset

	Bernoulli NB	Multinomial NB	Max Entropy	SVMs	Decision Tree
One	0.90 / 0.86 / 0.90	0.87 / 0.79 / 0.90	0.94 / 0.87 / 0.96	0.85 / 1.00 / 0.77	0.85 / 1.00 / 0.69
Two	0.92 / 0.88 / 0.90	0.80 / 0.74 / 0.84	0.72 / 0.48 / 0.83	0.67 / 1.00 / 0.62	0.88 / 0.98 / 0.75
Three	0.91 / 0.91 / 0.89	0.86 / 0.74 / 0.96	0.90 / 0.84 / 0.92	0.70 / 1.00 / 0.56	0.86 / 0.98 / 0.78
Four	0.88 / 0.97 / 0.80	0.87 / 0.78 / 0.93	0.88 / 0.78 / 0.87	0.81 / 1.00 / 0.76	0.82 / 0.84 / 0.83
Five	0.90 / 0.80 / 0.95	0.78 / 0.53 / 0.96	0.86 / 0.65 / 0.96	0.83 / 1.00 / 0.74	0.76 / 0.99 / 0.71
Six	0.86 / 0.87 / 0.83	0.79 / 0.57 / 0.90	0.81 / 0.52 / 0.79	0.75 / 0.99 / 0.69	0.75 / 0.90 / 0.62
Seven	0.88 / 0.86 / 0.87	0.91 / 0.82 / 0.96	0.78 / 0.68 / 0.87	0.81 / 0.99 / 0.78	0.86 / 0.99 / 0.78
Eight	0.94 / 0.94 / 0.92	0.90 / 0.83 / 0.92	0.73 / 0.57 / 0.85	0.79 / 1.00 / 0.71	0.75 / 0.88 / 0.65
Nine	0.89 / 0.90 / 0.87	0.87 / 0.78 / 0.93	0.87 / 0.75 / 0.93	0.63 / 0.99 / 0.57	0.76 / 0.95 / 0.63

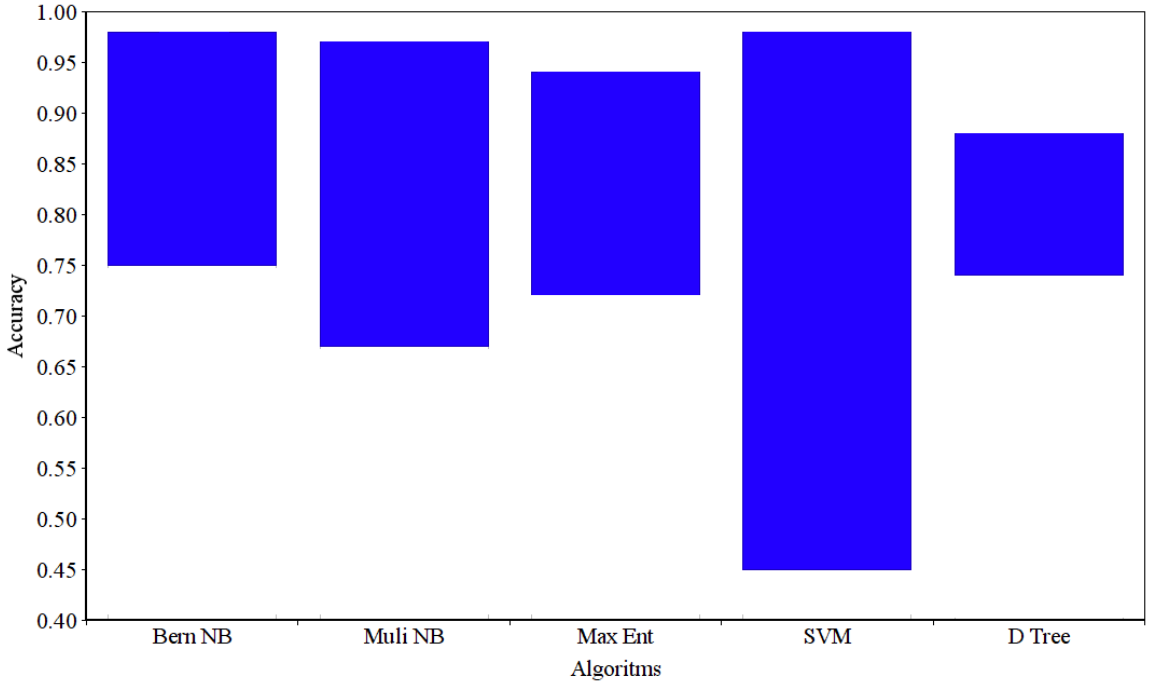

Figure 6.3: The spread seen for the algorithms of the Ten Hearing Dataset

about the same size as the vaccine bill, but should have much less noise. The different legislators seen are much more likely to consist of different individuals, however. Of all the utterances, 2958 were from legislator and 4555 were from non-legislator. If any noise is to be seen in the data, this would be the first quality of the dataset to consider.

Algorithm: For this size dataset, the results (table 6.3) are actually very good. Both Naive Bayes algorithms have accuracies in the range of 86-92%, but the Bernoulli algorithm has consistently high recalls for every featureset. Maximum Entropy does surprisingly well with un-preprocessed unigrams. This is reasonable as it uses entropy as its main calculations - making things more uniform could make the classifier worse. When considering just the NB implementations, featureset eight seems to be the best. Bernoulli NB is best overall in experiment eight in terms of accuracy, recall, and precision with ranges from 92 to 94%. As we can't expect a classifier to do significantly better than this without overfitting, this dataset can be considered categorized extremely well given these parameters.

It is interesting to note the interactions seen in SVMs and the Decision Tree algorithm. While both didn't' have great accuracies, they both had many experiments with high, even 100%, recall rates. This suggests that both mistook non-legislators for legislators often, or the weighting in diarization voting was too heavy. If it is the second case, changing the weighting threshold could improve accuracy, though they would be unlikely to improve past the values seen by the higher performing classifiers.

Spread: The ranges of values seen in Figure 6.3 are fairly even for all algorithms except for SVMs, which is extremely large. The Support Vector Machine range spans approximately from 45% to 97%. That is a range of over 50%! This means that in this case, SVMs are extremely data dependent and were not a very good fit. Otherwise, the spread of the other algorithms is acceptable. While it is a bit bigger than desired,

Table 6.4: Results of the experiments for the VFT Dataset

	Bernoulli NB	Multinomial NB	Max Entropy	SVMs	Decision Tree
One	0.83 / 0.76 / 0.89	0.83 / 0.97 / 0.74	0.84 / 0.78 / 0.84	0.66 / 1.00 / 0.58	0.67 / 0.99 / 0.61
Two	0.81 / 0.72 / 0.89	0.78 / 0.98 / 0.70	0.86 / 0.86 / 0.84	0.62 / 1.00 / 0.56	0.69 / 0.98 / 0.64
Three	0.90 / 0.92 / 0.90	0.85 / 0.98 / 0.78	0.86 / 0.84 / 0.88	0.72 / 1.00 / 0.65	0.75 / 0.98 / 0.68
Four	0.90 / 0.95 / 0.86	0.88 / 0.97 / 0.82	0.90 / 0.93 / 0.87	0.70 / 1.00 / 0.64	0.86 / 0.95 / 0.82
Five	0.80 / 0.64 / 0.94	0.83 / 0.79 / 0.86	0.83 / 0.70 / 0.87	0.70 / 1.00 / 0.64	0.65 / 0.98 / 0.59
Six	0.90 / 0.93 / 0.88	0.88 / 0.97 / 0.84	0.90 / 0.91 / 0.89	0.80 / 1.00 / 0.73	0.56 / 0.98 / 0.53
Seven	0.91 / 0.91 / 0.91	0.88 / 0.95 / 0.85	0.85 / 0.80 / 0.95	0.72 / 1.00 / 0.64	0.51 / 0.99 / 0.48
Eight	0.90 / 0.93 / 0.89	0.86 / 0.96 / 0.81	0.92 / 0.90 / 0.93	0.71 / 1.00 / 0.64	0.82 / 0.98 / 0.76
Nine	0.92 / 0.95 / 0.90	0.84 / 0.98 / 0.77	0.89 / 0.87 / 0.92	0.70 / 1.00 / 0.64	0.51 / 1.00 / 0.47

the spread isn't awful when considering the size constraints of this dataset. As we examine the next two bigger datasets, these spreads should decrease.

6.1.4 VFT Large Hearing Dataset

This dataset represents 61 different hearings, with 30863 utterances. These are split into 15310 legislators and 15553 non-legislators, meaning the information on each group is fairly even. This dataset should follow the same quirks as the Ten Hearing dataset, with the main difference being the size.

Algorithm: The results in table 6.4 are pretty easy to distinguish. Bernoulli Naive Bayes and Maximum Entropy swap back and forth on which has the best accuracy throughout the featureset experiments, with Multinamial Naive Bayes coming in third, SVMs in fourth, and Decision Trees being last. Once again, SVMs and Decision Trees have very high recall rates, but a low accuracy meaning most things are getting classified as a legislator regardless of features. By examining the other three classifiers, a best featureset can't be determined off accuracy alone. Eight and nine

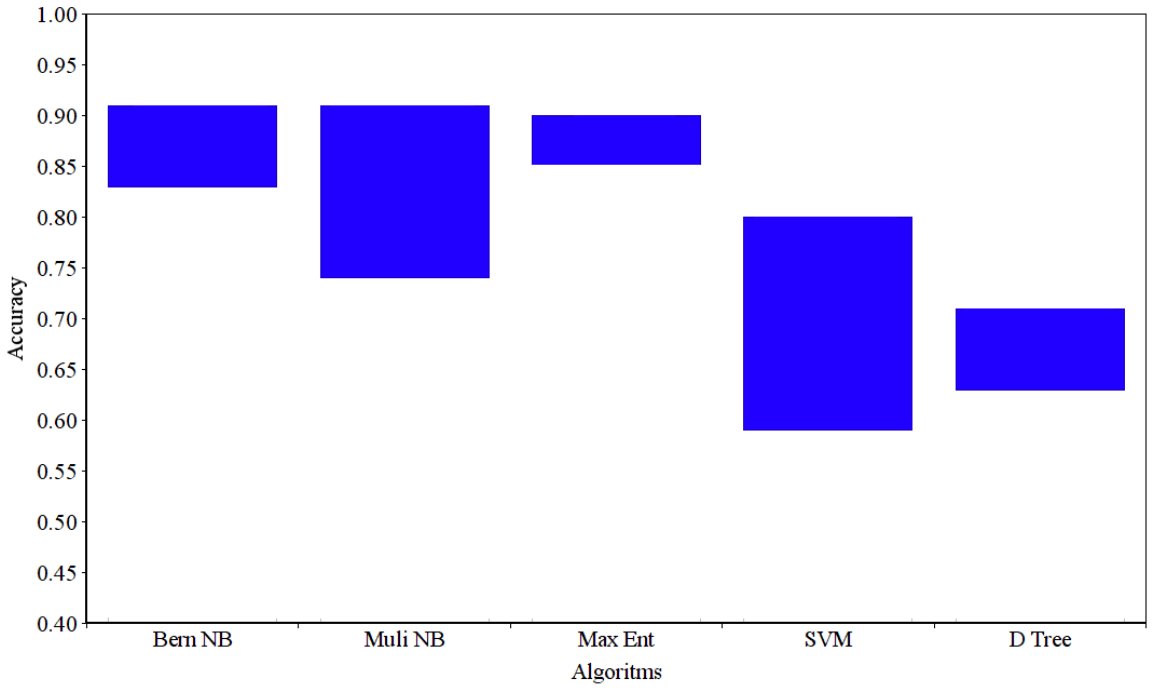

Figure 6.4: The spread seen for the algorithms of the VFT Dataset

are the clear contenders with 92% accuracy in separate classifiers, but featureset nine of Bernoulli NB is considered the best overall due to the high average recall rate.

Spread: While we were able to define the best featureset and algorithm from table 6.4, Figure 6.4 may tell a slightly different story. The highest value for the spread of Bernoulli and Maxent are both about the same, but the total spread of Maxent is much smaller. The question has now become, "what is more important, a smaller spread or a higher recall?" For the use of VFT, we have already defined recall as what is important, but outside the project, it is a question worth considering.

As we commented on in the last spread section, the range of each classifier does indeed seem to have gotten smaller. It is interesting that Decision Trees (disregarding the first small noisy dataset) tend to have small spreads but low accuracies. This could mean that the algorithm is simply not given a big enough depth cutoff to achieve a proper classification.

Table 6.5: Results of the experiments for the Full Database Dataset

	Bernoulli NB	Multinomial NB	Max Entropy	SVMs	Decision Tree
One	0.86 / 0.80 / 0.90	0.86 / 0.98 / 0.78	0.84 / 0.74 / 0.90	0.72 / 1.00 / 0.64	0.71 / 0.99 / 0.64
Two	0.83 / 0.74 / 0.89	0.81 / 0.98 / 0.73	0.82 / 0.73 / 0.88	0.68 / 1.00 / 0.60	0.77 / 0.98 / 0.69
Three	0.93 / 0.95 / 0.91	0.92 / 0.98 / 0.87	0.92 / 0.91 / 0.92	0.80 / 1.00 / 0.71	
Four	0.92 / 0.98 / 0.87	0.91 / 0.98 / 0.86	0.93 / 0.95 / 0.90	0.76 / 1.00 / 0.67	
Five	0.81 / 0.65 / 0.94	0.86 / 0.80 / 0.92	0.81 / 0.68 / 0.91	0.80 / 1.00 / 0.71	
Six	0.92 / 0.95 / 0.90	0.92 / 0.97 / 0.89	0.89 / 0.87 / 0.92	0.84 / 1.00 / 0.75	
Seven	0.93 / 0.95 / 0.91	0.92 / 0.97 / 0.87	0.91 / 0.92 / 0.89	0.80 / 1.00 / 0.71	
Eight	0.93 / 0.96 / 0.91	0.92 / 0.97 / 0.88	0.90 / 0.87 / 0.93	0.78 / 1.00 / 0.69	
Nine	0.93 / 0.96 / 0.91	0.92 / 0.97 / 0.87	0.90 / 0.89 / 0.93	0.77 / 1.00 / 0.68	

6.1.5 Full Dataset

This is the final set of data that we examine in this work. It is the full database dump of the current utterances being used on the Digital Democracy websight. Consisting of 473,174 total utterances, 228,874 of these have been said by a legislator with 244,300 coming from non-legislators. It is hard to say what quirks this dataset might have, but it is of great interest in that it is a very real test of C-SALT's use for the Digital Democracy project.

Algorithm: The first thing immediately visible about table 6.5 is the lack of values in the Decision Tree column. This is because the large amount of data has caused this algorithm to take a very long time to complete. The first featureset for Decision Trees took about 21 days to complete, and that is for the smallest set of features we have. We simply haven't gotten results back for the remaining features (at the time of this writing it has been running for 38 days).

The results themselves show quite a few interesting points. First, both Naive

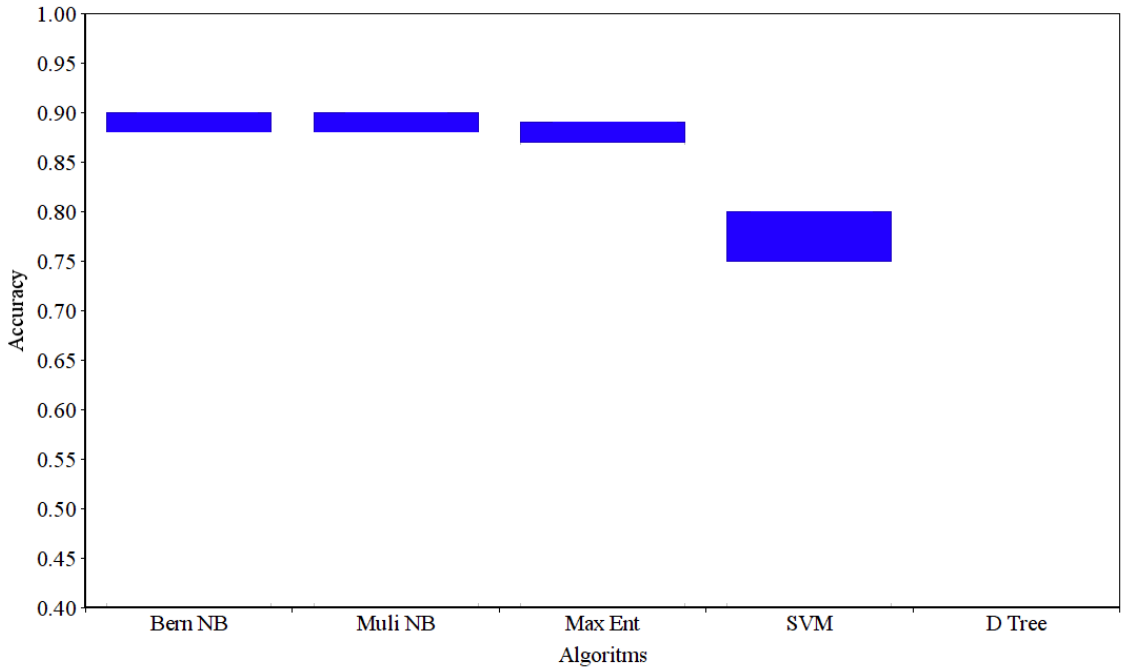

Figure 6.5: The spread seen for the algorithms of the Full Database Dataset

Bayes classifiers and the Maximum Entropy classifier have very similar results given each featureset experiment. When the accuracy for one set goes up, this trend is seen in the other classifiers as well. Overall, there are many different featuresets all with 92-93% accuracy and high recall. The best performing could be considered experiment three as it achieved the same high values as other experiments with less features. Maxent did perform slightly worse than the two Naive Bayes classifiers, as its accuracies and recall values are typically a few percentages lower. This data really shows the impact of having a large amount of data to train with. Given enough examples, and the use of diarization voting, the algorithms are able to classify individuals into each group even with the use of only the more basic features.

Spread: While not really having a spread to show for the Decision Tree classifier is unfortunate, the data on the spread of the other classifiers is quite interesting. The first thing of interest in Figure 6.5 is that both Naive Bayes classifiers and the Maximum Entropy classifier all have the same range. The NB classifiers ranges even start and end at the same numbers, while Maxent is only 1% lower on both the high and low ends. Each individual spread is also very small. The three equivalent ranges only have a margin of 2%. The SVM range is still larger than the others, but is significantly smaller than the ranges observed for the previously smaller datasets. These observations are consistent with our beliefs about the effect of an increased availability of data.

This data suggests that given enough data, there classifiers perform nearly the same in terms of accuracy for our setting. This is important information to have in that we can focus on the classifier with the best computation time and space complexity when choosing which algorithm would work best in a classification system. This would mean that the Bernoulli NB classifier would be the best choice for this dataset.

	Small (10 Hearing)	VFT
One	0.89/0.89/0.86	0.83/0.79/0.87
Two	0.90/0.97/0.83	0.83/0.77/0.89
Three	0.89/0.97/0.79	0.90/0.94/0.87
Four	0.85/0.99/0.78	0.89/0.97/0.84
Five	0.88/0.84/0.85	0.82/0.72/0.91
Six	0.90/0.98/0.84	0.90/0.98/0.85
Seven	0.89/0.94/0.85	0.90/0.96/0.85
Eight	0.94/0.97/0.89	0.91/0.94/0.89
Nine	0.93/0.98/0.85	0.91/0.93/0.89

Figure 6.6: The accuracy, recall, and precision results of running the ensemble classifier on the small ten hearing dataset and the VFT dataset.

6.1.6 Ensemble Classification

After viewing the results we achieved for each dataset, we made the decision to try one more classifier. Namely, we decided to combine the Bernoulli Naive Bayes, Maximum Entropy, and Support Vector Machine algorithms into one Ensemble classifier. This classifier functions by running each classifier on each diarization group as previously done, and then votes between all three algorithms. If at least two of the classifications agree that the overall style should be considered that of a legislator, the the ensemble classifier as a whole makes the legislator style attribution to the diarization utterances.

Figure 6.6 shows the results of running this ensemble classifier on both the small ten hearing dataset as well as on the large VFT dataset. In comparison to the other algorithms ran in the small dataset, the ensemble classifier performs much better. The while the overall accuracy seen doesn't surpass what featureset eight of the Bernoulli

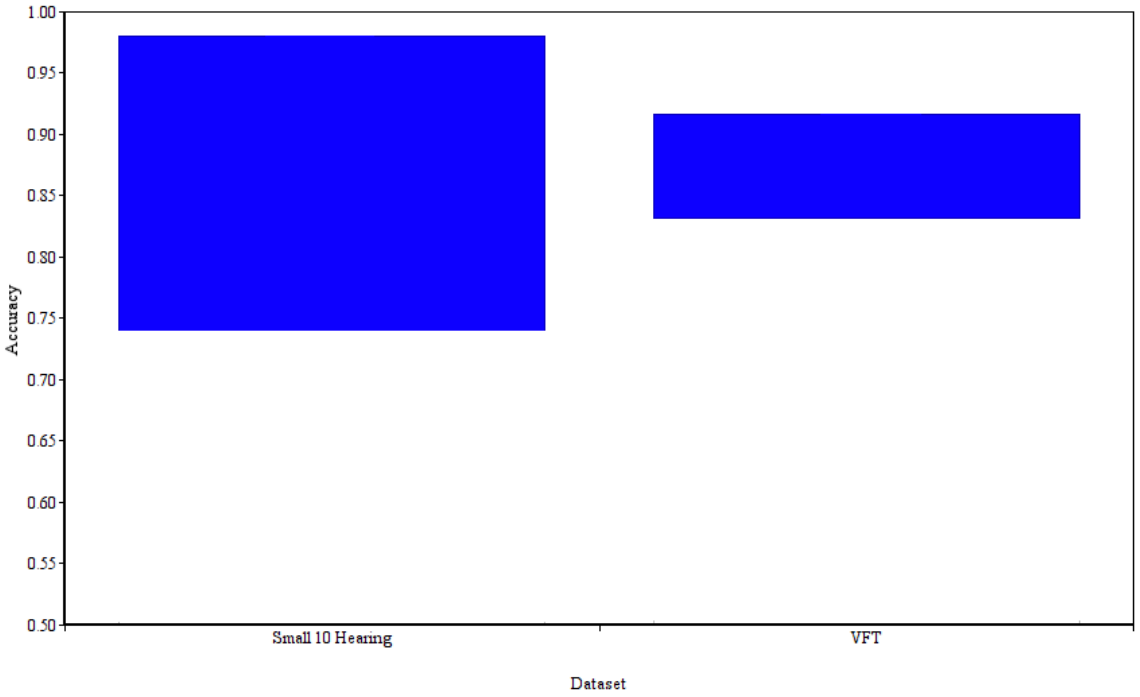

Figure 6.7: The spread of the ensemble classifier.

Naive Bayes classifier, the accuracies seen are significantly higher (94% compared to 98%). While this is great for the smaller dataset, the ensemble classifier didn't seem to be able to improve either the accuracy or the recall of the larger VFT Dataset. Figure 6.7 shows the spread of the accuracy seen across both datasets. Both spreads seen are similar the the spreads examined in the better performing classifiers of the respective datasets.

6.1.7 Overall Conclusions

For most of the datasets, the simplest Bernoulli Naive Bayes algorithm performed the best in terms of accuracy and recall. The Maximum Entropy Algorithm was a close second, and actually tended to perform better in terms of consistency. Given enough data, both Naive Bayes and the Maxent algorithms converge to very similar performances. Surprisingly, Support Vector Machines was one of the worse algorithms

we observed. Its accuracies tended to be better than Decision Trees, but were much less consistent. With that said, it could probably be improved with tuning specific to the algorithm. When initially testing diarization voting, it was in relation to Naive Bayes. A different threshold could impact SVMs greatly.

Decision Trees low performance can be somewhat attributed to the depth cutoff that was given to it. With the amount of features we have across large amounts of data, the depth had to be set low or it would take too long to be a reasonable use option. In terms of speed, Bernoulli NB is much faster than any of the other algorithms. Maximum Entropy and SVMs typically take a few more hours than Bernoulli given the largest set of data. Multinomial Naive Bayes took approximately two days, and Decision Trees still have not completed all their experiments. This alone speaks to how much better Bernoulli Naive Bayes performed in the test setting.

In terms of featuresets, experiments eight and nine were seen performing well consistently, with six being useful in noisy environments. This suggests that on top of n-grams, the selected features and entity recognition are useful in achieving higher recall rates and better classification. Featureset six could also show that there is a correlation with speakers taking turns in conversations, thus giving a feature available in the spoken field that is not explorable in a typical written setting. Many times, the results of unigrams alone or n-grams seen in experiments one, two, and three performed well. This may seem like there is no need for advanced classification techniques at all, but keep in mind the diarization voting process being used. As shown in previous sections, this provides a fairly significant impact to most featuresets. Also, as accuracy gets higher, achieving higher accuracy gets more and more difficult. This fact makes the small improvements the other featuresets provide more impressive than they may seem.

6.2 Effects Of Addition Processing

After testing the various featuresets, we then examined the effect of two post processing classifiers. The addition of them, however, did not have a desirable impact on the results. The accuracy itself stayed almost the exact same given the featureset, while the precision values went up a small margin with the recall values dropping down. As we are trying to remove incorrectly classified non-legislators, the increase in precision is expected. The recall value in turn decreased due to these additional classifiers making errors.

In most cases, the original classification for a diarization was correct. This can be seen from the overall 90% and up accuracies of the later experiments. With this in mind, there are only a few secretaries or lobbyists that have been misclassified. If the post processing classifiers worked perfectly, these misclassified diarizations would be reclassified, the recall rate would remain the same, and the precision rate would increase. Any errors at all in these classifiers would then, however, misclassify an already correct set of utterances for a diarization. While the classifiers we made for post processing were fairly good, above 90% accuracy, the amount of error they had was still too much to warrant their further use.

In our case, lower recall is bad; if higher precision was desired this would have instead been a useful addition. Overall, this suggests that given classified values with previously high accuracy, trying additional processing to fix errors is only worthwhile when: mistakes in the addition process cause little impact to the current results, and you are trying to increase the metric most important to the work.

As treating this classification as a "post processing" technique didn't work as we desired, we attempted to include two new features that would perform the functionality we were looking for, without having to deal with the harsh consequences of errors.

Accuracy	Recall	Presision
0.92	0.92	0.92

Figure 6.8: Effects of changing post processing classifiers to features

The major way secretaries were being classified were through features sush as looking for the phrase "vote is", the presence of many question marks, and basic bigrams. To replace this, we included the "vote is" feature in the full classifier and added one new feature. Namely, we look at the number of times a name is said in an utterance. As secretaries call roll and state votes, they say many names, thus differentiating them from legislators.

Replacing lobbyist classification is even easier. Before, we simply looked for the presence of a lobbyist name followed by the phrase "on behalf". Normally, lobbyists were the only ones that would say their name followed by who they represented. We simply made the presence of "name on behalf" a feature. The results of adding these new features to the Bernoulli Naive Bayes algorithm, based off featureset nine, and using the large VFT dataset is shown in Figure 6.8. Overall, the accuracy stayed the same as in the earlier experiment, the recall went down 5%, and the precision went up 2% (the disproportionally could come from different datasets being examined in the average of the runs). Once again, as accuracy didn't go up and recall went down, these features are not something that we would want to use in the VFT classifier. They are however, viable features when precision is the performance metric looking to be increased.

6.3 Most Informative Feature Reduction

While the broad use of features has shown the ability to achieve a fairly high accuracy, it is important to examine what accuracy we can achieve given a more limited scope

a features. As having more and more features will diminish the importance of each individual feature, less is sometimes more.

To examine the effect of limiting the domain of features, we first must determine what the best features are. The best features may come from different feature extractors, and thus we start by extracting as many features as possible. This includes unigrams, bigrams, and trigrams from a word and letter level, entity recognition, selected features, and the previous/next utterances unigrams, bigrams, and trigrams. From this very large set of features, we then extract the most important aspects.

Originally, we decided to try and use Principle Component Analysis (PCA) to determine which traits were the most important. The idea behind PCA is to take the data it is given, represent it with orthogonal variables (or vectors), and using this information determine which parts of the data have the most influence [1]. Scikit-learn has an implementation of PCA that we attempted to use, but we could not find a way to extract the information we needed without tearing into their code.

Upon examining Scikit-learn's documentation, and the math behind PCA, we attempted our own implementation. There are only two steps that were needed to obtain the information we desired from PCA. First, make a correlation matrix between all the features, then compute the orthogonal vectors (eigenvectors) with the corresponding values representing the importance of each vector (eigenvalues). This would have worked perfectly, except when system memory was considered. As the correlation matrix is an N by N matrix and we have hundreds of thousands of possible features using a big dataset, there is simply not enough RAM (even with 256 gigabytes) to perform this computation, and PCA in tern.

As PCA was not a feasible option, we turned to some of the build-in functionality that NLTK has for Bernoulli Naive Bayes. Namely, NLTK leverages the fact that the features are binary true or false values to quickly calculate which features are

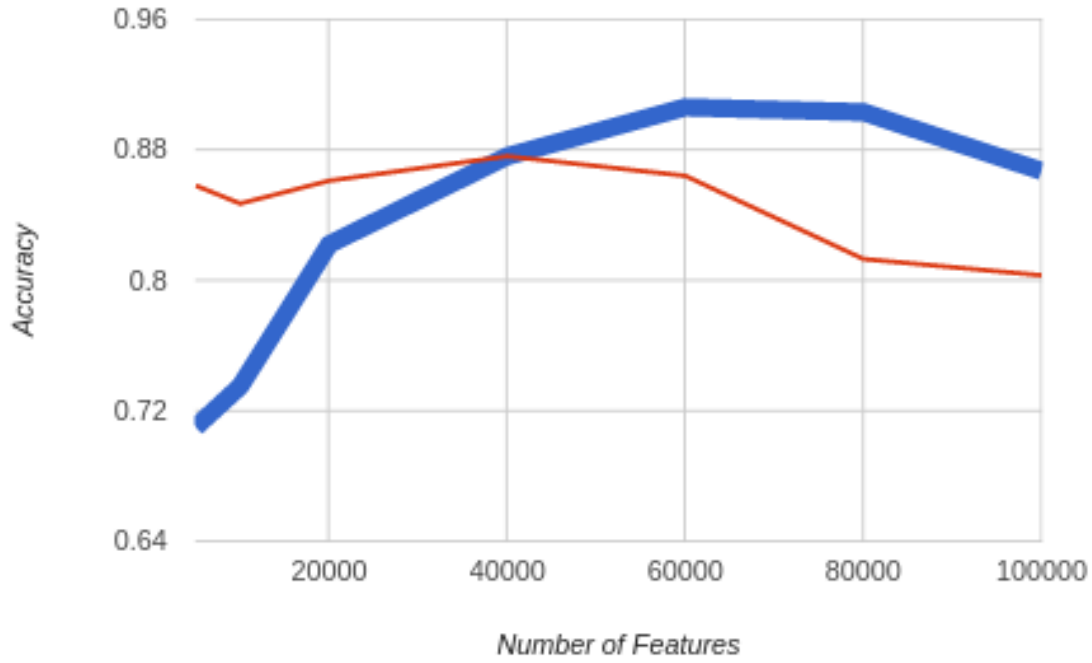

Figure 6.9: The accuracy of the classifier with respect to the number of most important features used. The thick line represents a 1/6th diarization voting threshold while the thin line represents a 1/2 threshold.

the "most informative". This allows us to quickly determine which features are most valuable, and then only consider those in classification. The thick line in Figure 6.9 shows how the increasing number of features effects accuracy of C-SALT. As more features are added, accuracy increases as expected. The accuracy then reaches a plateau around 60,000 features with 90% accuracy. The graph then starts to fall off as extra features that don't provide much information are continued to be added.

The amount of features it took to reach the accuracy plateau was surprisingly high. After reexamining the data, we noticed something surprising. After reaching just the 2000 most important features, recall for the classifier was almost 100% while the precision was miserably low. Figure 6.10 shows the values of precision and recall as features increase corresponding to the thick line in Figure 6.9. This information suggested that the classifier was grossly misattributing non-legislators as legislators

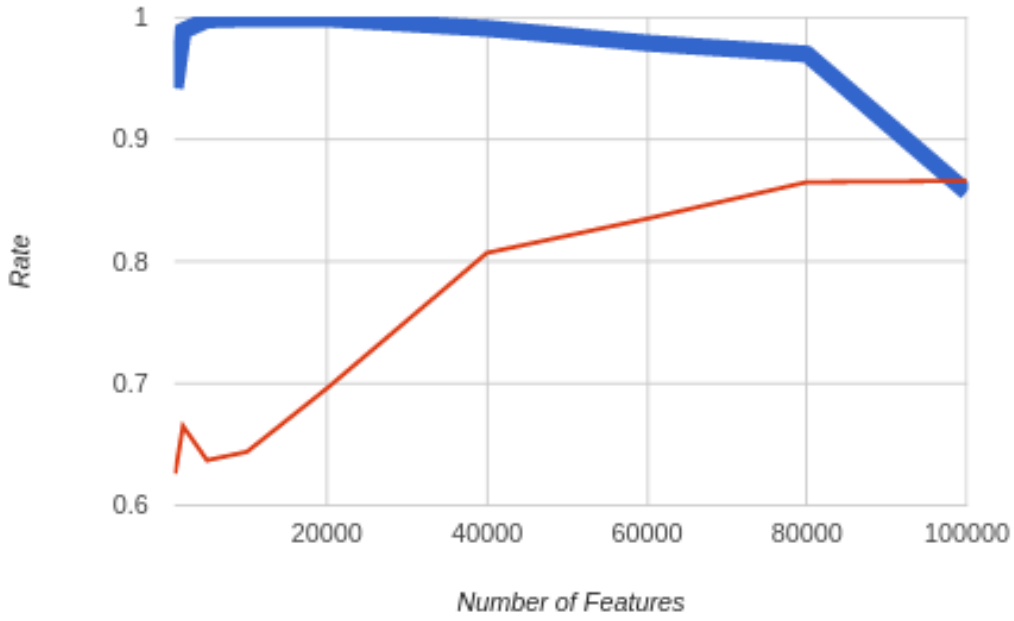

Figure 6.10: Recall (thick line) and precision (thin) in relation to standard 1/6 diarization threshold.

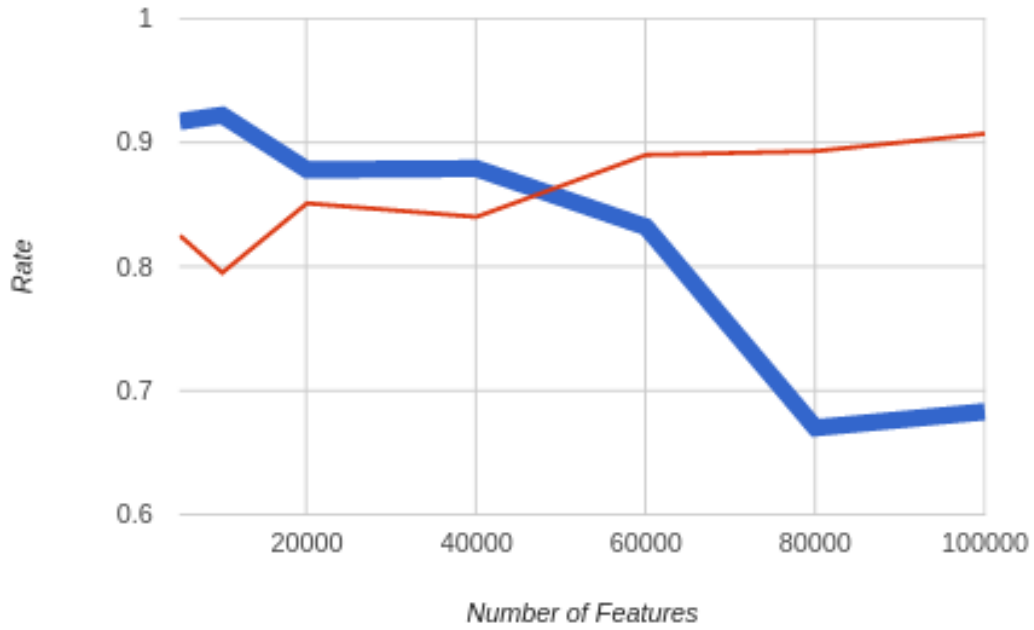

Figure 6.11: Recall (thick) and precision (thin) in relation to 1/2 diarization threshold.

in this experiment. C-SALT, however, has a way to address this issue. Namely, the aforementioned diarization voting threshold. When using all of the data, our experiments showed that a threshold on 1/6th was the best threshold value. Limiting features to those that most distinguish between the two categories could change that.

As the recall value is maximized at such an early number of features, we decided to set the voting threshold at the standard "majority" value of 1/2. The thin line in Figure 6.9 and Figure 6.11 show the results of this change. Lowering the threshold significantly lowered the number of utterances that were needed to reach the accuracy plateau, though the maximum accuracy achieved is lower.

Examining how the most informative features impacts accuracy is important to insure that we are not drowning out the more influential traits in each style category. As neither experiment showed and accuracy above what we were able to achieve in testing, we can conclude that this is not an issue that greatly hinders us. It is interesting to note to amount of features needed to achieve the accuracies we see, however, and shows just how nuanced speaker style can be.

6.4 Research Question Results

Here we readdress the main research question we proposed at the beginning of the paper and answer it given the results we found.

6.4.1 Classification Algorithm

For spoken word attribution, the best algorithm somewhat depends. Surprisingly, Bernoulli Naive Bayes seems to perform the best overall when designating individuals between two groups. For datasets that had more noise, Maximum Entropy seems to be a better option as it generally has a lower spread than the Bernoulli algorithm.

As the dataset being used gets bigger, the differences in the algorithms decreases, making Bernoulli NB the best option due to time and space complexity.

Multinomial Naive Bayes tends to receive good results as well, but the spread of the results that are received tend to be on the larger side which is undesirable. In general, SVMs and Decision Trees did not perform very well in out experiments. This is somewhat surprising considering what our background research found. We believe SVMs could be improved to work better with specific parameter tuning, but would not be significantly better than the algorithms we already have. Decision trees might be able to be improved with more features or longer depth cutoffs, but the amount of time is takes to run makes a bigger depth infeasible on bigger datasets.

6.4.2 Extracted Features

In terms of features used in spoken word attribution, there are several that build on top of each other. The basis that should be started with are the use of Unigram, Bigrams, and Trigrams. We then found that Entity Recognition in various forms, utterance locality with respect to surrounding utterances, and selected features specific to the data to be the most useful additions on top of the Ngrams. With the diarization voting working in the background, even Unigrams all on their own worked fairly well. Letter level Ngrams, stemming, and stopword removal typically lowered accuracy in this setting, as well as increased computation time. These findings are relevant to the legislator setting, however, and backgrounds of other data may find use for these feature.

6.4.3 Effectiveness with Varied Data

Text classification of spoken language performed well across the datasets that we examined. In out most constricted dataset, with low data amount and medium noise

(One Committee), we were able to achieve a best average of 89% accuracy with a 99% recall. Our other small dataset with high noise (Vaccine Bill) still had a reasonable high max accuracy of87% with recall of 100%. Note that these values were not achieved with the same classifier, but are achievable nonetheless. All of the other datasets that we examined go up from there, with the larger sets getting higher and higher accuracies until a max of about 93%.

6.4.4 Overall Speaker Identification Improvement

Here we examine the overall contribution that C-SALT can provide to various other classifiers. First we examine how much C-SALT can improve a text classifier that is trying to identify an individual legislator. While it is not a classifier that is used in the VFT speaker recognition process, it provides evidence of C-SALT's overall use as an additional tool to be used for increasing the accuracy of other classifiers. We then examine the facial and audio classifiers ability to identify legislative speakers with and without C-SALT.

Text Speaker Identification: A Bernoulli Naive Bayes text classifier that tries to identify the identity of a legislator was created to test the benefits of the thesis. When given data, it trains on each different legislator as its own category, with all of the non-legislators being grouped into a single catch all category. By doing this, the classifier will ideally filter out non-legislators on its own into the catch all category. Otherwise, each utterance should hopefully be identified with the correct legislator that corresponds to it.

With initial testing, we knew that this classifier would perform poorly. Being able to identify the nuances in each legislators speech off words alone simply was not feasible given the number of people that are being considered. This classifier, once again, is just being used to show the improvement C-SALT provides to such

a classifier. On its own, the text classifier cannot correctly attribute a legislator to a diarization id. In fact, the classifier can barely classify any of the utterances correctly. Even though many of the same features are used when extracting features from the public utterances, with so many possible categories to place an utterance in, the classifier almost always fails to classify them correctly into the catch all category. More simply, many of these non-legislator utterances are classified as a single incorrect legislator identity.

On its own without using C-SALT, the text classifier only achieves an accuracy of 00.63% across all utterances on the large VFT dataset. When C-SALT is used to filter out the legislators before the individual identity classification occurs, the accuracy is increased to 01.18%. This is an 87.3% increase in accuracy. While the overall accuracy is still terrible, the increase from the original classification is still rather significant. This accuracy mainly comes from the fact that there are less non-legislators for the classifier to label incorrectly. These results shows that C-SALT definitely provides a benefit in the text identification field.

Audio Speaker Identification: The voice classifier tries to filter out legislators on its own to a certain degree, similar to the catch all category in the text speaker identifier. As C-SALT cannot perfectly filter out non-legislators, this is needed to help with their accuracy for those incorrect utterances that slip through. This allows us to compare the accuracy of this classifier with and without C-SALT. In comparing the results, we ensure that the same utterances are used for both training and testing in each run.

When ran alone, the audio classifier that uses SVMs is able to correctly identify a legislator 44.0% of the the time. Considering that this is the individual identity of a legislator across about 120 people, this is actually a fairly decent accuracy. C-SALT, however, can improve this number. With its inclusion, the accuracy of the

classifier increased to 66.4% (with C-SALT performing at 89% accuracy). That is a 50.9% increase from the original accuracy. Here too, C-SALT shows that it provides a significant benefit to the accuracy of speaker identification, this time in an audio medium.

Facial Speaker Identification: Facial recognition similarly has a mechanism for filtering out non-legislators that may slip through, and so we evaluate its classification ability with and without C-SALT. Alone, the face classifier was only able to achieve 30.6% accuracy. When using C-SALT, identification of a speaker climbs to 46.4% (with C-SALT performing at the same 89.5% accuracy). This is a 51.6% increase from the original accuracy. It is interesting to note that C-SALT improves both facial and audio recognition by about 50%, which may be related to the fact that we are reducing the number of utterances considered by approximately 50% as well. These results show C-SALT is useful to increasing facial classification. Along with the results from audio and text classification, we have shown that regardless of classification field, C-SALT is a useful addition to speaker identification.

Chapter 7

CONCLUSION

7.1 Future Work

Here we discuss additional research that could be continued off this work. This is by no means a comprehensive list, but merely suggests some of the possible avenues to be explored.

7.1.1 Further Analysis

At the base level, future research could look into additional classification algorithms and features could be examined. There are various modifications that can be made to individual algorithms, and this work tried to pick the best ones given minor testing. Given the right tuning, however, it is possible that different versions of algorithms may preform better. Feature selection can be spread much further than we have taken it.

The use of parts of speech, for example, could prove to be insightful into style classification. Sentiment of a given topic could be relevant as well, especially in a legislative setting. If a certain bill has many legislators advocating it, and many non-legislators against it, then the sentiment of the utterances being examined would be extremely valuable. The speed of text classification could also be increased by running certain aspects of the code in parallel. While the overall time to run most of the datasets isn't very long, the option to improve is still there.

Other techniques could also be included. The use of Ada Boosting, for example, could prove useful. With many weak learning algorithms good at classifying certain aspects of the data and then coming to a consensus, we may get a more accurate or

consistent result. In line with multiple classifications, we could adapt this system to separate legislator and non-legislators, but also several groups within the legislature. As we have already looked at, identifying lobbyists or secretaries is possible, and other groups certainly exist.

7.1.2 Additional VFT Uses

In terms of VFT classification, a text classifier could also be used to help with decision making between the various classifiers. Facial and audio recognition do not always agree on the identity of an individual, and need a way to come to a consensus. There are times when the number one choice for one classifier may be number two in another, and vice-versa. With a strong enough text classifier on just these two individuals, ties between classifiers could be broken.

7.1.3 Other Areas

Finally, different settings other than the California State Legislature could be examined. Classifying the different styles of groups could extend, in the legislative sense, to other states or countries. It could extend to police work by classifying the style that a guilty person speaks in compared to an innocent one. This work could be applicable to many setting that need to separate individuals into groups. Different algorithms or features would most likely be needed, but the general principle should hold.

7.2 Final Thoughts

We examined the feasibility of using the presence of a given style to classify speech of people in the California State Legislature. After experimenting with various algo-

rithms, methods of feature extraction, and datasets, we have shown that this type of conversational style classification is achievable and worthwhile. Given enough background information, we are able to preform this task with over 90% accuracy.

In the overall scope of speaker identification, the inclusion of C-SALT significantly narrows the field of entities in the dataset that need to be classified. This greatly increases the accuracy of both facial and audio classification, allowing for an overall higher accuracy than either could have achieved on their own. While C-SALT doesn't perfectly extract every legislator in a given data corpus, the losses incurred are still outweighed by the benefits of so many non-legislators being removed from consideration. With an overall increase in facial recognition from 30.6% to 46.4% and audio recognition from 44.0% to 66.4%, there is no doubt that C-SALT is an essential addition to the VFT process, and greatly improves speaker identification accuracy as a whole.

BIBLIOGRAPHY

[1] H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[2] M. N. Anyanwu and S. G. Shiva. Comparative analysis of serial decision tree classification algorithms. *International Journal of Computer Science and Security*, 3(3):230–240, 2009.

[3] H. Azarbonyad, M. Dehghani, M. Marx, and J. Kamps. Time-aware authorship attribution for short text streams. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 727–730, New York, NY, USA, 2015. ACM.

[4] S. Bird. Nltk: The natural language toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, COLING-ACL '06, pages 69–72, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[5] G. Bordogna and G. Pasi. A fuzzy linguistic approach generalizing boolean information retrieval: A model and its evaluation. *Journal of the American Society for Information Science*, 44(2):70, 1993.

[6] J. P. Campbell. Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, Sep 1997.

[7] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo. Support vector machines for speaker and language recognition. *Computer Speech & Language*, 20(2):210–229, 2006.

[8] C. E. Chaski. Whos at the keyboard? authorship attribution in digital evidence investigations. *International Journal of Digital Evidence*, 4(1):1–13, 2005.

[9] J. Chen, H. Huang, S. Tian, and Y. Qu. Feature selection for text classification with naive bayes. *Expert Systems with Applications*, 36(3, Part 1):5432 – 5435, 2009.

[10] H. L. Chieu and H. T. Ng. A maximum entropy approach to information extraction from semi-structured and free text. *AAAI/IAAI*, 2002:786–791, 2002.

[11] M. Cristani, G. Roffo, C. Segalin, L. Bazzani, A. Vinciarelli, and V. Murino. Conversationally-inspired stylometric features for authorship attribution in instant messaging. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 1121–1124, New York, NY, USA, 2012. ACM.

[12] J. Diederich, J. Kindermann, E. Leopold, and G. Paass. Authorship attribution with support vector machines. *Applied Intelligence*, 19(1):109–123.

[13] D. Doyle. Stopword lists. http://www.ranks.nl/stopwords, 2000.

[14] H. Drucker, D. Wu, and V. N. Vapnik. Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on*, 10(5):1048–1054, 1999.

[15] J. Goldstein-Stewart, R. Winder, and R. E. Sabin. Person identification from text and speech genre samples. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 336–344, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[16] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, Jul 1998.

[17] D. I. Holmes. Authorship attribution. *Computers and the Humanities*, 28(2):87–106, 1994.

[18] D. I. HOLMES. The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing*, 13(3):111–117, 1998.

[19] G. Hripcsak and A. S. Rothschild. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298, 2005.

[20] A. Jucker. *Social Stylistics: Syntactic Variation in British Newspapers*. Topics in English linguistics. Mouton de Gruyter, 1992.

[21] P. Juola and J. Sofko. Proving and improving authorship attribution technologies. In *Proceedings of Canadian Symposium for Text Analysis (CaSTA)*, 2004.

[22] T. M. Khoshgoftaar and E. B. Allen. Controlling overfitting in classification-tree models of software quality. *Empirical Software Engineering*, 6(1):59–79, 2001.

[23] F. Khosmood. Automatic source attribution of text: A neural networks approach. Master's thesis, Cal Poly San Luis Obispo, 2005.

[24] F. Khosmood. *Computational Style Processing*. PhD thesis, University of California Santa Cruz, 2011.

[25] R. Kohavi and F. Provost. Glossary of terms. *Machine Learning*, 30(2-3):271–274, 1998.

[26] M. Koppel, J. Schler, and S. Argamon. Authorship attribution: What's easy and what's hard? June 2013.

[27] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas. Supervised machine learning: A review of classification techniques. 2007.

[28] D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In C. Ndellec and C. Rouveirol, editors, *Machine Learning: ECML-98*,

volume 1398 of *Lecture Notes in Computer Science*, pages 4–15. Springer Berlin Heidelberg, 1998.

[29] T. Lidy, A. Rauber, A. Pertusa, and J. M. I. Quereda. Improving genre classification by combination of audio and symbolic descriptors using a transcription systems. In *ISMIR*, pages 61–66, 2007.

[30] B. Liu. *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Science & Business Media, 2007.

[31] E. Loper and S. Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[32] K. Luyckx and W. Daelemans. Authorship attribution and verification with many authors and limited data. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 513–520, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[33] J. Mayfield and P. McNamee. Single n-gram stemming. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 415–416, New York, NY, USA, 2003. ACM.

[34] A. McCallum, K. Nigam, et al. A comparison of event models for naive bayes text classification. Citeseer, 1998.

[35] F. Mosteller and D. L. Wallace. *Applied Bayesian and classical inference: the case of the Federalist papers*. Springer Science & Business Media, 2012.

[36] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[37] N. Nitta and N. Babaguchi. Automatic story segmentation of closed-caption text for semantic content analysis of broadcasted sports video. In *Multimedia information systems*, pages 110–116, 2002.

[38] C. D. Paice. An evaluation method for stemming algorithms. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 42–50, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[39] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, Nov. 2011.

[41] D. M. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[42] V. V. Raghavan and S. M. Wong. A critical analysis of vector space model for information retrieval. *Journal of the American Society for information Science*, 37(5):279, 1986.

[43] J. Ramos. Using TF-IDF to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.

[44] D. A. Reynolds and P. Torres-Carrasquillo. Approaches and applications of audio diarization. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 5, pages v/953–v/956 Vol. 5, March 2005.

[45] S. R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. 1990.

[46] G. Salton, E. A. Fox, and H. Wu. Extended boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036, 1983.

[47] J. Savoy. Authorship attribution based on specific vocabulary. *ACM Trans. Inf. Syst.*, 30(2):12:1–12:30, May 2012.

[48] J. Savoy. Feature selections for authorship attribution. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 939–941, New York, NY, USA, 2013. ACM.

[49] E. Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556, 2009.

[50] G. Tambouratzis, S. Markantonatou, N. Hairetakis, M. Vassiliou, G. Carayannis, and D. Tambouratzis. Discriminating the registers and styles in the modern greek language-part 1: Diglossia in stylistic analysis. *Literary and linguistic computing*, 19(2):197–220, 2004.

[51] S. E. Tranter and D. A. Reynolds. An overview of automatic speaker diarization systems. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1557–1565, Sept 2006.

[52] J. Turian. Using AlchemyAPI for enterprise-grade text analysis. Technical report, Technical report, AlchemyAPI (August 2013), 2013.

[53] P. Verdonk. *Stylistics*. Oxford University Press, 2002.

[54] P. Willett. The porter stemming algorithm: then and now. *Program*, 40(3):219–223, 2006.

[55] H. Zhang and D. Li. Naive bayes text classifier. In *Granular Computing, 2007. GRC 2007. IEEE International Conference on*, pages 708–708, Nov 2007.

[56] Y. Zhang, R. Jin, and Z.-H. Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, 2010.