DEVELOPING FEATURE AND DECISION LEVEL ENSEMBLES FOR CLASSIFYING FAKE REVIEWS

Douglas R. Fraser

A Dissertation submitted to
the School of Computing Sciences of The University of East Anglia
in partial fulfilment of the requirements for the degree of
MASTER OF SCIENCE.
SEPTEMBER 28, 2017

SUPERVISOR(S), MARKERS/CHECKER AND ORGANISER

The undersigned hereby certify that the markers have independently marked the dissertation entitled "**Developing Feature and Decision Level Ensembles For Classifying Fake Reviews**" by **Douglas R. Fraser**, and the external examiner has checked the marking, in accordance with the marking criteria and the requirements for the degree of **Master of Science**.

Supervisor: _____
Dr. Wenjia Wang

Markers: _____
Marker 1: Dr. Wenjia Wang

_____
Marker 2: Dr. Jason Lines

External Examiner: _____
Checker/Moderator

Moderator: _____
Dr. Wenjia Wang

DISSERTATION INFORMATION AND STATEMENT

Dissertation Submission Date: **September 28, 2017**

Student:	**Douglas R. Fraser**
Title:	**Developing Feature and Decision Level Ensembles For Classifying Fake Reviews**
School:	**Computing Sciences**
Course:	**Data Mining and Knowledge Discovery**
Degree:	**M.Sc.**
Duration:	**2016-2017**
Organiser:	**Dr. Wenjia Wang**

STATEMENT:

Unless otherwise noted or referenced in the text, the work described in this dissertation is, to the best of my knowledge and belief, my own work. It has not been submitted, either in whole or in part for any degree at this or any other academic or professional institution.

Permission is herewith granted to The University of East Anglia to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

_____

Signature of Student

Abstract

As e-commerce has grown, so has the problem of fraud and spam. One type of spam (which is also fraud) is fake reviews or opinion spam. They essentially are covert marketing, to persuade readers of the worth (or lack thereof) of a product or service. The number of reviews is now so large that only automation can efficiently address classifying them. This research investigated how effective ensembles for classifying reviews as fake or authentic might be created or improved.

Both feature level data fusion (combinations of feature sets based on different ways of analyzing the text) and decision level data fusion (customized hybrid ensembles) were investigated, as well as ensemble methods. A thorough process first experimented with individual feature sets and several classifiers (individually) to establish performance baselines. Then complexity was methodically increased using these techniques: ensemble methods, combined feature sets with a single classifier, and ensemble methods with those models.

Feature level data fusion resulted in a slight improvement in accuracy, but ensemble methods generally did not. Hybrid ensembles using a majority voting rule were then investigated. Ensemblement of heterogenous classifiers that use different feature sets (combined or not) increased accuracy noticeably, beyond the average of the individual classifiers. However, as classifiers became more complex (in terms of the feature set used), the number of potentially useful classifiers and their diversity became overriding issues and prevented a thorough examination of all possible ensembles.

To overcome this, sampling schemes were developed. These schemes investigated how ordering the pool of classifiers based on formulas using accuracy and pairwise diversity might reveal the better ensembles. But the classifier pool's size was still an issue. In a preliminary investigation into winnowing the pool, the idea of diversity and similarity vectors arose. A diversity vector is a vector of the pairwise diversity measures between one classifier and others. Similarity vectors are created by using a distance function such as cosine similarity to compare the diversity vectors of classifiers. Thus similarity vectors are a way of seeing what classifiers in a set can be grouped together based on their intra-diversity (or lack thereof). These two ideas show promise in better understanding the internal dynamics of ensembles, how classifier accuracy and diversity impact ensemble diversity, and how the ensemble creation process might be optimized.

Acknowledgements

I would like to thank my supervisor, Dr. Wenjia Wang, for his many suggestions and constant support during this research. I would like to also thank Chris Bishop from the UEA Learning Enhancement Team for both his advice on dissertations as well as answering my linguistics related questions, as well as Dr. Geoffrey Guile for his advice and feedback.

Finally, I am very grateful to my sister, Leslie Fraser, for her support and editorial advice.

Douglas Fraser

at Norwich, UK.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

> "Listing is all screwed up" - "The thumbnail is a shirt. The product shown is a shoe. The description is a book. This reviewer is confused."
>
> — Helene (from Amazon.com)

1.1 Online Reviews: Electronic Marketing

As the Internet has grown, humanity's ability to communicate has become more efficient, effective, and complex. E-commerce and marketing are significant parts of this increase in communication; businesses now have the ability to serve a larger, even global, customer base without most of the effort required twenty or thirty years ago. This is especially true for businesses in the consumer related sectors. The growth and reach of online retailers such as Amazon and Alibaba are a testament to the economic and social impact of e-commerce. Customers can search for products and services just by hitting the Enter key. But the fundamentals of commerce have not changed; for example, there are still sellers and buyers, satisfied and unhappy customers, product related issues, and fraud. The Internet is merely the new medium currently facilitating commercial processes.

One of these processes is Word Of Mouth (WOM) marketing. People's social behaviors include telling others of what they've bought, their satisfaction with a service or product, or their advice and recommendations. In marketing, this sharing of knowledge online is known as "electronic WOM". Message boards, chat rooms, and e-mail lists are all social media spaces where this happens, but the most prominent is websites that support online reviews. Online review spaces support a variety of functions for the parties involved: customers, potential customers, and the merchants (King et al., 2014). For customers, the ability to praise or complain about the product, the merchant, or the service fulfills various psychological needs. Potential customers can evaluate the item or service based on others' evaluations. As for merchants, online reviews serve as ways to cheaply get customer feedback and to get free marketing from satisfied customers. Online reviews also increase the perceived credibility

of a merchant and its products. A merchant that prevents customers from reviewing its products is one to be avoided. Without reviews, a potential customer has nothing but price and assumptions to go by; they can not evaluate a physical good to be bought over the Internet nor can they get detailed experiential information on service providers like hotels. So online reviews are an important aspect of the trillion dollar e-commerce economy.

1.2 The Problem Of Fake Reviews

However, as with all human endeavors, there is a down side to online review spaces: fake reviews (also known as "review spam", "opinion spam", or "shill reviews"). The field of marketing has a long history of deception and fraud, notably the "snake-oil" salesmen in the late 1800s in America. Fake reviews are the 21st century version, written for the purpose of influencing potential customers, i.e., positive marketing to increase the perceived reputation of the product or merchant. Alternatively, fake negative reviews are left by competitors to disrupt sales (Segal, 2011). An analogous version of fake reviews in the real world would be person to person stealth marketing campaigns. These campaigns typically involve actors, pretending to not be marketers, who interact with potential customers in social settings. At some point in the interaction, the product being marketed becomes a topic of discussion. The term "covert marketing" is perhaps a better umbrella term for these types of marketing strategies as it references the disingenuousness involved.

The growing importance and pervasiveness of social networking, along with the 24 by 7 nature of the Internet, has only made the problem of online covert marketing worse because now the attention of potential customers must be fought for constantly. And merchants, especially on aggregation sites such as Amazon or TripAdvisor, have a greater number of competitors to deal with. For these reasons, fake reviews have become prevalent. This problem has grown to the point where now even a separate industry dedicated to generating fake reviews exists, with groups of writers coordinating their efforts in writing fake reviews. A related effort is "paid-for" reviews where the reviewer actually does receive the object reviewed (free or for a discount), but it is illogical to assume they will always be objective (Morran, 2016; ReviewMeta, 2016a,b).

The need to filter fake reviews for sites like Amazon is obvious. But even third party sites such as TripAdvisor are affected by fake reviews as their business models are based upon being a trusted source for information about an industry. The economic fallout from fake reviews is not trivial. The cost has been estimated to be in the hundreds of millions

of dollars, which is understandable given the economic benefits of favorable reviews and the reputation economy in general. Fake or biased reviews are a serious enough problem that businesses have sued consumers and Yelp for bad reviews (Severance, 2016) and in turn, Amazon has sued the sources of fake reviews (Northrup, 2016). Government agencies have also acted to protect consumers (Gara, 2013). Given these economic hazards, and the billions in e-commerce revenue, developing effective ways for filtering out review spam is as paramount as it is for deterring e-mail and Web spam.

1.3 Existing Approaches

Beside legal approaches, another way to reduce the impact of fake reviews is to educate consumers on how to spot such reviews. There have been numerous articles published about the problem trying to raise consumer awareness (Popken, 2010; Ballenger, 2011). The heuristics recommended are common sense types ones such as ignoring excessively positive reviews or ones filled with marketing hype. But these heuristics are not sufficient because of several issues. One, the consumer must first be aware of them and how prevalent covert marketing is amongst online reviews (Bambauer-Sachse and Mangold, 2013). The psychological phenomenon known as the Truth Bias impedes this; it makes people want to believe what they read, see, or hear to be true. Otherwise, it is cognitively exhausting and time consuming to constantly be evaluating the veracity of new information. Second, it is much harder to evaluate the credibility of an online review; studies have shown humans to be fairly inept. An important factor is the absence of associated information, e.g., about the writer, so the text is the only source of information and what the reader might infer from it based on their own knowledge, judgements, or projections. Another factor is the quality of the writing in the review; the heuristics mentioned are sufficient for detecting badly written fake reviews which predominated in the early days. But now that opinion spam is an important aspect of the reputation economy and is so lucrative, professionally written spam that avoids being blatantly obvious is more the norm and so these easily bypassed heuristics are not sufficient.

To address the problem, Amazon, TripAdvisor, Yelp, and others have implemented a variety of strategies which can be summarized into three groups: “verifiable actions”, “social relationships”, and “location information” (Ma and Li, 2012). The first involves verification processes to ensure the reviewer actually did purchase the product or service. But Amazon recently discontinued its Verified Purchaser program and has banned incentivized reviews because abuse by review clubs was becoming evident (ReviewMeta, 2016a). The second

method involves adding social networking features and to allow only members to submit reviews. The theory is spammers are less likely to put forth the effort to create and update realistic but fake profiles. Participation in the online community also serves as a signal the reviewer is more likely to be authentic (Kamerer, 2014). But this does not prevent competitors or authentic (but hostile) reviewers from submitting biased reviews as the lawsuits of affected businesses against Yelp attest. Finally, geolocation and geotagging involving mobile phones can verify a reviewer was at a business or lives in the area (based on their profile). But the number of reviews submitted through mobile phones is low and profile information can be faked. So it seems there are no fail safe solutions based on engineering of the review process and leveraging human related factors.

What is left are technological solutions, and given the ever-increasing amount of review data on the Internet, automation is essential. Classification of reviews akin to email spam filtering using machine learning has been shown to be a promising approach as evidenced by the studies reviewed in Chapter 2. The field of linguistics and specifically computational linguistics have served as sources of ideas on how to process the text. However, filtering review spam, based only on the text, is a harder one than for e-mail spam. This is because of the natural language processing (NLP) aspect; review spam is as syntactically and semantically valid as an authentic review. The growing professionalism of spam reviewers is also a factor making it harder to detect review spam; near-duplicate reviews and excessive sentiment are no longer common indicators of spam. For these reasons, behavioral and metadata based features have been shown to be more predictive, in of themselves, than text only based features (Mukherjee et al., 2013b). Non-text features such as metadata (e.g., the IP address) are harder to mask or manipulate. Some examples would be statistical aberrations in the ratings, patterns in how reviewers post their reviews, and even detection of coordinated behaviors amongst a group of reviewers. But text and non-text based features are of course complementary and an industrial solution should use a variety of techniques based on the available raw data.

1.4 Aims and Objectives

To date, most of the research into classifying fake reviews has focused on feature engineering (Crawford et al., 2015). This paper builds upon those studies, investigating how an effective and accurate ensemble might be created using those features. Both feature level data fusion and decision level data fusion were examined along with how ensemble methods improved classifier performance. The overall intent was to construct a more holistic view of the text

using these NLP based features; ideally, different types of features would reinforce each other and improve classification accuracy.

The primary objectives of this study were the following:

1. Develop classifiers that use sets of features evaluated in other studies - ones based on different aspects of the text
2. Investigate how combining feature sets (feature level data fusion) improves performance
3. Evaluate how ensemble methods improve classifier performance
4. Develop hybrid ensembles (decision level combination of the classifiers from previous objectives) and evaluate their performance

Secondary objectives were the following:

1. Evaluate a novel set of features based on the emotional tone of the text and an analysis of the writer's presumed personality
2. Evaluate different schemes for efficiently creating an effective hybrid ensemble

Chapter 2

Related Work

2.1 Overview

This chapter is a summary of the studies used as inspiration for this research. First, the foundation of the field of fake review classification is discussed along with an overview of the whole field. This is followed by a review of the papers used as sources of ideas for feature engineering and the extent of the studies of ensembles for review classification.

2.2 Background Information

Jindal and Liu (2007, 2008) is the foundation of the review spam field. Dividing the types of spam found in online review sites into three categories, they show Types II (brand focused opinions) and III (non-reviews) are very amenable to being classified (an Area Under the Curve (AUC) score of over 98%). However, Type I is a harder problem as the text constitutes an actual review, albeit an inauthentic one; subsequent research has focused primarily upon this type. Using a set of straightforward features created from the review metadata, reviewer information, and product information, Logistic Regression (LR) classifiers achieved an AUC of 78%. This may seem low, but their methodology and results are a good baseline for comparison given that real world data from Amazon was used and the review text was only superficially analyzed. Another important part of the paper is that they describe their findings on near-duplicate and duplicated reviews as well as other aspects of reviews.

Subsequent research has examined the problem from many different angles. The majority of studies can be grouped into the following rough categories: text based feature engineering, non-text based features, and detection of collaborating spammers (spamming groups). Another significant area of research has involved investigating different methods such as positive-unlabeled learning to address the class imbalance problem (only 2% to 6% of reviews in the real world are estimated to be fake (Lau et al., 2010; Ott et al., 2012)). The

most recent research has started to focus on topics such as ensembles and the practical aspects of building a classification system, e.g., using Hadoop. So a thorough review of the entire field is out of scope for this paper (see Heydari et al. (2015); Crawford et al. (2015)). Instead, the focus is on the research that investigates text oriented features and ensembles which are discussed in subsequent sections.

2.3 Text Based Features

Text has a number of different aspects: structural (e.g., sentence length), syntactical (grammar), lexical (word choice), and semantic (meaning). All are possible sources of machine learning features, with the first two falling under the field of stylometry. The following subsections discuss the studies that have used these types of features.

2.3.1 Stylometry

There is no one specific set of features, e.g., parts of speech (POS), that are deemed as canonical for stylometric analysis and an exhaustive list would be out of scope. No consensus has been formed on what could be essential because an important factor is the particular medium, e.g., textual analysis of historical documents would surely rely on different features than a stylometric analysis of Twitter based communication.

For reviews, the tenets of deception theory were an initial source of stylometric type features; an example would be deceptive communication involves the use of longer sentences, but less of a diverse vocabulary or complexity of the sentences or the overall communication. If one is trying to mislead another person, keeping communication simple, easily understood, and superficially persuasive is logical. Burgoon et al. (2003) is an early study that used a variety of simple text based measures as features to evaluate the level of deceptiveness within text. Their results show a decision tree could classify the text of an interaction to determine which individual was the truth teller and which was not.

But it should be noted they studied interpersonal synchronous communication while online reviews are a form of asynchronous one way communication. This difference is the cause of some ostensibly contradictory results such as whether longer texts are more likely to be truthful versus not; the nuances of interpersonal deception theory and the characteristics of the communication are crucial factors in this regard. The unexpected results of Yoo and Gretzel (2009) are one example of this contradiction. They also use deception theory as a theoretical basis for creating stylometric type features. In the results, some hypothesis

were 100% wrong with regard to fake reviews, e.g., fake reviews were more complex and had more self-references contrary to what traditional deception theory posits. Exactly how deception theory needs to be modified to account for variables such as synchronous versus asynchronous communication or the medium of communication (verbal versus text) is still being explored by researchers.

Other studies that use stylometric feature sets are Ramyaa and Rasheed (2004) and Zheng et al. (2006) which focus on authorship attribution. Results were variable, based on the training, data sets, and specific features used, but both achieved accuracies up into the 80% range. Authorship attribution may be a different problem than that of classifying reviews written by unknown authors, but the hypothesis that fake reviewers might have a detectably different style is a logical one, given Burgoon et al. (2003). Consequently, many studies have used features also found in research into authorship attribution. A notable example would be Shojaee et al. (2013) which uses up to 234 basic features divided into two sets (lexical and syntactic). Using the combined sets resulted in a classification accuracy of 84%. Finally, Shrestha et al. (2016) is an interesting study that explicitly combines authorship attribution with fake review detection to detect the reviews written by one author under multiple accounts; the inherent duplicity is a strong signal of inauthenticity.

More complex features like the grammatical structure of the sentences have also been used; Feng et al. (2012) achieved an accuracy of 91% when combining a bag-of-words approach with the grammatical analysis (constituency parsing). However, adding the grammatical analysis based features only improved results 2% above the baseline. Xu and Zhao (2012) use POS, POS bigrams, and dependency parsing of the sentences amongst the combined sets of features. But the cost/benefit ratio, given the complexity of parsing text and the resulting size of the entire feature set, is perhaps not worth the small increase in accuracy given the results of these two studies.

2.3.2 Lexical Analysis

Another source of features used in many studies is the Linguistic Inquiry and Word Count (LIWC) text analysis tool (Pennebaker et al., 2001). It is a way of categorizing words into several of 80 categories along various dimensions: parts of speech, psychological processes, time and causality related words, and conceptual abstractions or base concepts (e.g. sex, death, illness). Ott et al. (2011, 2013) both use the LIWC categories in different combinations with POS features, unigrams, bigrams, or trigrams, along with Naive Bayes and SVM classifiers. The dataset was claimed to be 'gold-standard' as Amazon Mechanical Turk was used as a

source of known fake reviews along with reviews from TripAdvisor and Hotel.com assumed to be authentic ones. The reported accuracies all range from 87% to 89%. Ott et al. (2013) also found that the review sentiment is a factor as there are important differences between fake negative and fake positive reviews that impact the performance of a classifier trying to classify just “fake" versus “authentic".

But as Mukherjee et al. (2013a,b) show, the Ott dataset can not be regarded as gold-standard because of linguistic differences between it and a set of real-world reviews. Using the Kullback-Leibler and Jensen-Shannon divergence measures of the word distribution, Mukherjee et al show the real-world set has a smaller difference in the word distribution for fake and authentic reviews compared to the Ott dataset. This results in the Ott dataset being much easier to classify, especially considering n-grams were used as features; Mukherjee et al.’s results for classifying real world reviews in the same manner as in Ott’s studies achieved an accuracy of only 68%. Li et al. (2014) is a follow-on to Ott et al. (2013) that uses an enhanced version of the Ott dataset, adding more reviews for different domains than just hotels as well as deceptive reviews written by employees and truthful ones written by customers. This new dataset and collection process address some fundamental issues with the first set, but their analysis is complicated by mixing multiple domains and different data collection processes which impact the generation of the review text. But one important result is that they show LIWC and POS based features are robust across review domains; however, using unigram features still results in a slighter higher accuracy.

2.3.3 Readability

A third way of characterizing text that combines structural properties and lexical ones (in a sense) involves readability formulas. Common variables in readability formulas are the length of sentences or the number of syllables in a word. Ong et al. (2014) is one example that uses five of the most common formulas to calculate the readability of product reviews (along with other measures). Their results show there is a statistically significant difference between fake and authentic reviews even though the difference in grade levels that the formulas purport to measure is typically less than 1. In a practical sense, as it applies to humans, it is not significant, but the conclusion is readability can be a useful feature.

Two studies that do use readability formulas as features are Banerjee and Chua (2014a) and Banerjee and Chua (2014b). Based on these papers, it obviously is one study and two different views of the results that examine slightly different things. Interestingly, there seems to be a conflict between them. Banerjee and Chua (2014b) state, based on the results, that

"manipulative reviews were generally less readable than authentic reviews". This matches the results of Ong et al. (2014) along with the slight difference in grade level. However, Banerjee and Chua (2014a) state "genuine reviews were more difficult to be read compared to deceptive reviews". The reason for this apparent conflict lies in how the latter study uses the readability measures to calculate text 'complexity' and 'reading difficulty'. The basis for their derivation of these two measures is suspect and the root cause lies in another study, along with the imprecision of the English language. The other studies surveyed that have used readability as a straightforward measure, and not derived other features from it, show results in line with Banerjee and Chua (2014b); this study did not derive anything from the readability measures.

2.3.4 Semantic Analysis

The term 'semantic' is a very general one; for these purposes, it denotes more abstract properties of a text that are based on the function and nature of the exact words an author uses. The emotions an author is trying to express, i.e., the sentiment, is one such property and so has been used as a possible marker of inauthenticity in consumer level heuristics. One example would be that excessively positive reviews are more likely to be fake. Sentiment analysis is a large field; most studies have focused on investigating and improving techniques for determining the sentiment within a review (typically movie ones). But a few studies have examined the relationship between authenticity and sentiment. Jensen et al. (2013) validate the consumer level heuristics mentioned in the Introduction, showing that the affect intensity (level of sentiment, not just polarity) and even-handeness of a review impact the human assessed credibility of a review. Peng and Zhong (2014) and Chen et al. (2014) both use sentiment in different ways to classify restaurant and store reviews, respectively. Using different methods, they achieve results in the mid-70% and mid-80% accuracy ranges. Feldman (2013) is a good overview of the sentiment analysis field that served as a source of ideas, specifically the need to examine not just the overall review sentiment, but sentence and word level as well.

As for more abstract features, a novel one is the perceived personality of the writer. Initially this may seem of little use, but if fake reviews are a form of covert marketing, then the impression left by the text on the reader, as desired by the writer, is an important factor. Therefore one aim of a fake reviewer is to write a credible review. Credibility is composed of several components: authoritativeness, expertise, and trustworthiness (Fogg and Tseng, 1999). Such qualities would be more easily inferred from text if the writer is seen as open,

conscientious, and not neurotic which are part of the Big 5 personality model (OCEAN, which stands for Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism) (Goldberg, 1990). Mairesse et al. (2007) is an early study that investigates how the OCEAN characteristics can be assessed through different types of analyses of written text. Results were generally modest, but there was a correlation between the personality measures, the text, and the writer's personality as evaluated through other means.

Koven et al. (2014) and Christopher and Rahulnath (2016) are two recent studies that specifically examine classifying reviews using personality measures as features. The former achieved an accuracy of 79% using extraversion, conscientiousness, and openness to experience within their feature set. The latter study, however, had an unorthodox methodology. Reviews were first classified as fake or authentic with a standard procedure utilizing features used in other studies. Then the fake reviews were evaluated along each of the OCEAN characteristics. Openness, extroversion, and low levels of neuroticism were strongly associated with the fake reviews. But there was no comparison to personality characteristics or levels of them for authentic reviews which would have been useful.

2.4 Ensembles

Finally, as the two literature review surveys mentioned indicate, there has not been much research to date into the use of ensembles for classifying fake reviews. Banerjee et al. (2015) is one that combines 9 different classifiers into a voting scheme as well as evaluating each classifier's performance in isolation. This voting ensemble achieved the highest AUC of 81% in classifying a set of real world hotel reviews. But little else is mentioned of this ensemble except that the default parameters in WEKA were used. The features used (for all classifiers) were a combination of stylometric, readability measures, POS, and lexical ones.

Heredia et al. (2016) is a more thorough investigation that looks at ensembling just bag-of-words features using four base classifiers (C4.5 Decision Trees, SVM, Logistic Regression, and Multinomial Naive Bayes) and three ensemble methods (Bagging, AdaBoost, and Random Forest). The ensemble methods were somewhat productive; for the SVM and Logistic Regression classifiers, the AUC improved 2 to 3%. The Decision Tree classifiers improved greatly (over 10%), but ensembling the Multinomial Naive Bayes classifier resulted in little improvement. As it by itself was already the best classifier in terms of performance, Heredia et al reasonably conclude there is little advantage to using ensemble methods given the requisite amount of effort and increased time for training. But the exact set of features

used is a factor, how a classifier responds to them, and the dataset itself is an enhancement of the original Ott dataset whose specific statistical properties are surely something else to consider.

The Ott dataset is also used in Ahsan et al. (2016) along with a set of 2000 unlabeled reviews from Yelp. The interesting aspect of this study is the combination of active learning (to first separate the Yelp reviews into likely spam and not spam based on their cosine similarities) and then merging this with the Ott dataset. The resulting hybrid data is used to train and test several classifiers and a voting scheme determines the final class label. The accuracy of 88% is impressive, but again the features are n-grams which, just like in Ott's original set of studies and so many others, tend to result in accuracy scores around 90%.

Heavily relying upon word choice makes the classification system dependent upon several factors, impacting the development of a review classification system that can handle a wide range of different types of reviews. The reviews used in the training set would be the most obvious factor. Another is the type of good (search or experience) being reviewed; the vocabulary and distribution of it surely would be different based on the good, e.g., hotel reviews focus on different things than reviews of computer products. It is for these reasons this study investigated how more generalized or abstract features such as readability could be used in place of ones based on the specific vocabulary. Determining the utility of these features could alleviate the need to have multiple classifiers or ensembles that are focused on only specific review topics.

Chapter 3

Research Methodology

3.1 Overview

This chapter presents the methodology designed for this study after identifying and justifying more specific questions that need to be addressed to achieve the aim and objectives of this research. The phases of the research process are then described in detail. Finally, a list of the tools and datasets used for this study is provided along with links to relevant papers and URLs.

The following, provided for clarity, is a list of the basic concepts or variables examined in this research and the relevant terminology used.

1. Individual Feature: a specific measure created by processing the review text using a NLP based technique or approach
2. Feature Set: a collection of individual features which are conceptually related
3. Combined Feature Set: an agglomeration of multiple feature sets into one
4. Ensemble Method: the classic methods, such as bagging or AdaBoost, used to create an ensemble from one base classifier and varying the training data
5. Ensemblement: the generic approach of selecting heterogeneous classifiers out of a pool and combining their decisions with a rule

3.2 Research Questions

There has been only a limited amount of research into using ensemble methods for classifying fake reviews based on the literature review. These studies (Ahsan et al., 2016; Heredia et al., 2016) ensemble only a single base classifier and use a single feature set. As for an ensemble of multiple classifiers, only one study (Banerjee et al., 2015) uses decision level data fusion

and a voting scheme and also only a single feature set across all the classifiers. Given these facts, the obvious first questions that arise are “Do ensemble methods add any value when using other feature sets? Would using multiple distinct feature sets, in a decision level data fusion scheme, improve performance?"

But using multiple distinct feature sets is not that same as feature level data fusion. Using multiple distinct feature sets is meant to imply different feature sets are given to different classifiers akin to a Mixture of Experts type design. Feature level data fusion, however, consists of merging two or more feature sets into one which is then provided to a single classifier. Many studies in Chapter 2 investigate feature set level fusion as part of their research into feature engineering. But the number of feature sets used is typically small (up to four) and are dominated by lexical analysis and n-gram type features. Secondary questions therefore were “What feature sets already researched are useful? What has not been combined with other sets?" After refinement, the main questions to be addressed were assembled into the following list. They were ordered so that the answer to one provides insight into the following ones.

1. Using a single base classifier, what is their baseline performance when using each individual feature set?

2. Do ensemble methods, with these individual classifiers and individual feature sets (in isolation), improve performance?

3. Does feature level data fusion (combining multiple feature sets into one) improve performance when using a single classifier?

4. Do ensemble methods, with the models used in the previous question, improve performance?

5. How does performance improve using a custom ensemblement procedure and using the best models from previous stages?

In the process of examining Question 5, a sixth major question arose which was relevant to investigating how to, given time limits and the scale of the problem, efficiently sample the entire set of possible ensemble compositions when using a large pool of individual classifiers:

6. How can individual classifier accuracy and pairwise diversity measures be utilized in finding ensembles with the highest possible accuracy?

The exact methodology used, summarized in Figure 3.1, addresses these questions through a series of phases which build upon each other. These phases can be seen in the expanded steps. Two studies (Alyahyan and Wang (2017) and Xia et al. (2011)) were used as a model or a source of ideas for this research. Alyahyan and Wang (2017) investigate how feature level data fusion can be combined with decision level data fusion to classify multimedia datasets. Investigating how diversity measures and model accuracy can be used to improve the ensembles is an important focus. Similarly, Xia et al. (2011) investigate how two distinct feature sets, with three base classifiers and different combination rules, can be used to improve the classification of sentiment in 5 different datasets of product and movie reviews.

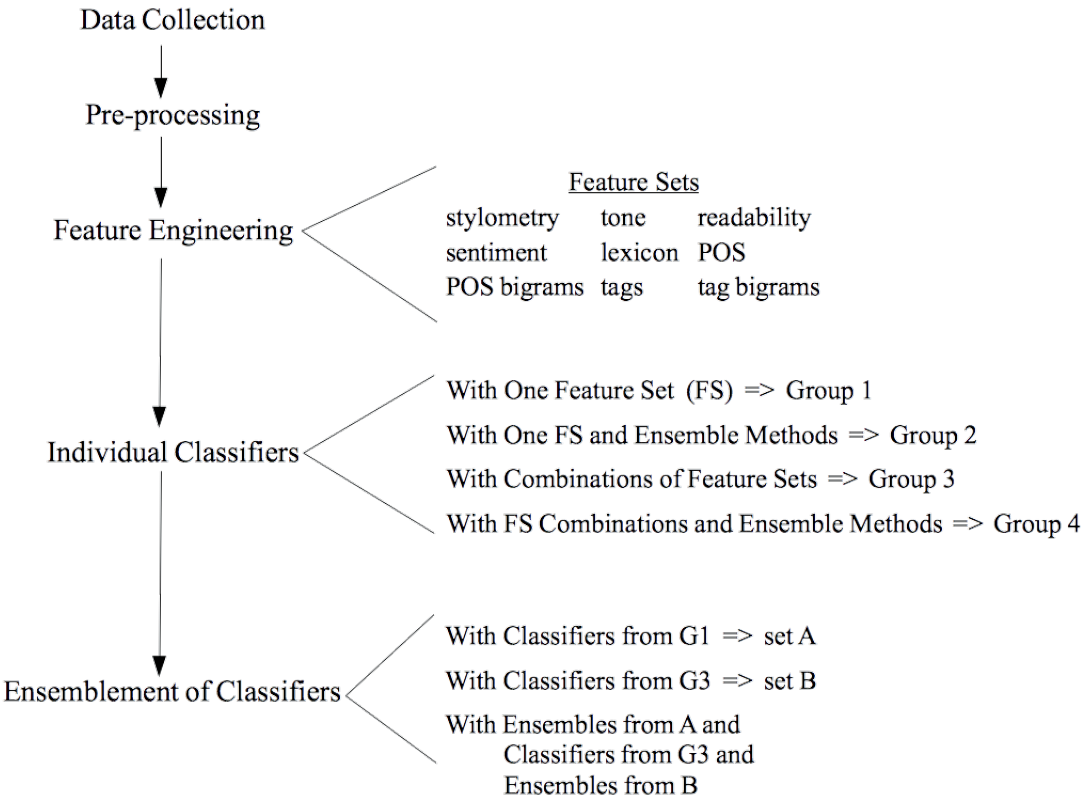

Figure 3.1: Methodology Overview

3.3 Data Collection and Pre-processing

The literature review resulted in a surfeit of possible sources of data as many studies have been done on the problem of classifying reviews. The majority were available on the Internet or on request from the paper's authors. Therefore a decision was made to not create a custom dataset through web scraping or other means. It would have been a time sink and unnecessary, given the intended focus of this project. A dataset used in other studies is a validated one and is better understood in some sense as well as an already cleaned one. Additionally, the other studies that used the set could be a source of ideas in the process of analyzing results.

The dataset chosen comes from Ott et al. (2011, 2013), i.e., the "Ott dataset". The rationale was that it has been used in many studies (being one of the first collated) and it is a labeled set. The 800 reviews labeled as authentic were gathered from a variety of websites such as Yelp and TripAdvisor. The source of the 800 reviews labeled as fake was Amazon Mechanical Turk where the researchers paid random people to write reviews of hotels they presumably had not been to. A notable aspect of this dataset is that both positive and negative authentic and fake reviews have been collated, so there are actually 4 labeled classes even though only 2 (fake and authentic) are the ones focused on for the purposes of filtering review spam. The polarity was treated as a binary feature; in the investigation of feature selection, it was included in some subsets as well as with all other individual features (or not). The text files were processed and the data put into a MySQL database per standard software engineering practices because a database built on text files is not efficient.

3.4 Feature Engineering and Data Analysis

The most common techniques or approaches to analyzing the text to create features are n-grams, bag-of-words, and TF-IDF. A large majority of the studies found in the literature review used n-grams or bag-of-words when analyzing the Ott dataset as the original study used this approach. Its performance serves as a baseline for evaluating the usefulness of other types of features investigated, especially when combined with the different n-grams feature sets. But there has never been an investigation into the utility of combining all these different feature sets together in some fashion while not using n-grams. Relying upon just n-grams for classifying real world reviews, as Mukherjee et al. (2013a) show, is not a viable approach. Therefore, n-grams should be considered as just one possible technique for deriving features instead of the primary one.

Thus this study investigated features derived using a variety of other linguistically oriented techniques. These consisted of readability measures, stylometry, sentiment analysis, lexical analysis, and shallow syntactical analysis. A novel approach, that of tonal analysis and inferences about the writer's personality, was also used. Chapter 4 contains more details on these topics and exactly how individual features were derived. Other possible approaches were considered (constituency parsing to examine sentence structure and linguistic frames). But due to their complexity, they were left for a further study.

Once the reviews were processed and the features derived for each feature set, histograms of them were generated along with density functions. Correlation diagrams of the features

within a feature set were also generated. This was done to get an initial understanding of the data, the distribution of the values, and to see if there were any unusual patterns or features that obviously might be of use or not. For the features with integer values, such as the count of the number of sentences within a review, bar charts were also generated.

3.5 Use of Individual Feature Sets

3.5.1 Individual Classifiers

Once the reviews were analyzed and features created, custom software was written to train models and generate statistics and graphs of the classification results. These scripts were based upon the scikit-learn library; scikit-plot was used to generate basic diagrams such as a graphical version of the confusion matrix and AUC curves. Each script read a collection of related features (a feature set) from the database in isolation; this was to establish a baseline of performance relative to that feature set. The following pseudocode is an overview of the process; the results and more details can be found in Chapter 5. The performance statistics for each run were stored within a MySQL database along with information about the configuration of that run.

```
for featureSubset in subsetCollection:
  for preprocessingConfig in ppCollection:
    preprocess the input data
    for randomSeed in seedCollection:
      split input data into a training set and test set (70 / 30)
      for classifer in classifierCollection:
        train the classifer
        get predictions using the test set
        calculate performance statistics
```

For efficiency, Python's multiprocessing library was used to spread the workload over 4 CPU cores. Each process worked on a subset of the set of different combinations of individual features to investigate feature selection. It was not performed methodically, however, due to the overall time required to do so; subsets were chosen in an exploratory fashion to test different ideas about how individual features might be related. Overall, the results were not noticeably different than using the set of all individual features; therefore, for efficiency and code simplicity, all subsequent experiments used the entire set (which included polarity).

The preprocessing step was necessary as the scikit-learn library expects input data to be standard normally distributed data (Gaussian with zero mean and unit variance). The data

analysis phase revealed all of the continuous features to have something akin to a normal distribution. There were a few features, such as the number of passive sentences, that had a limited range; they were treated as categorical variables and preprocessed using scikit-learn's OneHotEncoder to turn them into a set of binary variables. Preprocessing of the discrete features consisted of binarization or no preprocessing.

The classifiers used were ones available in the scikit-learn library: LR, C4.5 Decision Trees, SVM, Naive Bayes, Multinomial Bayes, and Bernoulli Bayes. Each classifier was always configured with the default settings. Hyperparameter optimization through automation was briefly investigated, but the cost/benefit ratio was too high (e.g. several hours to typically get an increase of 1% to 2% in accuracy). Also, optimizing classifiers at that level was not a focus of this research. Multinomial Bayes and Bernoulli Bayes were only used when the feature set was composed of nominal features and not continuous real number valued ones. As for cross-validation, Monte Carlo cross-validation (with 10 separate runs using a constant set of 10 random seeds) was chosen as preliminary investigation showed some possibly significant differences in performance when the percentage of the data used for testing varied from 10% to 40%. The Ott dataset is small (only 1600 samples) and using 10-fold cross-validation would result in a test set of 160 samples which was viewed as too little.

A final important point is that, for organizational purposes, the feature sets were labeled as 'continuous' ones and 'discrete' ones. This provided a way of comparing feature sets in a more general manner, and was used through the subsequent phases. The label depended on the nature of the individual features within the feature set. The continuous set consisted of readability, stylometry, tone, and sentiment. The discrete set was lexicon, POS, POS bigrams, tags, and tag bigrams.

3.5.2 Individual Classifiers and Ensemble Methods

The process used in this phase was the same as in the first with the addition of a loop before the loop over the list of classifiers. This loop iterated over the ensemble methods (Bagging, Random Subspace, Random Patches, and two versions of AdaBoost) to use on the different classifiers. The scikit-learn default settings for the ensemble methods were used except for the second version of AdaBoost which had a higher learning rate. Investigation of hyperparameter optimization was not performed, as explained. A discussion of results can be found in Section 5.3. Stacking was also briefly examined to see how different classifiers worked together along with varying the classifier hyperparameters. The second level classifier

used was either Naive Bayes or Logistic Regression. But based on preliminary results and the previous phases' results, further investigation was unwarranted.

3.6 Use of Combined Feature Sets

3.6.1 Individual Classifiers

Again, the process was based on the first process. The difference was the feature sets were combined in a systematic manner based on the feature set labels. There were three main substages. First, all combinations of two or more continuous feature sets were processed, for a total of 11 (the single feature sets having been examined in the first phase). The second stage looped over the combinations of the discrete feature sets in the same manner; the total number examined was 25. Finally, all combinations of continuous and discrete feature sets were examined, for a total of 465. So every possible combination of two or more feature sets was examined.

The software developed in the first phase only required a small change based on using the Python's itertools package and its 'combinations' function. Given a list of things and the number of items to be chosen for one set, it returns an enumeration of all the combinations in a sorted order, e.g. item 1 plus item 2, then 1+3, 1+4 and so on (if the total number was 2). Thus the 3 lists of different types of combined feature sets was processed. An outer loop iterated over the number of items to be chosen, e.g. 4, 3, 2 when processing just the combined continuous feature sets (4 being the total number of continuous individual feature sets; 1 was not used as that would be repeating the experiments of the first phase). It was not apparent at the beginning, but having combinations ordered in a logical fashion were of some use in analyzing the results. Results are discussed in Chapter 6.

3.6.2 Individual Classifiers and Ensemble Methods

Again, this phase was very similar to the second one. The same process was followed with an additional loop that iterated over the 5 different ensemble methods being investigated. A key difference here is that not every possible combination of feature sets was ensembled. Only the combinations of just continuous feature sets and just discrete feature sets (the two groups of 11 and 25) were. Based on the results and the results seen in phase 2 (one feature set and an ensemble method), the conclusion was that testing all 464 feature set combinations left would be fruitless as well as take a large amount of time.

3.7 Ensemblement of Classifiers

Investigating decision level data fusion required a different procedure, one based on using a combination rule and multiple classifiers. Again, custom software was written using the previously developed functions as building blocks. A pool of the best models (in terms of accuracy) was created from the results of the previous phases stored in the database as well as the experiment configurations. Three approaches, discussed in subsequent subsections, were taken to thoroughly examine how ensemblement could improve classification accuracy and the process of creating a hybrid ensemble. A key part of the process was efficiently selecting classifiers to use in order to adequately sample the entire set of possible ensembles; it quickly became apparent that a brute force approach of building every possible ensemble was impractical. This issue is what lies behind Question 6; the different schemes developed are briefly discussed below and more fully in Chapter 7.

These ensembles are Mixture of Experts type ones because there was no requirement that all classifiers in an ensemble use the same training set of data, unlike with Stacked Generalization. For example, an ensemble could consist of a Logistic Regression classifier that used the stylometry feature set along with a SVM that used a combined sentiment and tone feature set. Thus each classifier is an expert in a particular view of the text based on the feature set or feature set combination used. This could also be considered a Random Subspaces type ensemble, if the entire feature set is defined as the combination of all specific feature sets and the same base classifier is used repeatedly. A proper Mixture of Expert style ensemble would use a gating network to decide which experts are the best to listen to in a specific situation. But for this study, only a simple majority voting rule was used.

Because of this new procedure, the training and test split process involved using just the index numbers of the reviews and not an agglomeration of all feature values for all feature sets. For each review in the training set, the feature set values relevant for a particular classifier were read from the database before training. Thus each member of an ensemble did train on the same review at each step, but only using the feature values relevant for it. The same process was used for testing. The code ensured a 50/50 split between authentic and fake reviews, but the polarity of the review was not considered as that information was contained in the feature data.

3.7.1 Classifiers That Use One Feature Set

This first approach to developing an ensemble used only the models developed in phase 1. The feature set labels were used as a way to organize the process. First, just the models that used a continuous feature set were combined into an ensemble. A classifier pool ordered by decreasing accuracy was created; the minimum acceptable accuracy was set at 0.60. The purpose of this cutoff was twofold. First, limiting the size of the pool reduced the total number of ensembles to investigate. Second, an ensemble of classifiers with a large difference in accuracy would be counter-productive; logically, the worse classifiers could have a detrimental effect if their cumulative decision overrode the decision of the better classifiers.

Using Python's 'combinations' function, all the possible sets (of N classifiers, N being an odd number and starting at 3) were enumerated over while testing and training. To reduce the amount of unnecessary work, training and testing were actually performed in a separate step outside of the loop over the possible sets. This was possible because the classifier decisions are independent of the ensemble size, so there is no point in repeatedly testing and training the same classifiers over and over. Instead, the classifier's output is saved and then the loop over the possible sets combines the appropriate outputs using a majority voting rule. The loop over 10 different random seeds (which does influence the training and testing) was handled by farming out 10 different seeds to parallel Python processes. Thus the step involving the most work (training and testing) is only performed once per seed.

Given it was impractical to evaluate all ensembles, the first 10,000 to 20,000 ensembles were tested in order to gain a rough idea of performance. The ensemble size used when these experiments were done varied from 3 to 7. But beyond an ensemble size of 5, the first 20,000 ensembles were only a fraction of the total number possible; it quickly became exponential. The total number was also affected by the size of the classifier pool which was dependent upon the minimum level of accuracy set. As the potential pool got larger, e.g., there was far more acceptable classifiers that used discrete feature sets, selecting an acceptable cutoff limit became more difficult. For instance, if the highest classifier accuracy was 0.705, why should a classifier with an accuracy of 0.69 be excluded just because the cutoff was set to 0.70 in order to limit the pool to 25 classifiers (versus 40, if the cutoff was 0.68)? In other words, there were no guidelines on selecting an acceptable cutoff that did not prevent potentially useful classifiers from being included in the pool.

Therefore, a decision was made to investigate ways to sample the entire set of possible ensembles in a logical fashion. Sampling allows the entire set of possible ensembles to be explored as well as allowing for a low cutoff limit (which greatly expands the entire set).

Varying the cutoff limit and investigating how it affected the results of the sampling and the ensembles would provide some heuristics for choosing better initial cutoffs. Another motivation for sampling was that the graphs of the preliminary results of up to 20,000 ensembles revealed that a noticeable pattern was developing. This pattern is in how accuracy changed as the index number of the ensemble did (the index number is its index in the ordered list created by 'combinations'). Therefore, taking advantage of this pattern in some fashion was logically possible. This analysis is discussed at length in Section 7.2.

3.7.2 Classifier Selection Schemes

The term 'scheme' is used to define the process of ranking and subsequently ordering the classifiers before iterating over them to create ensembles. For each scheme, the best N classifiers (for an ensemble of size N) would be used; N being from 3 to the number of classifiers in the pool. These schemes were a way, in effect, to sample the entire set of possible ensembles in different ways. They also enable two things with regard to performance. The first is the evaluation of how important individual classifier accuracy was to the overall ensemble performance. The second is an investigation of how both classifier diversity and the balance between accuracy and diversity affected performance. 17 different schemes were created; the way the code was designed allowed for straightforward development of different ordering schemes. Section 7.3 discusses the details of these schemes, how they performed, and what they revealed about the relationship between ensemble accuracy, classifier accuracy, and classifier diversity.

3.7.3 Classifiers That Use Combined Feature Sets

The investigation into classifiers that used combined feature sets was like the first except the phase 3 models were used. Another change was, due to practical matters and time pressures, the first 10,000 ensembles were not calculated. Doing so was only a way to get an initial understanding of how all of the ensembles performed, leading to the development of the 17 schemes which showed some promise in reducing the amount of work needed to create well performing ensembles. It would still be useful to calculate the first 10,000 to verify that the best schemes are finding the higher quality ensembles, but time limitations precluded this. They also prevented a full examination of the tradeoffs between classifier accuracy and diversity; at this point, it was clear this was the new fundamental problem that prevented further progress. Section 7.4 discusses the results and how they lead to the change in focus.

3.7.4 Best Individual Classifiers and Ensembles

The initial idea for the final phase was to create ensembles out of the best classifiers that use one feature set, multiple sets, or even out of ensembles of individual classifiers (as their accuracy was comparable to the former). But results for the ensembles developed using just combined feature sets showed only a modest improvement in accuracy. This was unexpected and an investigation revealed the difference in accuracy between any two classifiers close together in the ordered pool tended to be fairly small. A small variance in classifier accuracy is of course acceptable, but a lack of diversity amongst the top ones would be detrimental. Another important factor was the number of classifiers in the initial pool.

At this point, it became clear the next step needed was to devise a procedure for winnowing the initial pool of classifiers even before using schemes. The goal of the winnowing process was to reduce the number of classifiers that were too similar to each other, i.e. if the pairwise diversity measure was very low, then testing either classifiers in ensembles would logically lead to very similar results. Chapter 8 discusses the preliminary work investigating a process for winnowing a classifier pool and what it revealed about the dynamic relationship between individual classifier accuracy, diversity measures, ensemble size, and ensemble accuracy.

3.8 Evaluation Metrics

The classification accuracy (averaged over 10 runs of the same 10 seeds) was recorded as well as the sensitivity and specificity (calculated from the confusion matrix). The AUC was recorded as well. Other studies have tended to use precision and recall, but as the dataset used is a balanced one and this research is not in the field of information retrieval, sensitivity and specificity were deemed appropriate. Precision and recall are more appropriate when the dataset is imbalanced (Saito and Rehmsmeier, 2015) and there are important considerations when using precision and recall as measures (Davis and Goadrich, 2006).

Statistical tests were used when needed to confirm the significance of the difference between distributions as well as notched box plots that serve as an informal test of such (McGill et al., 1978). The notches on a box plot show the 95% confidence interval for the median and comparing them serves as an approximate 95% test of the null hypothesis (based on the assumption of independent random samples from a normal distribution). Thus, when notches do not overlap, it is likely the medians differ significantly. But overlap does not rule out a significant difference, so when normality couldn't be assumed, appropriate statistical tests were carried out. A caveat to notched box plots is that comparisons of more than two is

equivalent to multiple simultaneous hypothesis tests. This means the multiple comparisons problem should be kept in mind, given the notches aren't adjusted accordingly; i.e., notched boxplots are an informal test. The notched box plots were created using R. Finally, critical difference diagrams were used to graphically show the rankings of the different ensembles.

3.9 Tools and Datasets Used

The following is a list of the datasets and software packages used in the development process for this research along with the related citations and a brief description.

3.9.1 spaCy

spaCy[1] is a Python library for NLP. In the interests of consistency, it was used as the common library for basic NLP tasks as opposed to the pastiche of toolkits used in previous research. This could be a possible source of inconsistency between previous research and the results in this paper. But because spaCy is a modern and industrial strength library that is kept up to date, the results should be of a higher quality.

3.9.2 Empath

Empath[2] (Fast et al., 2016) is a tool for analyzing the lexical categories of words within a text. It is similar to LIWC which is used in many papers found in the literature review. But LIWC is commercial software and also has a smaller database of lexical categories. Fast et al. (2016) show Empath's categories are highly correlated to the ones of LIWC (r=0.906) so substitution of one for the other was deemed acceptable; any differences in results should be negligible, but possibly still could have had an impact.

3.9.3 Python Natural Language Toolkit (NLTK)

The Python Natural Language Toolkit[3] was used primarily as a common interface to several tools which are contained within it. These tools are types not integrated into spaCy.

[1] https://spacy.io/, https://github.com/explosion/spaCy
[2] https://github.com/Ejhfast/empath-client
[3] http://www.nltk.org/

3.9.4 SentiWordNet

SentiWordNet[4] (Baccianella et al., 2010) was used for sentiment analysis of the unigrams of reviews. It provides not only separate positivity and negativity scores, but also objectivity as opposed to other tools investigated.

3.9.5 VADER

VADER[5] (Hutto and Gilbert, 2014) was also used for sentiment analysis, primarily because it handles negation and n-grams of multiple words.

3.9.6 pattern.en

pattern.en[6] (Smedt and Daelemans, 2012) is a Python module from the Computational Linguistics and Psycholinguistics Research Center of the University of Antwerp. The functionality within pattern.en largely overlaps with spaCy, so it was used only for its mood and modality functions. The sentiment analysis functions were also utilized as it was very easy to integrate and provides an alternative to the sentiment analysis results of SentiWordNet and VADER.

3.9.7 IBM Watson Tone Analyzer

The Tone Analyzer[7] was used to analyze review text along several high-level dimensions: emotional tone, language tone, and social tone (based on the OCEAN personality model). Something to note is that the Tone Analyzer has its own tokenization algorithm; there were some differences in how it parsed the review text and broke it into sentences compared to spaCy. References to the research behind the service can be found at `https://www.ibm.com/watson/developercloud/doc/tone-analyzer/references.html`

3.9.8 Stanford Parser

The Stanford Parser[8] (Petrov and Klein, 2007) was used for the grammatical parsing of sentences as opposed to the Berkeley Parser used in Feng et al. (2012). This is because the codebase for the Berkeley Parser is several years old, has no Python interface, and there were

[4] `http://sentiwordnet.isti.cnr.it/`
[5] `https://github.com/cjhutto/vaderSentiment`
[6] `http://www.clips.ua.ac.be/pages/pattern-en`
[7] `https://www.ibm.com/watson/developercloud/tone-analyzer.html`
[8] `https://nlp.stanford.edu/software/lex-parser.shtml`

issues with the recommended Python to Java library. The Stanford Parser, however, is being maintained and the Python NLTK has an interface to its Java package. This might be a source of differences, but again, an up to date tool is preferable.

3.9.9 Machine Learning Software

scikit-learn[9] (Pedregosa et al., 2011) is a prominent Python library for research into machine learning based systems. It was used as the basis for the code developed for this research along with its extensive documentation.

TPOT[10] (Olson et al., 2016) is a Python library for automating machine learning pipelines which uses genetic programming. It was used for the preliminary investigation of hyperparameter optimization.

brew[11] is a Python library focusing specifically on machine learning ensembles. It was used to investigate the utility of stacked generalization.

scikit-plot[12] was used for easily graphing most of the classification results from scikit-learn.

3.9.10 Source Code

The pypi module "readability"[13] was used to parse the review text and report different readability measures. It was selected out of the numerous modules available due to its wider variety of measures. The code was then enhanced with additional functionality; information on those readability measures was acquired from the documentation for the R package "koRpus"[14]

The code and information on Yule's K and I measures was acquired from `https://gist.github.com/magnusnissel/d9521cb78b9ae0b2c7d6` and `http://cmessner.com/blog/?p=127`

3.9.11 Datasets Used

Myle Ott kindly supplied the datasets used in Ott et al. (2011, 2013) which serve as the foundation for this research.[15]

[9] http://scikit-learn.org/stable/index.html
[10] https://rhiever.github.io/tpot/
[11] https://github.com/viisar/brew
[12] https://github.com/reiinakano/scikit-plot
[13] https://pypi.python.org/pypi/readability
[14] https://cran.r-project.org/web/packages/koRpus/index.html
[15] http://myleott.com/op_spam/

Chapter 4

Feature Engineering

4.1 Overview

Multiple approaches were taken in deriving features from the review text: stylometry, readability formulas, syntactic and sentiment analysis, lexical semantics, and personality and tone analysis. The motivation for using these instead of the more common n-grams was to investigate if capturing a more holistic view of the review was possible, hopefully improving classification performance. Other studies have used these features, except for personality and tone analysis, but not together in a comprehensive way. Each approach is briefly defined, along with the set of associated features, and a rationale provided explaining their applicability to classifying reviews.

These approaches all fall under the rubric of NLP, which first requires specific types of preprocessing to be performed. These steps are tokenization, stop word removal, lemmatization, POS tagging, and chunking. Tokenization is the process of segmenting a sequence of characters into linguistic units such as words or punctuation. Based on this, sentences can then be constructed. Stop word removal consists of removing, as needed depending on the task, very common words such as “and" or “the". Lemmatization is the process of deducing the lemma of a word, which is the base word as found in the dictionary. Part of speech tagging is the process of deducing the role of a word within a sentence, e.g., the verbs. Chunking consists of grouping the tokens, based on their lemmas and POS, into related units such as noun phrases. All of these basic tasks were handled by the spaCy NLP library as part of the initial steps of creating features out of the review text. spaCy was used (unless otherwise noted) to provide a baseline of consistency in contrast to reusing the variety of tools used in other research; there were some differences in results in preliminary testing.

4.2 Stylometry

Stylometry is the analysis of text with regard to its linguistic style, e.g., the vocabulary used or the length of sentences. Most stylometric features are simple measures easily calculated by breaking down the text into different components, e.g., the number of sentences or counting the number of word types. Other measures such as the diversity of the vocabulary are straightforward formulas. The rationale for using stylometric features is based in psycholinguistics. Specifically, the key theory is that unconscious processes affect the language a person uses and can not be fully compensated for, but it possesses quantifiable and distinctive features (Ramyaa and Rasheed, 2004). An example is that liars have been shown to use less first-person pronouns in spoken language, presumably to psychologically distance themselves from their lies. On this basis, investigations into deceit with written text have also used stylometric features. But research is still ongoing as the communication medium and the type of communication (asynchronous or synchronous) are important factors in what statistics are indicative of deceit.

Table 4.1 lists the stylometric features used. They were chosen partly because other studies had used them successfully and because spaCy or the "readability" Python module automatically calculated the measure when processing a review. Other possible features which were excessively complex to calculate and would have required some amount of code development were not used in the interests of time.

4.3 Readability

The readability of a text is another stylistic measure used in numerous studies. Readability refers to how easily a reader might be able to understand a piece of text. However, there is no formal definition of readability; a variety of measures have been proposed, all based upon empirical studies and heuristics (Janan and Wray, 2012). There is a small set of well known readability measures used in numerous studies, but for this research, some more obscure formulas were also used. The rationale was that most of the formulas, especially the common ones, are variations on a theme that involve evaluating a text typically based on the number of words, sentences, characters, and syllables. The result is the American educational level required to comprehend the text. Given the probable high correlation of that subset of formulas, others based on a more abstract scale were also incorporated in the interests of diversity. The Python module "readability" was easily adapted to report these other measures; all features can be found in Table 4.2.

Table 4.1: Stylometric Features

Feature	Description
num_sent	Number of sentences in the review
num_words	Number of words in the review
words_per_sent	Average words per sentence
chars_per_word	Average length of a word
sylls_per_word	Average number of syllables per word
type_per_token	Ratio of word types (e.g. nouns) to tokens
long_words	Number of words longer than 5 characters
complex_words	Number of words more than 2 syllables
not_in_dc	Number not in Dale-Chall list of common words
num_stopwords	Number of stopwords
stopwords_per_words	Average stopwords per words
num_puncts	Number of punctuation marks
puncts_per_chars	Average punctuation marks per characters
passive_sents	Number of passive sentences
yulek_words	Yule's K statistic for words
yulei_words	Yule's I statistic for words
yulek_lemma	Yule's K statistic for word lemmas
yulei_lemma	Yule's I statistic for word lemmas
unique_pos_tri	Unique POS trigrams normalized by the total number
long_words_words	Long words per total number of words
complex_words_per_words	Complex words per total number of words
not_in_DC_per_words	Ratio of not_in_DC and total number of words

One rationale for using readability as a feature, besides it being a way to characterize a text, is that if fake reviews are a form of marketing, then more readable reviews are more easily comprehended by a larger majority of readers. Logically, an overly complex review would put off readers if it seemed to require a college level education to comprehend. This theory is supported by the observations noted in Vasquez (2014) that a large percentage of online reviewers use writing reviews as a creative outlet to express themselves in well-written literary style reviews. The readability measures of those reviews are undoubtedly higher than for prosaic reviews, and so readability can serve as a mechanism for separating these authentic reviews from the rest.

4.4 Syntactic Analysis

Syntax refers to how sentences are constructed and the rules for doing so. Only a shallow analysis of the review text was performed, specifically the parts of speech categories that

Table 4.2: Readability Features

Formula	Description
ARI	Based on characters per words and words per sentences
Coleman	4 separate measures that use number of syllables, sentences, pronouns, and prepositions per 100 words
Coleman-Liau	Based on letters per 100 words and sentences per 100 words
ELF	Based on number of syllables above one (per word) in sentences
Flesch Reading Ease	Based on words per sentences and syllables per words
Farr-Jenkins-Patterson	A variation on the Flesch Reading Ease formula
Fucks	Fucks' Stilcharakteristik uses number of words, characters and sentences
Gunning Fog	Uses average sentence length and percentage of complex words
Strain	Based on number of syllables within sentences
Tuldava	Based on average word length and average sentence length
Dale-Chall1	2 measures based on average sentence length
Dale-Chall2	and number of words not on Dale-Chall list of easy ones

spaCy automatically assigns when parsing text. spaCy uses the Google Universal POS tag set[1]; the number of instances of each POS were counted, each POS being a feature. POS bigrams were also collated, e.g., determinate+noun, as it was simple to code and might reveal something when used as features. spaCy also supports a more sophisticated POS tagging system based on the Penn Treebank set[2]. This is a more refined version of the Google set containing more categories. This research refers to these features not as the POS feature set, but the tag feature set. Bigrams of the tags were also calculated. The motivation for using POS as features is again psycholinguistic research has shown there is something of a bias with regard to certain categories used by liars, e.g., more nouns than usual or less pronouns. A table of these features is not included as they would be large; the URLs in the footnotes provide tables and an explanation of the POS.

More complex analyses of sentence structure are also possible, with dependency and constituency parsing. These analyses establish how words are related to each other, e.g., what adjective modifies what noun or what set of words is a noun phrase. Using these as a source of features was briefly investigated, but separate software would have been needed which was not immediately usable with Python. As a result, due to time limits and the already large number of feature sets, these two sources of features were left for later research.

[1] https://github.com/slavpetrov/universal-pos-tags

[2] https://ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

4.5 Sentiment Analysis

Sentiment analysis, also known as opinion mining, is an entire subfield of data mining. Its overall goal is to identify and quantify the emotional aspects of text or other forms of communication in order to determine any number of things. One is the affect (the emotional states) of the speaker or writer; another is the intended emotional impact on the receiver of the communication. The applicability of sentiment analysis for classifying reviews is obvious; a review that is overly positive or negative, excessively so to a human, is more likely to be a fake one trying to manipulate the reader. This idea is reflected in the consumer level heuristics mentioned in the Introduction. The utility of this heuristic has decreased as spammers have gotten more sophisticated in their writing, but sentiment analysis still can be useful as a feature.

Exactly what tool to use was a difficult choice. There are quite a number of sentiment analysis tools available and Ribeiro et al. (2016) benchmarked 24 versions. They show that results for each varied considerably across a number of test datasets. The reason is a number of factors: the training set the tool is built upon, the data being evaluated and how it compares to the training set, how well sarcasm and negation is handled, and the fact that positive messages or emotions are easier to evaluate than negative or neutral ones. Based on the results, the tools VADER, SentiWordNet, and pattern.en were chosen to try and balance various requirements. The first was to get a number of 'opinions' in case one method was incorrect about the sentiment in a review. The second was to use tools that were based on larger lexicons (older methods typically having a smaller one). Third, tools that had Python interfaces available were essential, and finally, methods that were trained on a set of product reviews (versus other sets like Twitter data). Another consideration was using tools that provided both positive/negative or positive/neutral/negative scores so that neutrality as a feature could be investigated. If the ranking of the tools on just product reviews was the only consideration, then the SentiStrength tool would have been used. But it requires Java and so for practical reasons was not used. SenticNet is another alternative, but the work to integrate it was deemed too much given time limits.

Mood and modality are two other affect related measures used as features, computed using the pattern.en Python module. Mood, a term in grammar, refers to the intent of the verb or the thought behind it. Pattern.en reports five moods: imperative, indicative, subjunctive, and conditional. Definitions and examples of these moods can be found in Table 4.3. Mood was investigated on the hypothesis that fake reviews, if they are more marketing influenced, would show a stronger tendency towards having sentences with an imperative mood. This reflects

the "call-to-action" philosophy of marketers, who consciously craft sentences to try to impel the reader to do something (i.e., buy the advertised product) after emotionally influencing them with the rest of the ad copy. Modality is a measure of the certainty expressed with a sentence. Pattern.en reports a score of -1.0 to 1.0, where scores above 0.5 are indicative of more factual statements, i.e., the statement is less easily interpreted as an opinion. It was not immediately obvious of what use modality might be as a feature, but it was available.

Table 4.3: Mood Definitions and Examples

Mood	Definition	Example
Imperative	A direct command or request	Do not get a room at this hotel!
Indicative	Factual statements or positive beliefs	This hotel was very expensive
Subjunctive	Expression of opinions or emotions	A fabulous hotel!
Conditional	If-then type statements	This hotel would be ok if it was cheaper

Table 4.4 summarizes the sentiment related features. Each review was evaluated in its entirety as well as each sentence individually and unigrams and bigrams (SentiWordNet can only deal with unigrams). The rationale for gathering all this data was that preliminary investigation revealed inconsistent behavior, e.g., one very positive sentence in a review (as it was interpreted) may counter the other slightly negative sentences, thus leading to an overall positive score which made no sense from a human interpretation. This inconsistency and need for improvement in all the tools was discussed in Ribeiro et al. (2016). Based on analysis of the results, the SentiWordNet features were dropped from consideration. Feature selection, i.e., subsets of this entire set of features, was investigated as part of the first evaluation of classification.

4.6 Lexical Semantics

Another way to analyze text is to categorize the more abstract concepts within the text based on the words used. An example would be words related to the concept of death, e.g., dying, illness, hospital, or funeral. A rationale for this approach is that fake reviews might predominantly focus on a specific set of abstract topics, such as dirty rooms or incompetent staff, compared to a wider variety in authentic reviews. As mentioned, one tool commonly used in many studies for lexical analysis (not just for classifying reviews) is the LIWC. But the LIWC was not used as it is a commercial product.

Table 4.4: Sentiment Features

Feature	Tool	Description
compound	VADER	The compound score reported for the entire review
positive	VADER	The positivity score reported for the entire review
negative	VADER	The negativity score reported for the entire review
neutral	VADER	The neutral score reported for the entire review
polarity	pattern.en	The polarity score reported for the entire review
objectivity	pattern.en	The objectivity score reported for the entire review
sumcmp	VADER	The total compound score over all sentences in a review
sumpos	VADER	The total positivity score over all sentences in a review
sumneg	VADER	The total negativity score over all sentences in a review
sumneu	VADER	The total neutrality score over all sentences in a review
sumpol	pattern.en	The total polarity score over all sentences in a review
sumobj	pattern.en	The total objectivity score over all sentences in a review
summod	pattern.en	The total modality score over all sentences in a review

Instead, a new tool called Empath was used, which has 194 categories and has been favorably evaluated against the LIWC. spaCy was first used to tokenize the review and then the lemmas of the words within the review (not including the stop words) were used as input to the Empath software. The output consisted of each inbuilt Empath category and the count and normalized count (over all words). Empath also allows for new categories to be created, which would be something to investigate in a follow up study to see if review specific topics might be useful features. Reviews are not about any generic thing, but a specific type of thing, so a classifier trained on hotel reviews would logically not work well on Amazon product reviews. A complete list of Empath's categories can be found in the source code at `https://github.com/Ejhfast/empath-client`.

4.7 Personality and Tone Analysis

One aspect of reviews that has not been explicitly considered is the perceived review credibility. Hypothetically, a fake reviewer would try to increase it by emphasizing one or more of the components of credibility: authoritativeness, expertise, and trustworthiness (Fogg and Tseng, 1999). These qualities can not be definitively measured as they are partly a function of the reader's interpretation of the review and reviewer, but they could be inferred through tangential measures. A possible one is the perceived personality of the writer; a reviewer who seems open, conscientious, and not overly emotional will be perceived as more credible.

If fake or authentic reviews are seen to have a large bias in terms of the reviewer's personality, then a classifier could utilize that knowledge.

Table 4.5 lists the different tones analyzed by IBM's Tone Analyzer service and the related measures. The social tone, or personality measures, were the initial focus, but as emotional tone and language tone were available, they were also used as features. They could be thought of as variants of sentiment analysis. The emotional tone features in particular were thought to be potentially useful, as in fake reviews might be excessively emotional in the same way the sentiment analysis might compute an excessive score. The values for these measures is a real number from 0 to 1. A score less than 0.5 indicates the tone is unlikely to be perceived within the content, while a score above 0.75 indicates a high likelihood. As the Tone analyzer has its own text parsing system, spaCy was not used in this process.

Table 4.5: Social, Emotional, and Language Tone Features

Group	Social	Language	Emotional
Tones	Agreeableness		Anger
	Conscientiousness	Confident	Disgust
	Emotional Range	Tentative	Fear
	Extraversion	Analytical	Joy
	Openness		Sadness

4.8 Summary

Table 4.6 is a summary of the different feature sets used in this research. These feature sets are only a subset of all possible ones; linguistic frames, constituency parsing of the grammar, and topic models are potential feature sets that were not investigated due to time constraints. The usual n-grams approach would also be useful to see if combining it with the entire extent of possible feature sets could improve upon published results.

Table 4.6: Feature Sets Used

Feature Set	Description	Number Of Individual Features
Readability	Formulas for calculating text readability	15
Stylometry	Measures of the style of the review text	22
Sentiment	A sentiment analysis of the review text	13
Tone	Different types of psychological tones within a text	13
Lexicon	Basic concepts underlying words	194
POS	Parts of speech categories	17
POS Bigrams	Combinations of two POS	225
Tags	A more detailed version of POS	56
Tag Bigrams	Combinations of two tags	1401

Chapter 5

Results with Individual Feature Sets

5.1 Overview

This chapter first presents the results from using just one individual feature set, in isolation, with the six different classifiers. How they performed and possible explanations as to why are discussed. The second part then examines how using ensemble methods on the models affected performance as the former serves as a baseline for the latter.

5.2 Individual Classifiers

Table 5.1 summarizes the mean accuracy, sensitivity, specificity, and AUC for each classifier, across each individual feature set. This set of data serves as a baseline for performance as the simplest configuration is used: one feature set and no ensemble methods. The classifiers used with each feature set are based upon whether the features are continuous ones or discrete ones, e.g., Gaussian Naive Bayes classifiers do not work with discrete features and TF-IDF was not used as a technique to create real valued features. The one exception is the lexicon feature set where the Empath software also provided real numbers which were the counts normalized over all words in the text; these were used as continuous features. Figure 5.1 is a bar chart of the mean accuracy (with an error bar for the standard deviation) for the LR classifier, which overall was more consistent in its performance than the others. Each feature set is discussed in detail in the following subsections.

Table 5.1: Results Per Feature Set and Classifier

Feature Set	Classifier	Acc.	Sens.	Spec.	AUC
Stylometry	Logistic Regression	0.676	0.714	0.639	0.740
	SVM	0.673	0.738	0.610	0.741
	Naive Bayes	0.543	0.979	0.113	0.681
	Decision Tree	0.591	0.585	0.598	0.592
Readability	Logistic Regression	0.667	0.687	0.645	0.717
	SVM	0.654	0.706	0.603	0.705
	Naive Bayes	0.562	0.647	0.480	0.592
	Decision Tree	0.548	0.562	0.533	0.548
Sentiment	Logistic Regression	0.610	0.654	0.568	0.664
	SVM	0.615	0.709	0.525	0.673
	Naive Bayes	0.553	0.799	0.311	0.586
	Decision Tree	0.547	0.548	0.547	0.547
Tone	Logistic Regression	0.632	0.667	0.599	0.674
	SVM	0.634	0.763	0.508	0.682
	Naive Bayes	0.622	0.589	0.657	0.668
	Decision Tree	0.562	0.553	0.571	0.562
Lexicon	Logistic Regression	0.721	0.725	0.719	0.789
	SVM	0.734	0.750	0.717	0.807
	Naive Bayes	0.628	0.767	0.491	0.691
	Multi Bayes	0.698	0.714	0.682	0.766
	Bernoulli	0.685	0.694	0.679	0.753
	Decision Tree	0.597	0.612	0.582	0.598
POS	Logistic Regression	0.701	0.764	0.641	0.765
	SVM	0.575	0.420	0.735	0.643
	Multi Bayes	0.692	0.736	0.651	0.760
	Bernoulli	0.636	0.802	0.473	0.672
	Decision Tree	0.595	0.604	0.587	0.595
POS Bigram	Logistic Regression	0.749	0.761	0.737	0.827
	SVM	0.715	0.730	0.702	0.792
	Multi Bayes	0.734	0.781	0.686	0.810
	Bernoulli	0.684	0.767	0.602	0.764
	Decision Tree	0.623	0.642	0.606	0.624
Tag	Logistic Regression	0.702	0.753	0.656	0.774
	SVM	0.610	0.491	0.731	0.665
	Multi Bayes	0.693	0.703	0.684	0.754
	Bernoulli	0.662	0.751	0.576	0.723
	Decision Tree	0.603	0.619	0.587	0.603
Tag Bigram	Logistic Regression	0.720	0.719	0.720	0.797
	SVM	0.725	0.697	0.752	0.799
	Multi Bayes	0.719	0.728	0.710	0.782
	Bernoulli	0.709	0.761	0.659	0.786
	Decision Tree	0.606	0.615	0.598	0.607

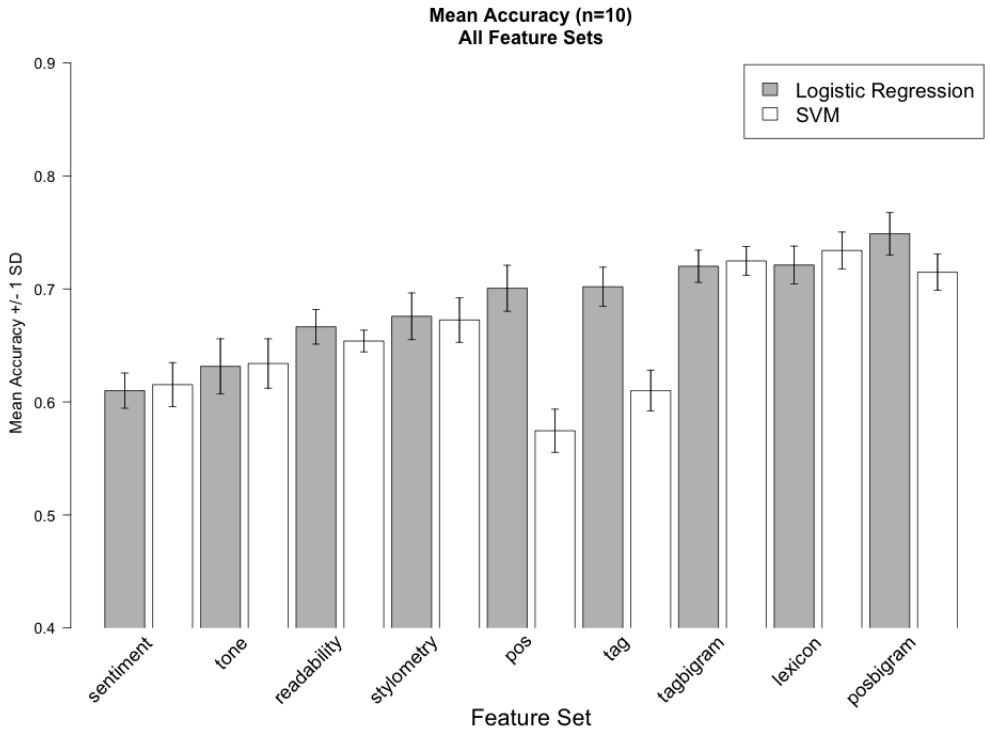

Figure 5.1: Accuracy Over Individual Feature Sets

5.2.1 Stylometry and Readability

Examination of the histograms and density plots of the stylometric and readability features provides an explanation as to why the Naive Bayes classifier performed so poorly. As the example in Figure 5.7 shows, the difference between fake reviews and authentic ones tended to be small for all the individual features; the distributions greatly overlapped and typically had similar peaks. Naive Bayes classifiers utilize the mean and variance of the distribution of values for a class, and if these are similar for multiple classes, the classifier will have a harder time distinguishing between them. Therefore the only slightly better than chance results are to be expected.

This is reflected in the large disparity between the sensitivity and specificity of the Naive Bayes for several of the feature sets; the hypothesis is that classifiers were detecting particular individual features as being very associated with one review class, but also consequently making more errors on the other class. The confusion matrices were noticeably imbalanced in terms of the ratio between True Negatives and True Positives. This implies some dependence amongst the stylometry and readability features, which is logical given how some are related. Figure 5.2 is a good example that reflects this; it is the Naive Bayes's ROC curves for one particular seed when using the stylometric feature set. The curves are plotted as if each class was the positive one; for our purposes, this is class 1 (the fakes). Thus the blue line

is almost equivalent to the random chance baseline. This is reflected in the high sensitivity and low specificity as seen in Table 5.1; the Naive Bayes classifier was very accurate at detecting fakes (few false negatives), but there were almost as many false positives (incorrectly classified authentic reviews). Several of the graphs for the seeds had lines like this while the others showed more equivalence between the two lines. This variability is likely due to the randomness of what reviews were used in training and test and what individual features the Naive Bayes classifier decided to use.

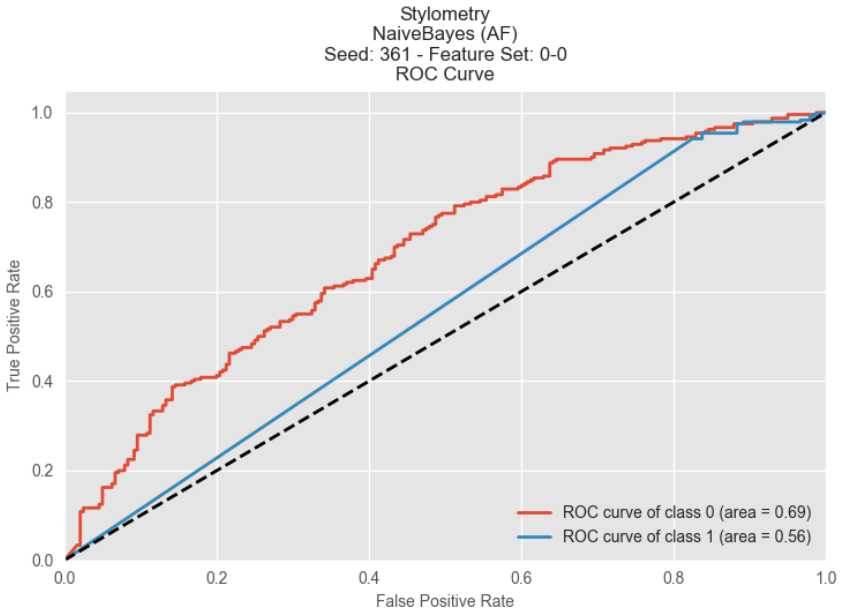

Figure 5.2: Naive Bayes ROC Curves (Stylometry Feature Set)

As part of investigating this issue, histograms and density plots based on dividing the review classes into 4 (authentic / fake, negative / positive) were created. Figure 5.3 is one example (the distribution of the uniq_pos_trigram feature). In it, the plot for fake positive reviews is contained entirely within the plot for authentic positive reviews and is separate in that sense from the fake negative reviews. Based on this, it seems there is a possible XOR aspect to this problem related to the review polarity. To quickly gauge the likely significance of the difference between medians, a notched box plot was created. Figure 5.4 shows that there is certainly a statistical difference between negative and positive reviews in terms of the median for the uniq_pos_trigram feature. But for both polarities, there is not a significant difference between authentic and fake reviews as the notches overlap. Based on these two figures, the review polarity looks to be an important factor that hampers classifying reviews correctly; this was not initially anticipated. If a classifier, due to the nature of the training set, learns how to classify fake negatives very well, it may not perform as well on fake positives.

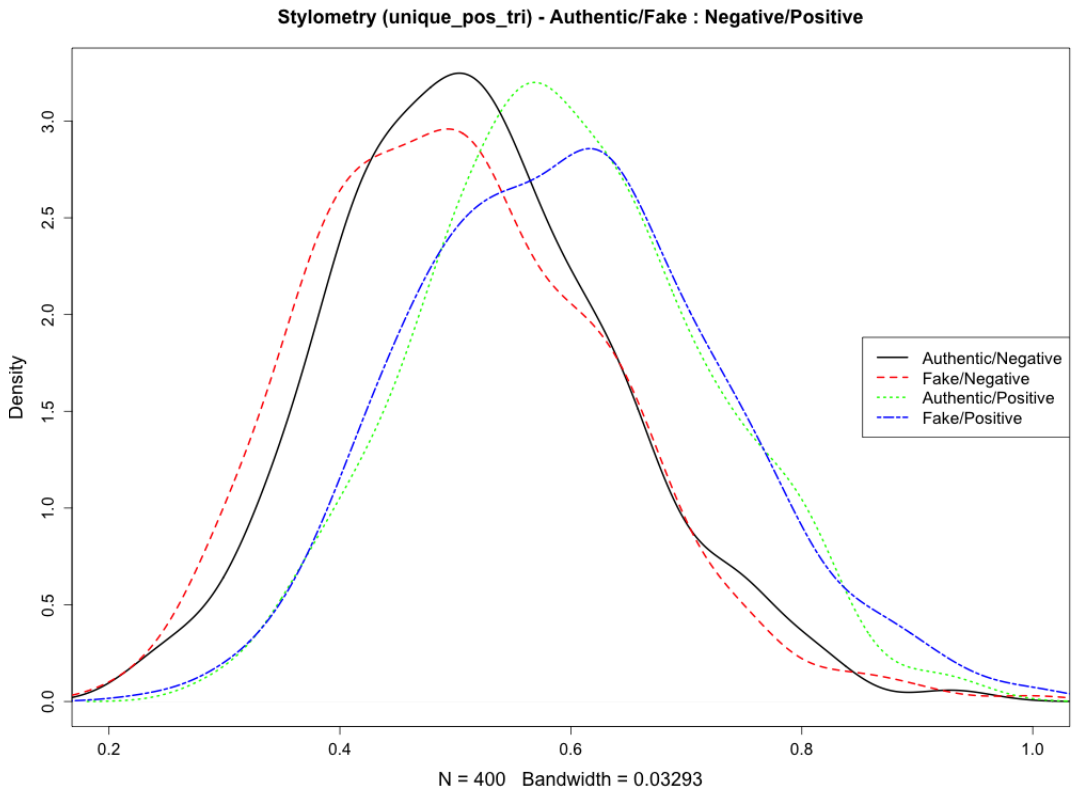

Figure 5.3: Density Plots For uniq_pos_trigrams

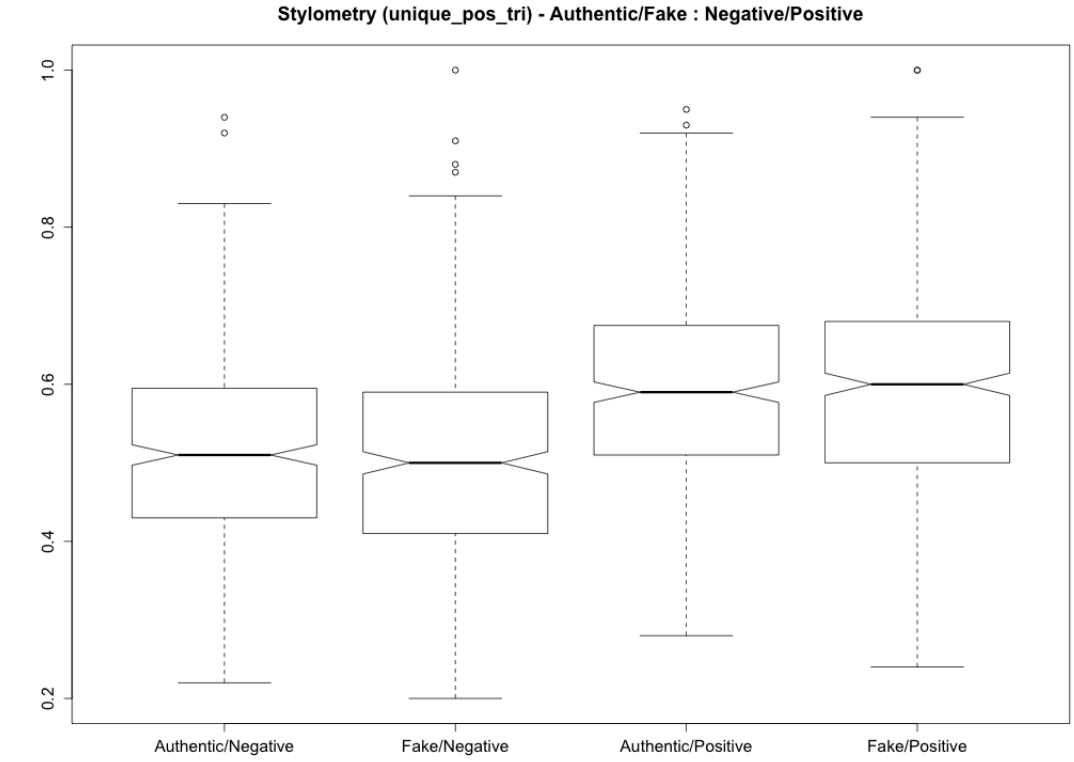

Figure 5.4: Box Plots For uniq_pos_trigrams

5.2.2 Sentiment

Table 5.2 reveals why all the classifiers had a less than desirable accuracy when using the sentiment feature set. This matrix was calculated by cross referencing the review type against the positivity feature level for the entire review. 0.5 and -0.5 are the cutoff points for definitive significance, e.g., a score of 0.5 reflects a definitely positive sentiment. As this table and related ones for the other sentiment features (not included) showed, negative reviews of either type were mis-scored more often than the ones with positive sentiment. If an individual feature is completely incorrect in what it is supposed to be measuring, that will certainly affect classification accuracy.

The reasons for this misclassification of the review polarity are many, but there are two major ones. First, the current state of the art in sentiment detection does not handle sarcasm very well and sarcastic reviews for bad hotel experiences was common. Second, sentiment analysis does not process conjunctions in a sophisticated fashion, e.g., sentences that include the word 'but' can be classified as overall positive even though a human interpretation would be negative. A more sophisticated analysis of the sentiment involving feature creation from unigrams or bigrams, for example, might result in some benefit. But that would have required more time and is a research project in its own right. So only the basic measures the sentiment analysis tools provided were used.

Table 5.2: Number of Reviews Per Class and Positivity Scores

	Positivity Score				
Review Class	< -0.5	< 0	0	> 0	> 0.5
Authentic / Negative	2	123	2	270	3
Authentic / Positive	0	0	0	353	47
Fake / Positive	0	0	0	330	70
Fake / Negative	3	176	2	219	0

As for mood and modality, there were some surprising results. The anticipation was that fake reviews would have more imperative sentences based on the hypothesis the reviewer would be more assertive and try to motivate the reader in some manner. Instead, there were no significant differences between fake and authentic reviews in this regard. Assertiveness is actually reflected in the modality score for authentic reviews; they contained more sentences that were interpreted as statements of facts or statements of higher certainty. Figure 5.5 shows the notched box plots of the modality score for the two types of reviews. The non-overlapping notches imply that the sum of the modality for all sentences in authentic reviews is statistically

different than that for fake reviews. So as an exercise, first the histogram and density plots for summodal were checked; they implied the distributions were skewed and R's skewness() function confirmed this. Therefore the Wilcoxon test was used to confirm the hypothesis that the distributions were not equal; a t-test was not used as a log transformation looked necessary and was more complicated. The calculated p-value was 5.005e-14, much less than 0.05, so this was confirmation the two distributions were significantly different.

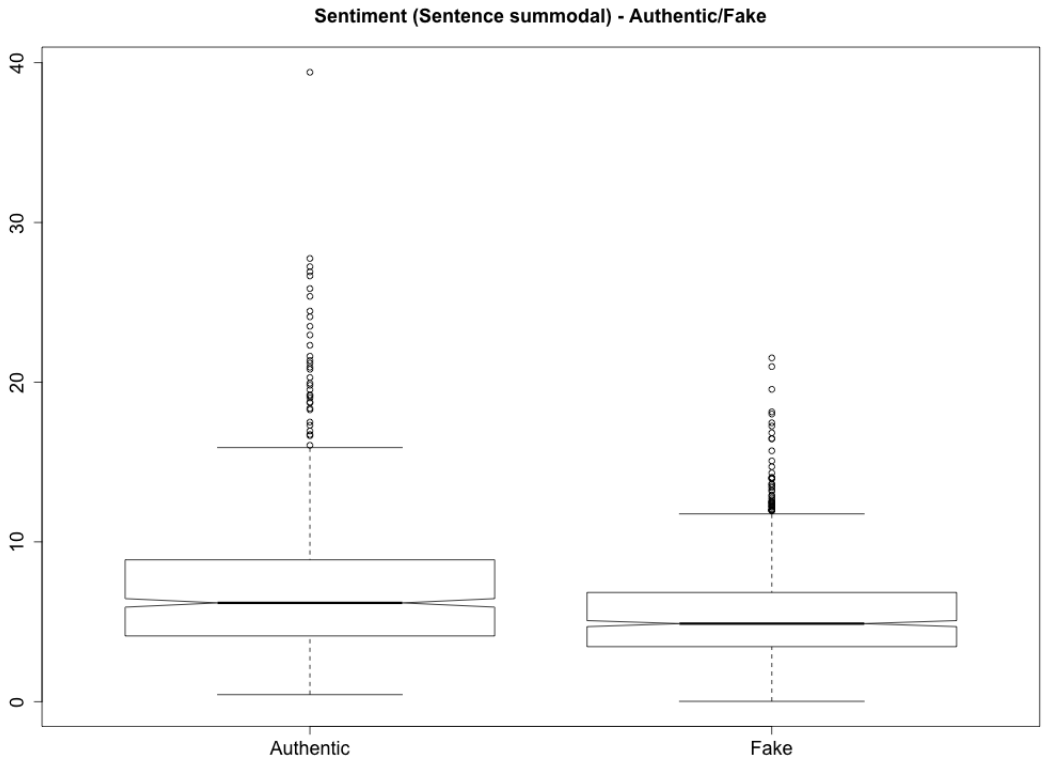

Figure 5.5: Comparison Of sum(sentence modality) For Reviews

5.2.3 Tone Analysis

How well the tone features would work was uncertain, but as the results show, this feature set does have some predictive power. Examining the histograms and density plots reveals that the distributions for fake and authentic reviews were always very similar and the only noticeable difference was sometimes higher peaks for one review type or the other. For example, authentic reviews tended towards more expressions of joy than fake ones, in that more authentic reviews had a score around 0.6 which was the place the highest peak exists. Fake reviews peaked at 0.65 and at a lower level. As for why the accuracy of the classifiers was less than 70%, an important factor is surely not enough text was provided to IBM's Tone Analyzer. Reviews tended to be short; their documentation states that accuracy is dependent on the amount of text and 600 to 800 words or more is the ideal.

5.2.4 Lexicon

Examining other studies that used the LIWC reveals comparable results to those generated using the Empath software. There is some slight variation (e.g., Ott et al. (2011) report an accuracy of 76.8% using a SVM and for this research, it was 73.4%) but there are three factors that can explain this. The most obvious is methodological difference due to the hyperparameter configuration (Ott et al. (2011) uses a linear kernel and nothing else was stated, while scikit-learn by default uses a radial basis function kernel and the other default settings were not changed). The second is how training and testing was performed. In Ott et al. (2011), a 5-fold nested cross validation was performed (as opposed to the 10 fold Monte Carlo procedure used) and in Ott et al. (2011), "Folds are selected so that each contains all reviews from four hotels; thus, learned models are always evaluated on reviews from unseen hotels.". This method of selecting reviews is different than the random procedure used in this study; it is not clear why the hotel was considered a significant variable, but it ensured balanced classes. Finally, the dataset of Ott et al. (2011) contains only positive reviews (of both classes); this study used both positive and negative reviews. As mentioned, polarity looks to have a more significant influence than initially thought and so a slightly lower accuracy for this study is not surprising. In the process of performing this research, it was noted that, when a certain random seed (or two) was used, the results always tended towards lower accuracy. This bias needs to be verified as it is only a suspicion, but if this is indeed the case, it indicates there might be statistical anomalies or correlations within the data such that the selection process for the training set can influence the accuracy more than expected.

5.2.5 Parts Of Speech

The noticeably poor SVM performance when using the POS feature set was unexpected, so an exploratory analysis was done. Comparing the number of fake reviews with a particular POS to the number of authentic reviews reveals only 196 reviews at most (out of 1600) could be distinguished by the presence or absence of a POS. Table 5.3 shows these 7 measures and the difference between the numbers of each class. Obviously, individual POS features do not have enough discriminatory power; SVMs end up with data points (for both types of reviews) with the same value in most dimensions. Exactly how this would affect computations of the hyperplane is an exercise left to the reader, but it would certainly have a negative impact given how few POS features there are. Increasing the number of dimensions (from 17 to

225) by creating POS bigram features addresses this issue; individual features now have more discriminatory power, more variance, and better reflect the differences between fake and authentic reviews. The improvement in all the classifier accuracies can be explained by this expansion of the dimensions; there are too few features when using the basic POS feature set and so decision boundaries are crude, increasing the generalization error.

Table 5.3: POS and Difference Between Authentic and Fake Reviews

POS	SYM	NUM	X	INTJ	PART	PROPN	PRON
# of Reviews	196	151	58	57	15	14	9

5.2.6 Tags

The tag feature set is merely a refinement of the POS feature set, i.e., several tags are grouped under one simpler POS. It is as if the number associated with an individual POS feature is comprised of several whole numbers (the tag associated values) in certain ratios. The expansion of the number of dimensions adds more space, in effect, between data points and the two classes as a whole. Consequently, the SVM classifiers can find a better hyperplane to distinguish between the two classes. The other classifiers work differently, so the lack of a significant increase in accuracy is expected. The Bernoulli Bayes classifier does improve somewhat; this can be explained by again more features equaling more independent binary variables. The binary variables associated with the POS features have been split into smaller components which are more precise or accurate in terms of correctly predicting the class (and are now independent); POS based variables are cruder and the components interfere with each other. As for the difference in results between using the tag feature set and the tag bigram feature set, the explanation is the same as for the difference when using POS and POS bigram feature sets. Bar charts of all this data were not included for reasons of space.

5.3 Individual Classifiers and Ensemble Methods

In general, using ensemble methods on an individual model (when using one feature set) resulted in little to no improvement. For this reason, a table of the results is omitted; Figure 5.6 is an example that compares the results for base, Bagged, and AdaBoosted classifiers when using the lexicon feature set. The other graphs for the different feature sets are very similar in the general lack of improvement; these graphs can be found on the supplementary DVD. The classifiers listed in each figure depend upon the nature of the feature set. For example,

Multinomial Bayes classifiers can not be used when there are continuous individual features present, so it and the Bernoulli Bayes classifier only show up in certain graphs. Table 5.1 lists what classifiers were used with what feature sets.

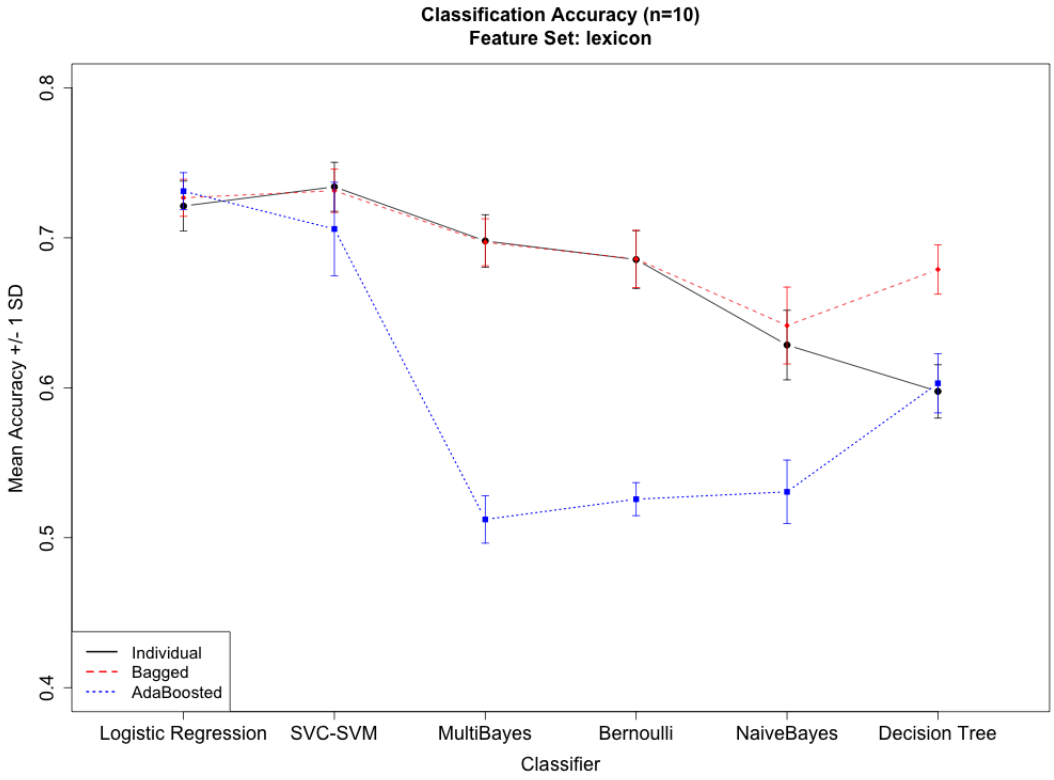

Figure 5.6: Accuracy Using Lexicon Feature Set

The sole exception when an ensemble method added significant value was when the base classifier was a Decision Tree and used with Bagging. This can be explained by first examining the histograms and density plots of the individual features within the feature sets; Figure 5.7 is one example that shows the histogram and density plot for the Yule's K measure for words. Both the authentic and fake reviews are graphed. The other features' graphs have been omitted as the majority of individual features have histograms and density plots conceptually similar. As seen, there is only a small difference between the two distributions and the histograms show the sets of values for Yule's K (for authentic and fake) greatly overlap. There is no specific range of values associated only with authentic reviews versus fake ones; only a few features in specific feature sets have ranges that are associated with just one class of review. They are at the extreme ends of the overall range of values which means they are for just a small number of reviews and so are not helpful overall.

Based on reviewing all the individual feature histograms, the conclusion was there were no very discriminative features for a Decision Tree to consistently utilize. The majority displayed a distribution quite reminiscent of a normal distribution, in fact, for both types of reviews. This results in the Decision Trees having high variance, based on the exact set of

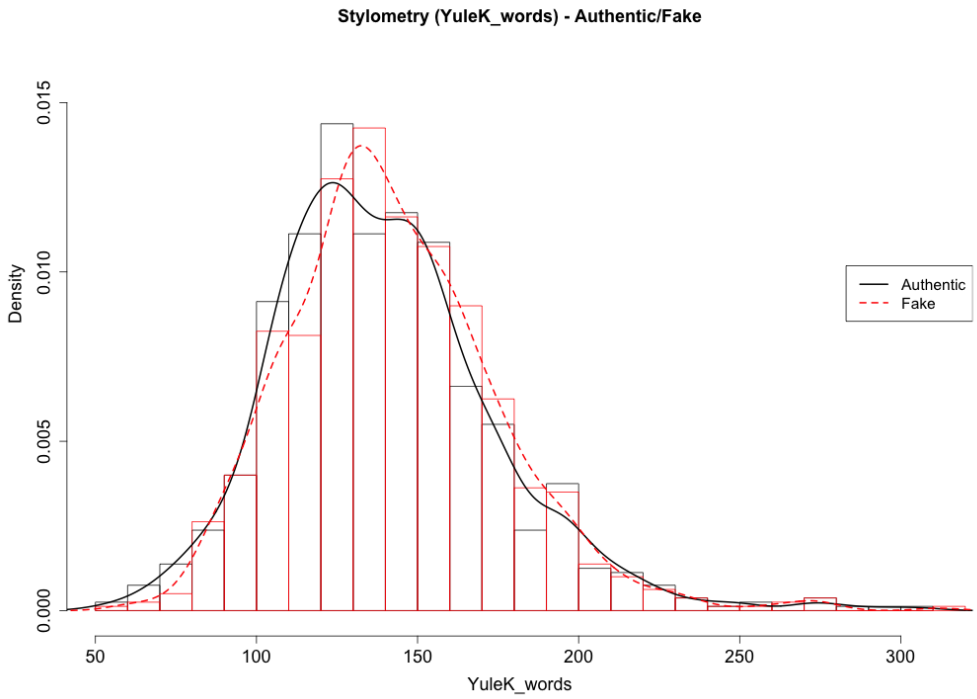

Figure 5.7: Histograms and Density Plots For Yule's K (words)

reviews within the training set and the statistical trends or differences between the authentic and fake reviews in it. How well a review in the test set maps to, or is similar to, the appropriate subset in the training set determines the accuracy of its classification. Thus the models are weak ones; devising the training set so it is representative of the entire distribution would alleviate this, but then a more sophisticated way of creating the training set is needed.

So the results are poor performance for a Decision Tree across all the feature sets as Table 5.1 shows. The possible XOR nature of the dataset is also a factor; fake negative reviews have some distinct differences to fake positive reviews and so a Decision Tree biased towards finding negatives will fail on the positives. Hyperparameter optimization may be of some use, but would require a thorough analysis of the specific nature of the features to configure the Decision Trees appropriately. Pruning overfitted trees is also an option, but scikit-learn lacks supports for this. So when the Decision Tree is Bagged, the nature of the bagging algorithm compensates for this variance in how Decision Trees perform. Using a large number of trees, all trained on a different subset, is in effect a brute force approach to the problem; the overall training set across all the trees becomes more representative of the entire set. Thus the majority opinion on whether a review is fake, or not, wins.

As for AdaBoost, the reason it consistently failed to significantly improve classifiers (and greatly degraded performance for some) possibly lies in the nature of the algorithm as well. Boosting involves iteratively training classifiers and assigning weights to the training samples based upon the current error on that sample. Weights are also assigned to classifiers

to minimize the overall error at each stage. In this manner, classifiers are added that attempt to correct for misclassified samples. If the training set is skewed towards one polarity of fake reviews over the other (negatives versus positives), then subsequent classifiers beyond the initial one could be biased towards the opposite polarity. Then when a negative fake review is seen in the test set, there could be a higher probability that it is classified incorrectly because the majority in the ensemble mistake it for an authentic review and override the classifiers who do detect negative fake reviews. Thus the overall effect is to reduce the accuracy when classifying the test set. This hypothesis could be tested by varying the distribution of polarities within the training set (both authentic and fake) in a controlled fashion and determining how that affects the performance of the boosted ensemble.

5.4 Conclusions

In general, the baseline performance of individual classifiers using different feature sets was adequate. Table 5.1 summarizes the answers to Question 1 from Chapter 3. Classifiers performed noticeably differently sometimes; this was traced back to characteristics of the data set, the features, and how the classifiers work internally. Improvements in the feature engineering to address these issues, especially for the sentiment based features, should provide benefit. The overall impression is the individual classifiers reached a limit to their performance. Hyperparameter optimization, a more thorough examination of feature selection, or feature creation would be ways to determine this.

What was unexpected was the answer to Question 2. Using ensemble methods improved performance very little, or was quite detrimental, except for with Bagging and Decision Trees. A hypothesis for this is the nature of the feature data; the features for both types of reviews have very similar distributions and so it is hard to classify things correctly as there are no very discriminatory individual features. Metaphorically speaking, training several classifiers on different chunks of the entire training set won't help if the chunks are all very similar. Thus the classifiers' diversities (and decisions) won't significantly differ and the ensemble as a whole won't have a higher accuracy. Bagged Decision Trees overcome this problem due to the randomness inherent in creating Decision Trees. The differences between fake positive reviews and fake negative ones (as for authentic reviews) also seem to be a factor in why ensemble methods do not add value; e.g., classifiers in an AdaBoost ensemble might have a strong bias towards one type of review because of the training set and so will make more mistakes on the test set if there is a different distribution of positive and negative reviews.

Chapter 6

Results with Combined Feature Sets

6.1 Overview

This chapter first presents the results from using combinations of two or more feature sets as a single one in conjunction with an individual classifier. How they performed and possible explanations as to why are discussed. The second part then examines how using ensemble methods on the models affected performance as the former serves as a baseline for the latter.

6.2 Individual Classifiers

There are 2^9 - 1 (511) distinct combinations of 9 features sets, including instances of one feature set. The 9 individual feature sets were investigated in the first phase and results can be found in Chapter 5. As training and testing was fairly efficient in terms of time required, the other 502 combinations of two or more feature sets were all subsequently examined. To reiterate, for organizational purposes, these were divided into three categories: the 11 combinations of the 4 features sets with continuous data (stylometry, readability, tone, and sentiment), the 26 combinations of discrete feature sets (lexicon, POS, POS bigram, tag, and tag bigram), and the remaining 465 combinations of both continuous and discrete feature sets. These three categories are referred to in this chapter as 'continuous', 'discrete', and 'continuous+discrete', when it is not necessary to be more specific. The classifiers used for each of the three subsets differed, based upon the nature of the features and what was possible; e.g., a Multinomial Naive Bayes classifier with continuous individual features would have required special processing and so it was not used.

A table showing the accuracy, sensitivity, specificity, and AUC of the 502 experiments is out of scope, given the page limit. Figure 6.1 however is a concise summary showing the accuracy for the 511 experiments with a LR classifier. The red line marks the highest accuracy. As for the feature sets, they are arranged thusly: first by the type of feature

set (individual continuous and discrete feature sets, combined continuous ones, combined discrete ones, and then combinations of both) and second, by increasing accuracy within that subset. This shows that from the first 9 points (the individual feature sets), the discrete feature sets tend to perform better than the continuous ones. This also applies for the second set which are the points in black: the combined continuous feature sets and then combined discrete ones.

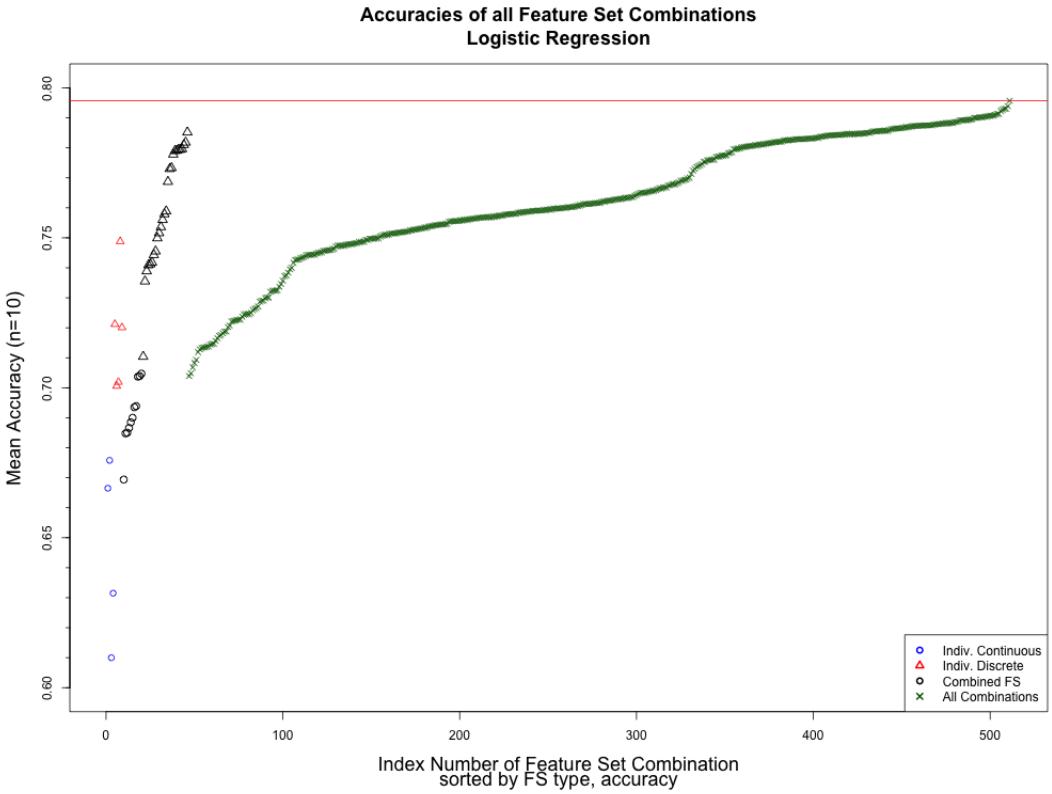

Figure 6.1: Accuracies For All Feature Set Combinations (Logistic Regression)

Figure 6.2 is the equivalent when SVM was used as the classifier. The same general trends as before are seen, with more variance in the results. It should be noted that the black triangle near the bottom around 0.60 is not an anomaly. It represents the accuracy for the combination of the POS (accuracy of 57.5%) and tag feature sets (accuracy of 61%). Given the POS feature set is a simplification of the tag feature set, it makes sense that combining the two would reduce accuracy (or increase it, from the point of view of the POS feature set). The first 8 continuous+discrete combinations (the lowest 8 green crosses) are also combinations that include the POS feature set. Comparing these to the data points associated with those feature set combinations with POS removed, it is obvious that adding a feature set can reduce the beginning accuracy. This occurs when the classifier, when using the new feature set by itself, has poor or lower accuracy. Logically, a large enough gap in the accuracies associated with the base combination and the newly added feature set would have a detrimental effect.

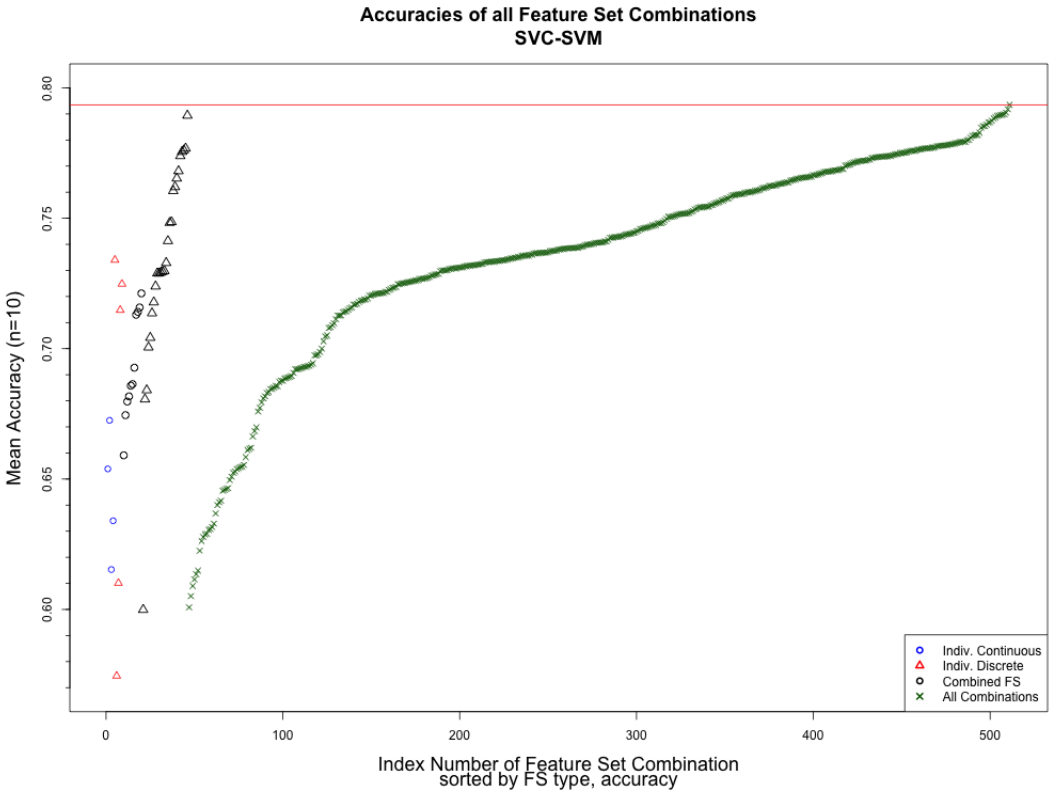

Figure 6.2: Accuracies For All Feature Set Combinations (SVM)

Therefore, feature level data fusion of feature sets of the same type adds some value, albeit under about 6%, except in the case of the POS feature set and SVM classifiers. One explanation is that the additional feature sets add knowledge about aspects of the text that buttress related information provided by the first set, which might be initially borderline in terms of determining the class. The nature of the relationship between two feature sets is also something to consider; there is certainly a deeper relationship between stylometry and readability, but not so much for stylometry and sentiment. The POS feature set is an exception, it seems, because of how inaccurate it is by itself with the SVM classifier and the deeper reasons previously examined as to why accuracy was only near chance. In effect, it only adds noise to the feature data.

Finally, the rest of the continuous+discrete combinations (the green crosses on Figure 6.1) range from an accuracy of 0.70 to 0.79. Surprisingly, this does not greatly exceed the highest value achieved for combined discrete feature sets. The first hypothesis as to why is that the addition of more individual features does not guarantee extra accuracy; given there is a fixed set of 1600 possible data points in the dataset, the curse of dimensionality means the predictive power of the classifier can decrease as the number of individual features increases. Hidden or unknown correlations between features of two different feature sets may also be a factor and this would affect the performance of a LR classifier. Statistical analysis, feature

subset selection, or PCA would be potential methods to confirm these hypotheses. Ordering classifiers differently in Figure 6.1 (e.g., based upon the feature set composition and not accuracy) may also reveal some hidden relationships between feature sets and how adding one always improves accuracy (or not). But ranking the usefulness of an individual feature set or other such questions was not the focus of this study.

Figure 6.3 compares the best of the feature set combinations (comprised of 2 to 9 feature sets, organized by accuracy) against the best individual feature set. A comparison of the values for the posbigram set and the lexicon-posbigram set (var.test() to confirm homogeneous variance, t.test() to test significance) resulted in a p-value of 0.005713. Comparing the lexicon-posbigram data to the lexicon-pos-tagbigram data returned a p-value of 0.1167. So the only significant statistical difference between these sets is between the posbigram and the lexicon-posbigram sets. This shows adding more feature sets to a combination does not automatically result in much of an improvement, but the additional feature sets may affect diversity which would be useful if these classifiers are used within an ensemble. Graphs similar to Figure 6.3 that start with a specific feature set as the base can be found on the DVD. They show how accuracy changes as different feature sets are combined with the base and what differences might be statistically significant.

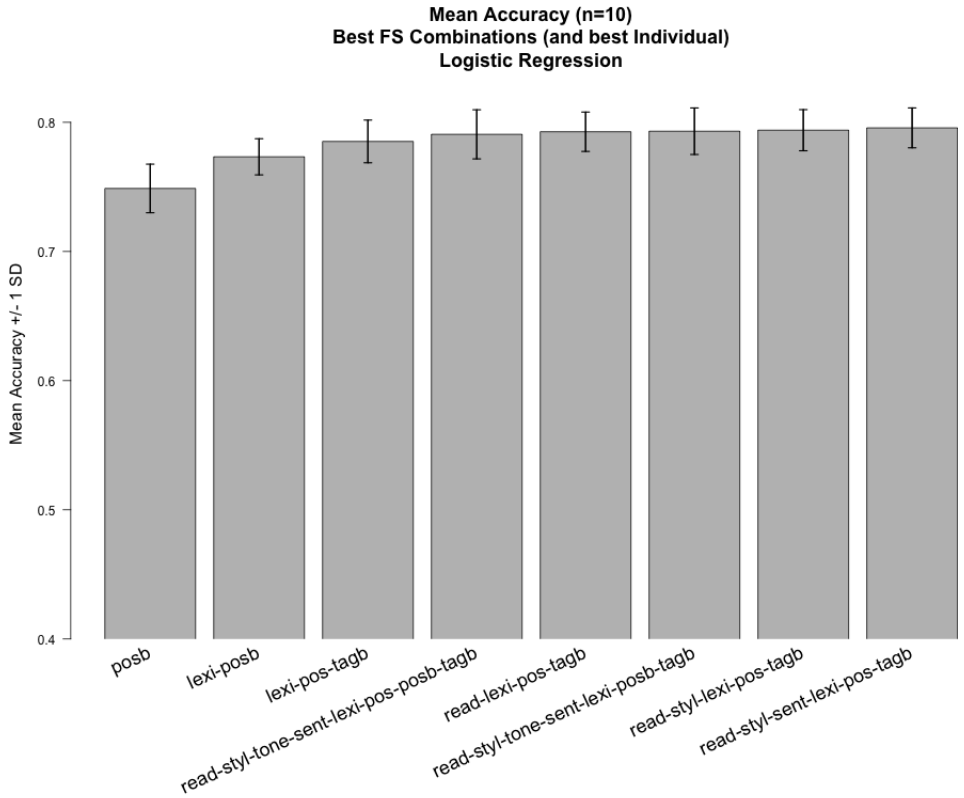

Figure 6.3: Accuracies For Best Feature Set Combinations (Logistic Regression)

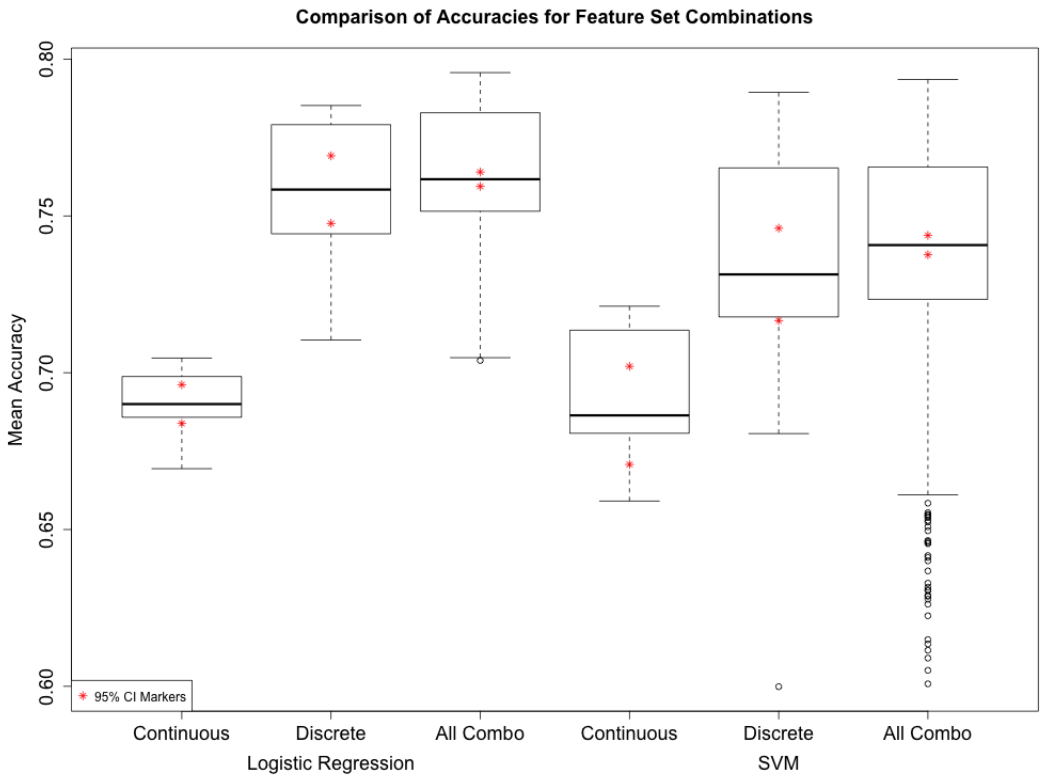

Figure 6.4: Box Plots of Accuracies For Feature Set Combinations

Figure 6.4 is another concise summary which shows the notched box plots of the accuracies of combined continuous feature sets, discrete ones, and continuous+discrete. The red stars mark the 95% confidence interval around the median. Using actually notched box plots resulted in some graphical errors due to not enough samples for the continuous feature sets. From this figure, the conclusion is that there is a likely significant difference between the continuous and discrete feature set combinations, but not between the discrete and continuous+discrete combinations.

6.3 Individual Classifiers and Ensemble Methods

The same general trends seen when using ensemble methods with classifiers that use individual feature sets was seen in the results for classifiers that use combined feature sets. To wit, only Bagging of Decision Trees improved accuracy significantly and AdaBoost tended to dramatically decrease performance. Figure 6.5 is one example; the other graphs have been omitted in the interests of space and can be found on the DVD. Again, the classifiers listed in each graph depend on the nature of the individual features as with ensemble methods and individual feature sets in the previous chapter.

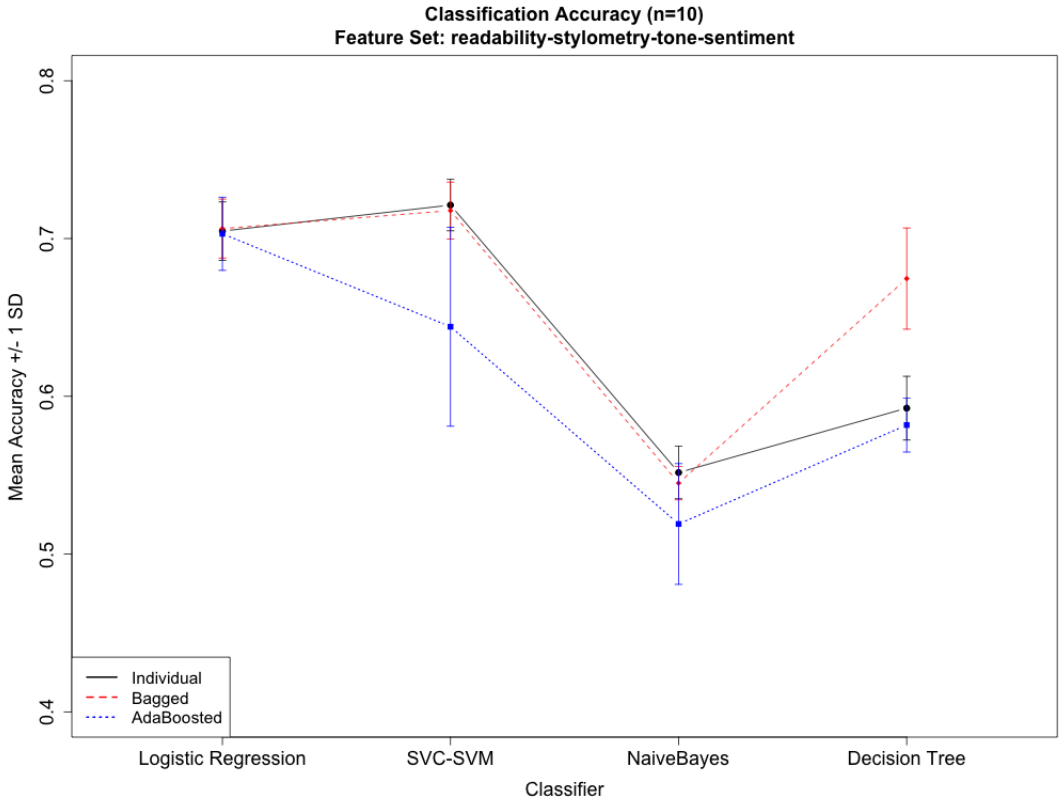

Figure 6.5: Accuracies For Readability-Stylometry-Tone-Sentiment Combination FS

Only the 37 combinations of combined continuous and combined discrete feature sets were tested. Upon calculating that ensembling the other 465 combinations would take at least 4 days of continuous operation, the experiment was halted. Preliminary results from ensembling some continuous+discrete combinations did not show promise. Also, there was no sign from the first 37 trials that an ensemble method might result in a noticeable improvement for the other 465 (except for Decision Trees, based on the conclusions discussed in Section 5.3). The ensemble methods of bagging and boosting are designed to reduce variance or bias in the classifier decision, not the feature data. So if combining feature sets into one improves accuracy (however little) it is not immediately obvious how variance or bias might be increasing. Therefore, to acquire a higher performance beyond that of the feature set combinations in Figure 6.1, ensemble methods were deemed useless as in Section 5.3.

To double check how adding the POS feature set might affect results when ensemble methods were used, the combinations that use it were calculated. A quick review of the graphs similar to Figure 6.5 revealed bagging and AdaBoost again did not improve performance with any classifier expect Decision Trees. These graphs also showed, when compared to those for combinations with POS removed, that adding the POS feature set was slightly detrimental, but not significantly (typically about 1%). This was when the classifier was a SVM; LR classifiers benefited in the same range when the POS feature set was added.

6.4 Conclusions

As all possible combinations of the 9 feature sets were examined, the definitive answer to Question 3 is that feature level data fusion has a modest impact in improving classifier accuracy, but is dependent upon certain factors. The primary one is whether dissimilar or similar types of feature sets (continuous or discrete) are being combined. Similar feature sets reinforce each other in a positive fashion, but further analysis is needed to fully investigate how the feature sets relate to one another to understand the results. A second factor is probably the curse of dimensionality which impacts classifier performance when individual features are increased and the size of the dataset is fixed. Thus adding more and more feature sets together does not automatically results in more accuracy. A larger dataset of reviews would aid in examining this hypothesis. Feature sets that do not work well with a specific type of classifier also can have a detrimental effect when combined with other feature sets. This implies something of a relationship between how a classifier works and the nature of the feature data. Finally, the answer to Question 4 is the same as with Question 2: ensemble methods do not generally aid in improving the classifier accuracy based on the same reasoning explained in Section 5.3.

Chapter 7

Classifier Ensemblement

7.1 Overview

This chapter examines the results of ensembling the models created in previous phases in the same methodical manner that maps to how the models were devised. Also included is an explanation of how the analysis at each stage led to a more refined understanding of the relationship between classifier accuracy, ensemble accuracy and diversity. How and why analytical techniques were developed that addressed the problem of an exponentially growing number of ensembles are also presented.

7.2 Classifiers that Use Only Individual Feature Sets

Looking at only the four continuous feature sets, Table 5.1 shows there are 9 classifiers (using an individual feature set) with an accuracy above 0.6. These nine were used as the initial pool and were ordered by decreasing accuracy. 0.6 was chosen as the cutoff to include the sentiment related classifiers; the range between the best accuracy (0.675) and the worst (0.61) lead to an hypothesis that using the latter would be very counter-productive. But as Figure 7.1 shows, this was not true. This figure compares both the minimum and maximum of the averaged (over 10 runs) accuracies of the ensembles of a specific size. The blue and black lines denote the maximum and minimums, while the red and green lines denote the averaged accuracy of the individual classifiers within the two ensembles. So blue and red are paired together and black and green. Graphing both minimum and maximum at the same time allows for an easier comparison of how increasing ensemble size affects ensemble accuracy (for this particular classifier pool) as well as how ensemblement improves performance. The error bars of 1 standard deviation make the image a bit complex, so the black and green lines were slightly offset. As the pool was so small, it was possible to evaluate all possible ensembles and to do so using 10 random seeds and the training and test datasets.

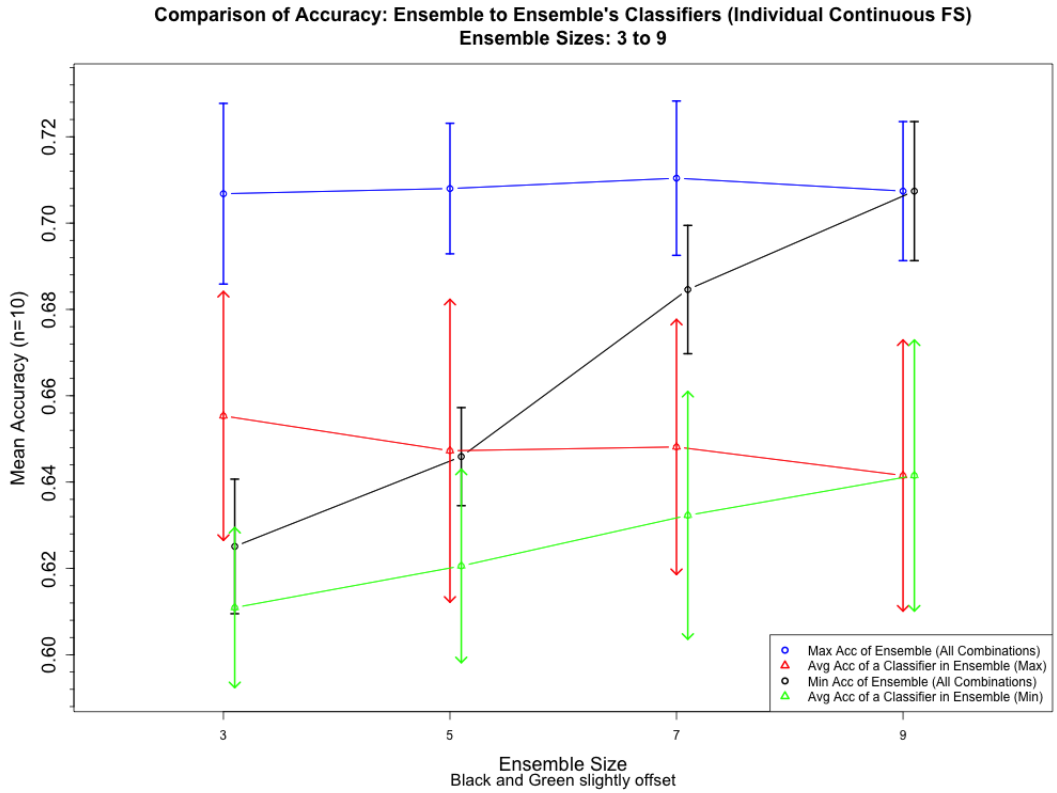

Figure 7.1: Ensemble Accuracy Versus Averaged Classifier Accuracy (Individual Continuous Feature Sets)

Comparing the blue and red lines reveals increasing the ensemble size (to all nine classifiers) slightly lowered the average classifier accuracy, but the ensemble's maximum accuracy stayed relatively constant. Thus adding less accurate classifiers did not impact ensemble performance much. And graphing the minimum (the black and green lines) reveals how performance improves as the ensemble gets larger and more accurate classifiers are added. The difference between the two ensembles' accuracies decreases notably as better classifiers are added. The overall conclusion then is ensemble creation should first use the most accurate classifiers. If worse classifiers are added, their impact will likely be small, up to the point where too many of them override the decisions of the best classifiers. And if an ensemble initially starts out with poor classifiers, adding more accurate ones will improve ensemble accuracy, and possibly significantly.

Figure 7.2 is the same diagram for the ensembles created from classifiers using individual discrete feature sets; the accuracy cutoff limit was 0.7. This limited the classifier pool to 11 and so examining all possible ensembles was tractable. The same trends in Figure 7.1 are seen; the maximum ensemble accuracy's larger drop is due to more classifiers at the low end of the accuracy scale (5 were under 0.72) than in the previous experiment. And the difference in accuracy is again fairly significant (about 5 to 6%).

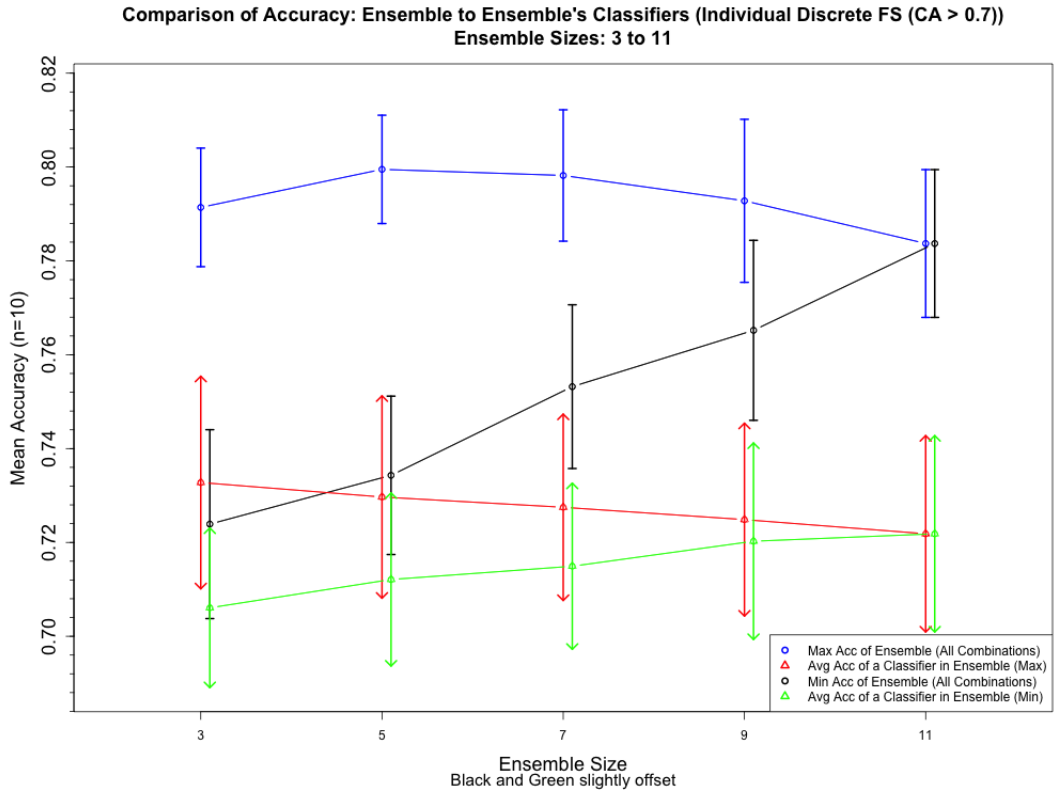

Figure 7.2: Ensemble Accuracy Versus Averaged Classifier Accuracy (Individual Discrete Feature Sets)

But when the cutoff limit was set to 0.6, the number of classifiers in the pool expanded to 20. The accuracy ranged from 0.748 to 0.610, so it was unlikely those at the bottom would contribute significantly to the ensemble. The acceptable cutoff limit was unclear, however. Using 0.69 would prevent two at the high end of 0.68 from being included and if 0.68 was used (or even 0.69), a complete analysis of all possible ensembles was impractical. To get an initial understanding of the data, the first 10,000 ensembles (the index number being that determined by the Python 'combinations' function) and their accuracies were calculated and graphed for ensemble sizes of 3, 5, and 7. As the ensemble size increased, a pattern started to reveal itself. Figure 7.3 shows the first clear picture.

The sawtooth type pattern was unexpected, but upon reflection, a reasonable hypothesis was devised. The classifiers were first arranged in the pool by decreasing accuracy. Then the 'combinations' function was used to select and combine them into an ensemble in an ordered fashion. For instance, the first 5 most accurate classifiers formed the first ensemble, then the second ensemble was composed of the first four and the sixth, skipping the fifth. The third ensemble used the seventh classifier in place of the sixth; thus the ensembles were ordered in a fashion corresponding to the order of the classifiers in the pool. So the overall trend downwards in accuracy was due to less and less accurate classifiers being added, or replacing

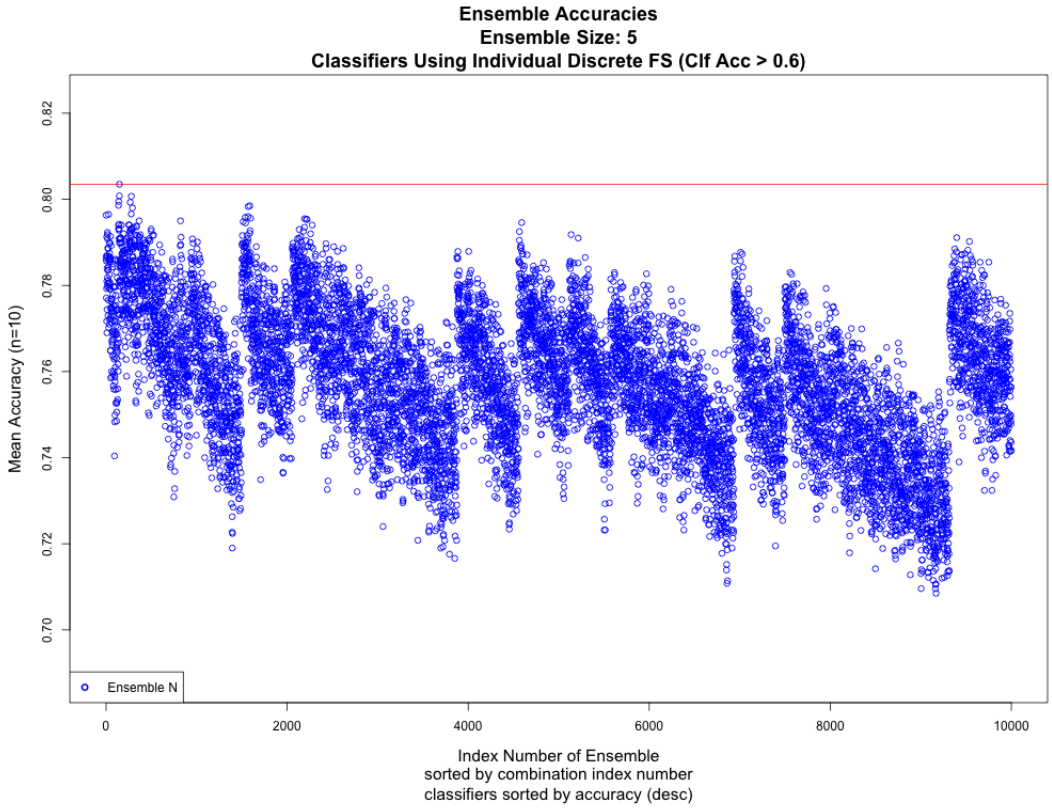

Figure 7.3: Accuracy Of First 10,000 Ensembles

the previous more accurate ones. This behavior is congruent with that seen in the graphs for ensembles using individual continuous feature sets. An important thing to note is that the best ensemble accuracy seen (the red line) is not associated with the very first ensemble (that composed of the five most accurate classifiers). Instead, the most accurate ensemble on this graph is one that comes later in the ordering, specifically the 148th ensemble that was created. The classifiers used in this ensemble were the 1st, 2nd, 4th, 5th, and 17th. Obviously the accuracy of each classifier in an ensemble is not the only important factor that determines the ensemble's accuracy.

The breaks, or jumps, in accuracy then became the focus. At what point in the ordering scheme does this happen and why? The initial hypothesis was that these breaks denoted roughly where the initial classifier used was moved forward, e.g., from the first classifier to the second. Thus the pattern gets reset, in a metaphorical sense. This pattern is also seen more distinctly in the plot when the ensemble size is 7 as it is denser. From this reason, further work was done using that dataset.

To verify this hypothesis, a plot was made where the color of the points was determined by the initial classifier, e.g., if the initial classifier was the first, points would be blue, if the second classifier, then red. In the process of writing the R code to create the plot, there was a realization that the first 10,000 points were not going to include any ensembles beginning

with the second classifier. This is due to the exponential growth in the total number of ensembles as the classifier pool size increases. Calculations showed the first 8568 ensembles all began with classifiers 1 and 2. So they were used as the basis in developing Figure 7.4. Each color represents what classifier is next in the ensemble list, past the first two classifiers. It is apparent breaks are occurring when the third classifier changes.

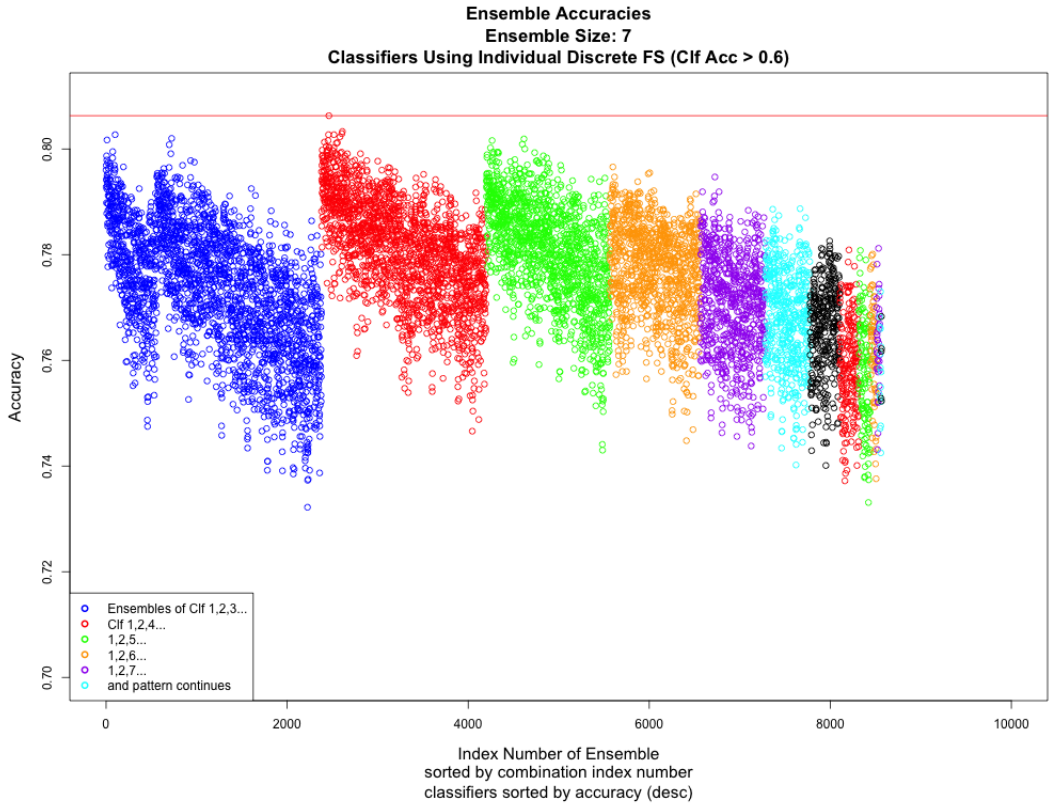

Figure 7.4: Accuracy Of First 8568 Ensembles

The obvious explanation for both the breaks and the fact that the best ensembles were not composed of only the most accurate classifiers is diversity. Why else would an ensemble made of the 1st, 2nd, 4th, 5th, and 17th be the most accurate? Just enough diversity would prevent the classifiers within an ensemble making the wrong decision from becoming a majority. So classifier 3, in Figure 7.3, agrees too much with the ones making the wrong decision, i.e., it does not sufficiently contribute to overriding the bad decisions in the same way as classifier 17 does. In other words, classifier 17 makes the correct decisions that perhaps 3, 4, and 5 do not and 1 and 2 do.

Referring to Figure 7.4, the blue points represent an ensemble that starts off with classifiers 1, 2, and 3. The red points are those that start off with 1, 2, and 4. But 4 has a lower accuracy than 3, so why is there is a notable increase in ensemble accuracy? The two classifiers in question are a Multinomial Bayes and SVM that use the posbigram and the tagbigram feature sets respectively (accuracies of 0.7337 and 0.7248). The pairwise disagreement between

the two was calculated between 0.25 and 0.31 (depending on the random seed and thus the training set). The average of 0.28 might seem low, but it implies about a third of the decisions of the two classifiers are different despite the accuracy of the two being within 1%. Therefore, based on this analysis, it is apparent how crucial a factor diversity can be. It succinctly explains the sawtooth pattern as well as the more accurate ensembles not being immediately at the beginning of the graph. Any number of measures (e.g., overall ensemble diversity) or a different ordering scheme could potentially result in revealing other useful patterns, but this line of analysis was not performed due to the time required.

7.3 Classifier Selection Schemes

The next set of experiments proceeded by using both groups (classifiers using individual continuous and individual discrete feature sets) in an ensemble. It was here that the problem of exponential growth became apparent; there were 29 possible classifiers if 0.6 was used for the cutoff. If 0.7 was used, then there were only 11, but that list was the same as the experiment using just individual discrete feature sets. If 0.65 was used, then only two of the continuous feature set related classifiers would be in the classifier pool. As an exhaustive search was infeasible, the idea of classifier selection schemes arose. If the sampling performed by these schemes was able to sample the top of the sawtooth pattern and verified, then a reasonable estimate of how the best ensembles performed could be made. There was no guarantee the absolutely best ensemble could be found, of course. But the purpose of this study is not to find it, but to gauge the benefit of creating custom ensembles in general.

The key parameters for a scheme are what measures are used in ordering the classifiers, the function to do so, and the direction of the ordering. Table 7.1 list those devised; Python's partial function library was instrumental in allowing different schemes to be quickly developed. A wide variety was used as it was unclear what might result, so varying the ordering would enable comparisons to be made. As for pairwise diversities, after a classifier pool was initially ordered according to decreasing accuracy, the pairwise diversities for each classifier to the others was calculated and stored. The average of those became a characteristic associated with the classifier. The median of the diversities was also computed; the basis for this idea was to use the set of pairwise diversities as a distribution. To average them reduces the distribution down to one number, but the distribution could be skewed and an average would not capture that knowledge. Using the median or the median with the standard deviation could possibly better reflect how a classifier was similar to the others as a whole.

Table 7.1: Classifier Selection Schemes

Name	Description	Ordering
ByAcc	Classifiers ordered by accuracy	High to low
ByAcc1	Like ByAcc, with first classifier removed from list	High to low
ByAcc2	Like ByAcc, with first 2 classifiers removed	High to low
ByAcc3	Like ByAcc, with first 3 removed	High to low
ByDiv	Classifiers ordered by average of pairwise diversities	High to low
ByDiv-R	As above, but reverse ordering	Low to high
ByAccDiv	Ordered by accuracy * average diversity	High to low
ByAccDiv-R	As above, but reversed	Low to high
ByAccMedDiv	Ordered by accuracy * median pairwise diversity	High to low
ByAccMedDiv-R	As above, but reversed	Low to high
ByAccMedSD	Ordered by accuracy * median * std dev of diversity	High to low
ByAccMedSD-R	As above, but reversed	Low to high
ByAccDivRO	Like ByAccDiv, but first third moved to end of list	Order modified
ByAccFS	Like ByAcc, but feature sets a factor in selection	High to low
ByAvgDivFS	Like ByAccFS, but using ByAvgDiv ordering	High to low
ByAccDivFS	Like ByAccFS, but using ByAccDiv ordering	High to low
ByAAD	Classifiers chosen by accuracy, then accuracy*diversity	High to low

The last 5 schemes were specifically developed to test ideas about how the ordering might be optimized so more useful classifiers were higher in the list. The schemes involving the feature sets first ensured classifiers using different feature sets were added, skipping over duplicates even though they might be next in terms of accuracy. Once all feature sets had been processed, the selection process rotated back to the top of the list, selecting from the unused ones. The final scheme first used the top 3 classifiers, then reordered the rest based on a measure of accuracy multiplied by diversity.

Once the ordered classifier pool had been assembled, the first N classifiers were selected while looping over the ensemble size (from 3 to as many classifiers as possible). Two types of diagrams were plotted to evaluate the behavior and performance of the schemes. The first example, in Figure 7.5, is a strip plot that allows for the 17 schemes to be quickly compared in terms of their distribution and variance. The red line marks the maximum ensemble accuracy over all schemes. As it shows, scheme 16 (ByAAD) resulted in the highest ensemble accuracy, equivalent to number 1, while 5, 7, and 9 found comparable ensembles.

The reason for the large variance for some of the schemes is undoubtedly due to the mixing of less accurate classifiers amongst the better classifiers, due to different ways diversity was used in the ordering formula. This is supported by the fact that all the schemes where the ordering was strictly by decreasing accuracy (schemes 0, 1, 2, 3, and 13) has the smallest

variance. The general decrease in the maximum accuracy between schemes 0, 1, 2, and 3 also support this (1, 2, and 3 ignore the top classifiers). The behavior of schemes 0 to 3 reflect what is seen in the plots of the first 10,000 ensembles. But the slight increase in the maximum accuracy for scheme 3 is something to note; it is a sign that the way classifiers were picked increased diversity somewhat and so the ensemble's performance (or range of it) did not deteriorate as badly as it did between schemes 1 and 2. Comparing the distributions of these sets of values to the distributions of the ensembles' overall diversity measures may reveal a connection.

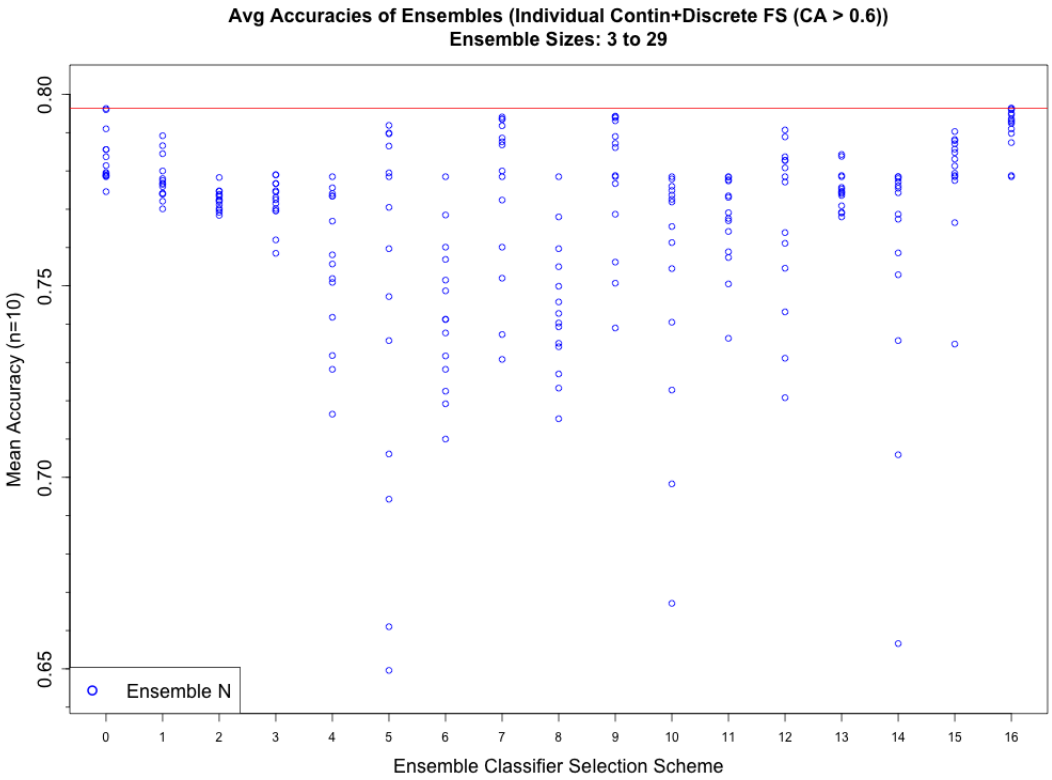

Figure 7.5: Strip Plot Of Ensemble Accuracy Per Selection Schemes

The next example (Figure 7.6) is of the same data in Figure 7.5, but graphed according to the ensemble size. The lines provide a view of the dynamic behavior of the ensemble accuracy as its size changes. It also shows the results of each scheme's sampling of the whole set of ensembles. Figure 7.6 only displays the notable schemes; the schemes involving the median, or the median and the standard deviation, did not perform substantially different than those using the average of pairwise diversity. The slopes of the lines tended to be roughly equivalent; only the accuracy at each point went up or down slightly. As Figure 7.6 shows, the ByAAD scheme resulting finding ensembles that were fairly consistent in their accuracy despite the increasing number of classifiers. The ByAcc scheme shows how including the classifiers based strictly on accuracy resulted in a faster decreasing ensemble accuracy. Finally, the

schemes that were reversals of each other (e.g. schemes By Div and ByDiv-R) behaved in an unexpected way. Initially, it was thought they would be roughly mirror images of each, along the horizontal axis. Instead, there is a general sense that they are rotated around the straight line that connects the first point and the last. This is only a supposition and unclear what it might imply, except that the impact of diversity is not entirely straightforward. Further analysis needs to be done, including graphing the scheme that involves ordering the classifiers by their accuracy in an increasing manner.

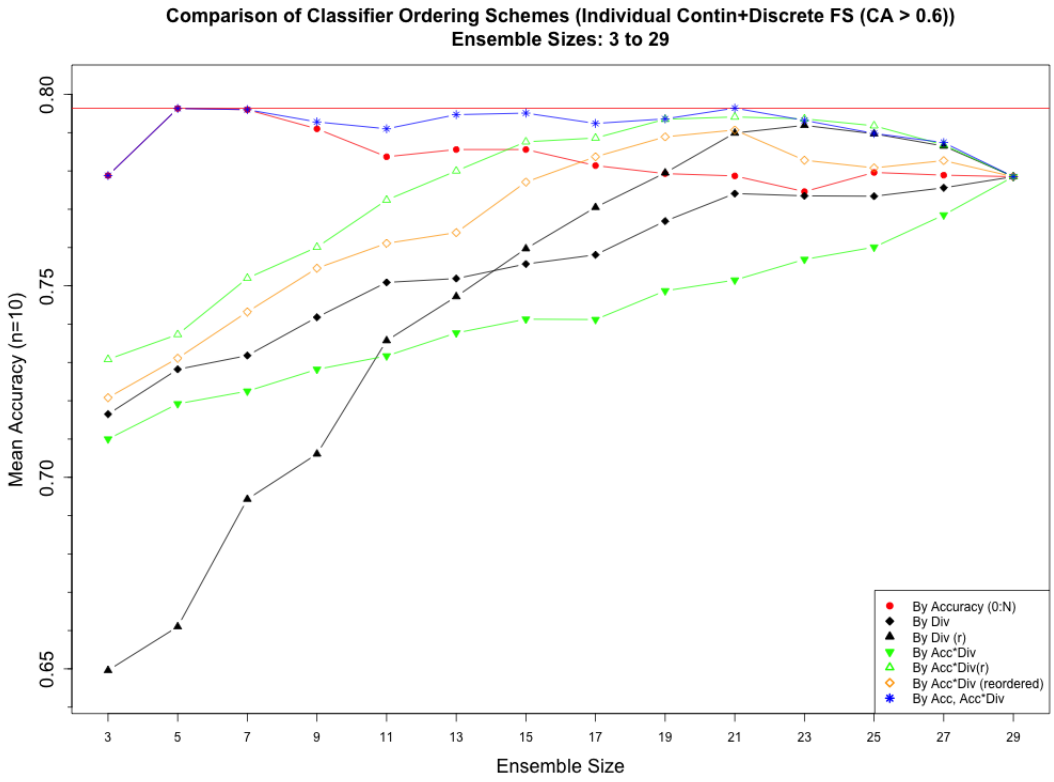

Figure 7.6: Line Plot Of Ensemble Accuracies Per Ensemble Size By Scheme

When the ensemble size was 29, all classifiers had been used, so consequently all schemes ended at the same point. The benefit of displaying how ensemble accuracy changes as size does, and based on the ordering of classifiers, reveals (in a qualitative manner) how diversity and individual classifier accuracy affect ensemble accuracy. A thorough analysis of all the data gathered is planned for a subsequent paper, but in general, there was no one scheme that was always the best to use. The composition of the pool and the inter-diversity of the classifiers were obviously important factors.

Figure 7.7 is a comparison of the overall ensemble accuracy (using scheme 16 and averaged over 10 runs) and the average of the individual classifiers' accuracy (when using both continuous and discrete feature sets). The standard deviation is shown as well. As the cutoff for classifier accuracy was 0.60 for individual classifiers, it is noticeably large, but

for the ensemble as a whole, it stays fairly steady. It is obvious how ensemblement aids in stabilizing the overall accuracy, even as less accurate classifiers are added as the ensemble grows larger.

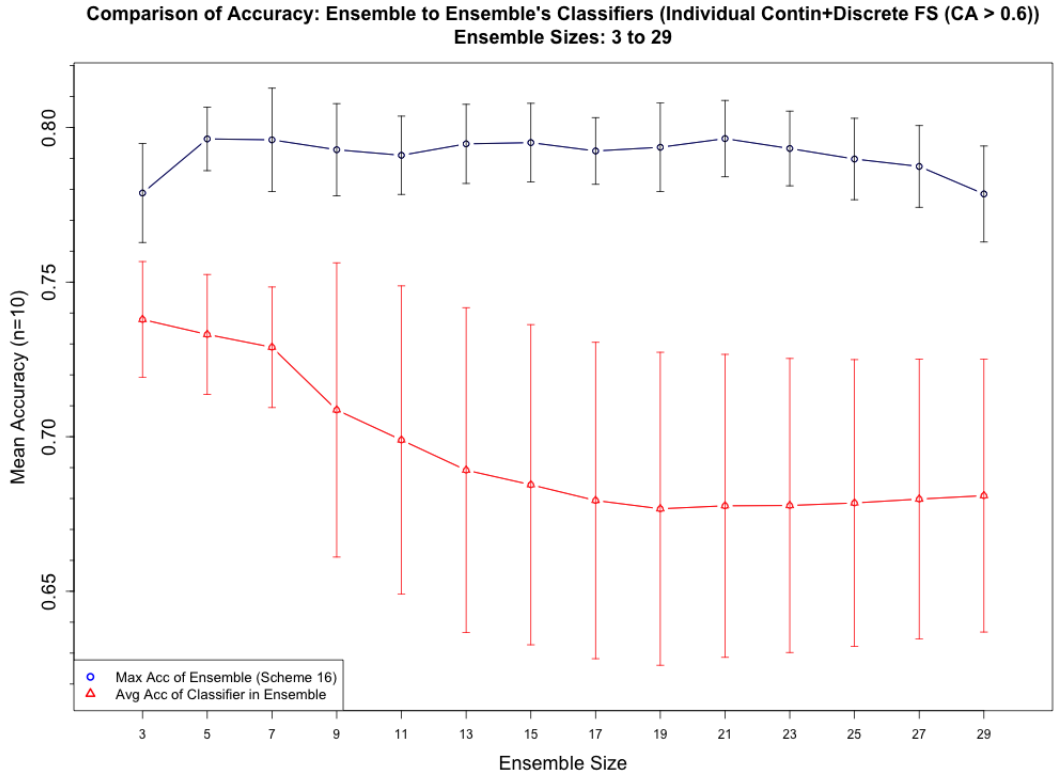

Figure 7.7: Ensemble Accuracy Versus Averaged Classifier Accuracy (Individual Continuous+Discrete Feature Sets)

Figures 7.8 and 7.9 are graphs of how the mean sensitivity and specificity changed as the ensemble size increased (for different schemes). Comparing the two reveals something interesting, which was noted in other similar graphs. Sensitivity (the ability to correctly detect when a review was fake) displays an overall trend of increasing as the ensemble size goes up, however the classifiers were ordered. But specificity decreases slightly or increases depending on the initial point. The schemes (not including the first) vary how classifiers are ordered in the pool based on different ways of weighting diversity, so it seems the ability to correctly detect authentic reviews (e.g., not make type I errors) is dependent on diversity in some manner (as well as accuracy).

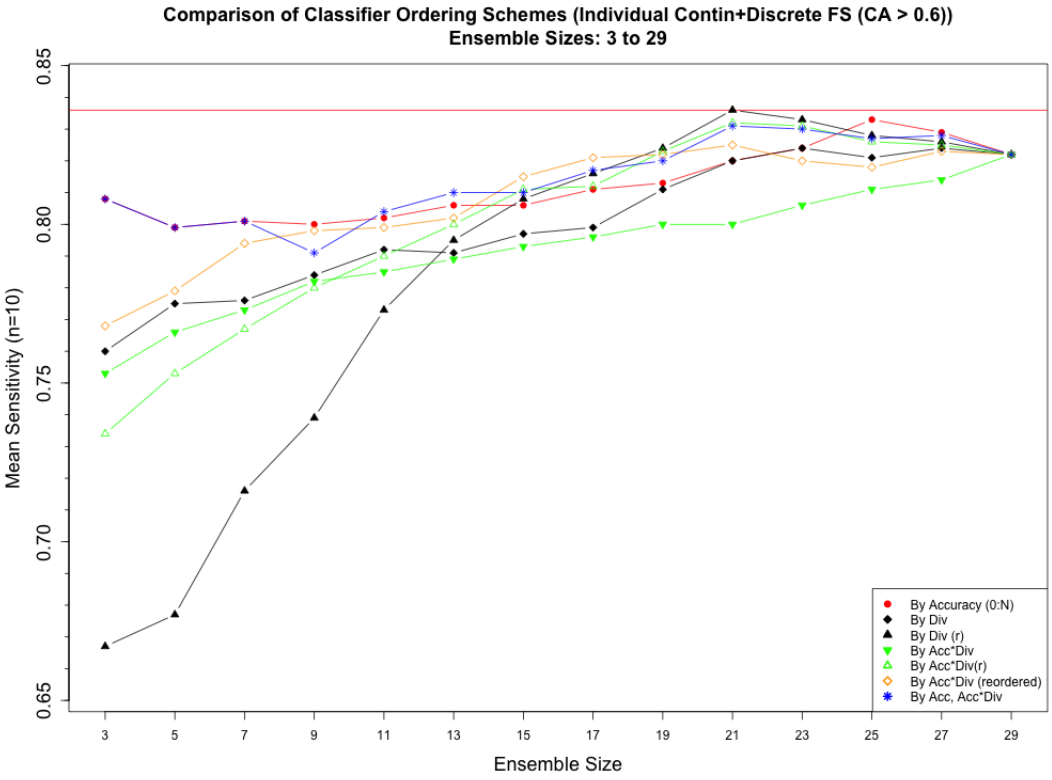

Figure 7.8: Mean Ensemble Sensitivity Per Ensemble Size By Scheme (Individual Continuous+Discrete Feature Sets)

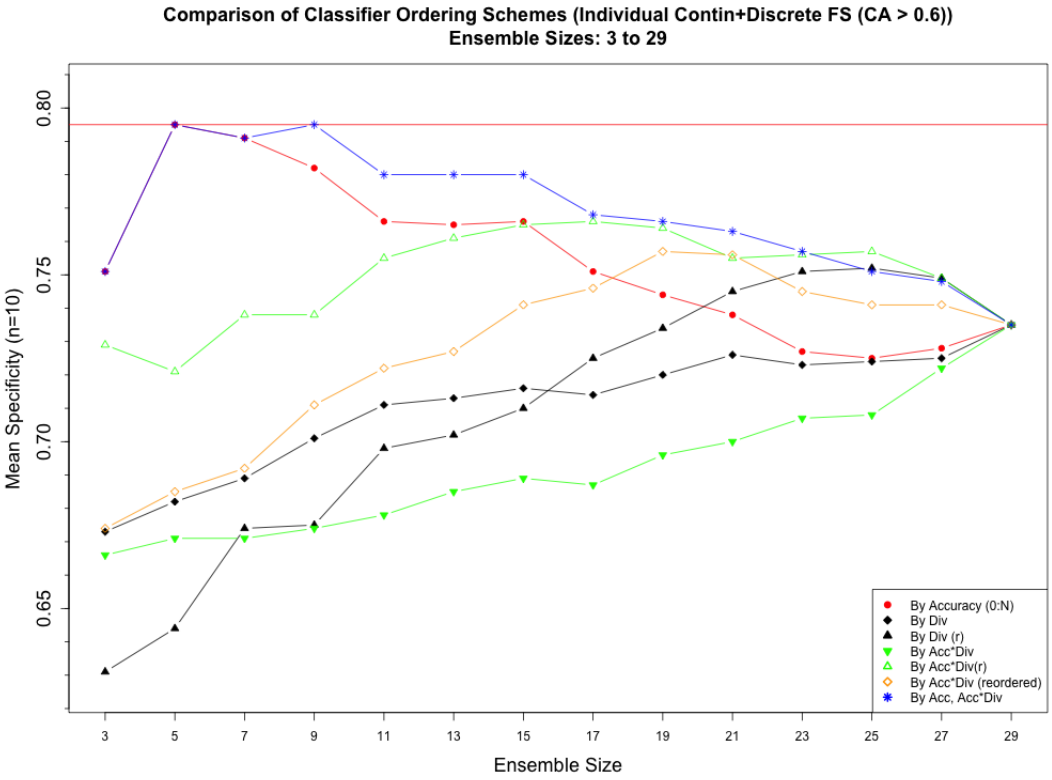

Figure 7.9: Mean Ensemble Specificity Per Ensemble Size By Scheme (Individual Continuous+Discrete Feature Sets)

7.4 Classifiers That Use Only Combined Feature Sets

The next phase involved creating ensembles from the classifiers that use combinations of feature sets as input data. Figures 7.10, 7.11, and 7.12 are examples of how the ensemble accuracy changed as the size increased (all for scheme 0). In Figure 7.10, as in Figure 7.7, there are signs of the stability of the ensemble accuracy (and its standard deviation) even as ensemble size increases and the averaged individual classifier accuracy decreases (and its standard deviation increases). This trend is even clear in Figure 7.11.

The sudden change in accuracy at certain points indicate places to investigate, i.e., what the diversity of the ensemble and the classifiers is like at those points, or how did the classifier composition change. Figure 7.11 is a good example of this; a sudden drop of about 2% was certainly dramatic. But Figure 7.12 is noticeably different; the differences in accuracies was very little and the standard deviations greatly overlap even as ensemble size increases. This implies ensembling was becoming less effective. In other words, the difference between the average individual classifier accuracy and the ensemble accuracy was generally decreasing, at least less than that seen in the first phase that used less complex individual classifiers.

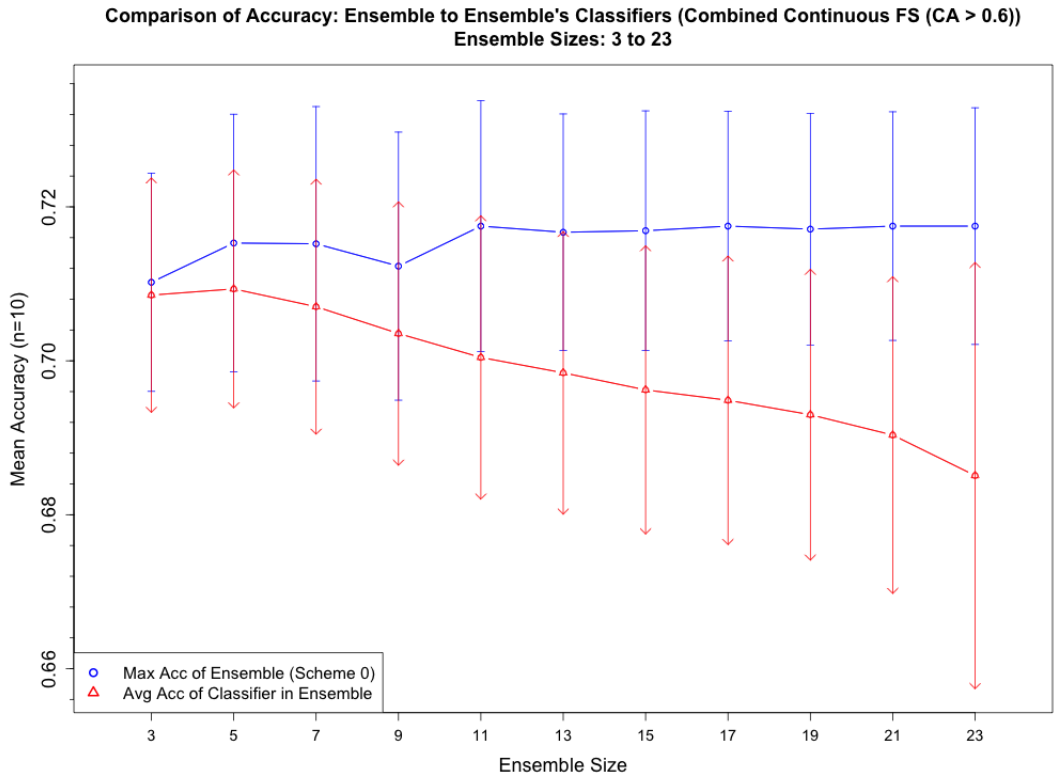

Figure 7.10: Ensemble Accuracy Versus Size
(Combined Continuous Feature Sets)

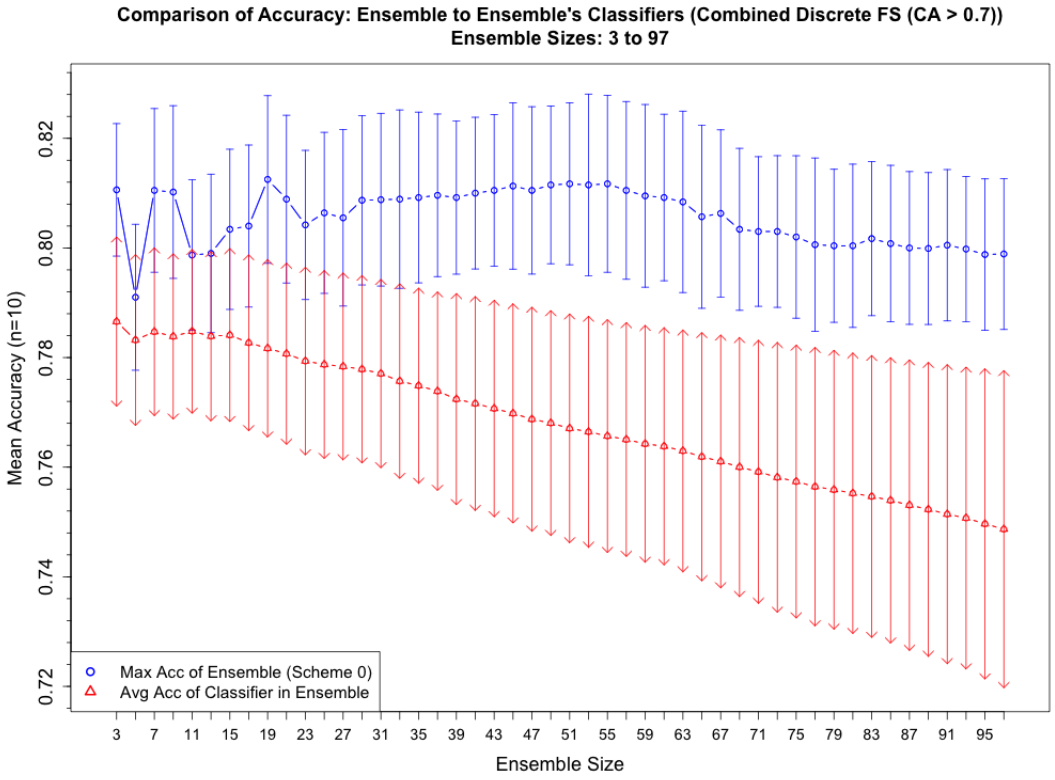

Figure 7.11: Ensemble Accuracy Versus Size
(Combined Discrete Feature Sets)

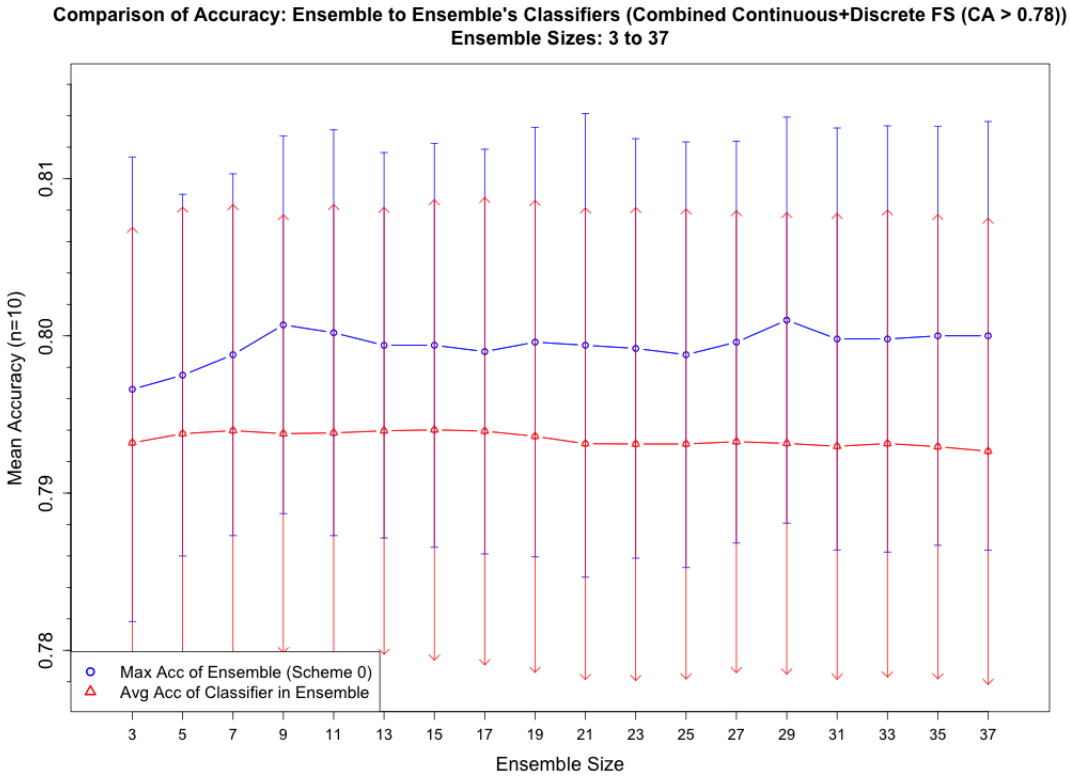

Figure 7.12: Ensemble Accuracy Versus Size
(Combined Continuous+Discrete Feature Sets)

The roots of this problem were traced back to two factors. The first was again the need to choose an optimal cutoff for selecting classifiers to be in the initial pool. As the classifiers became more complex (in terms of the type of feature set used), the possible choices kept increasing. The other factor was again diversity. The top classifiers in the first scheme (that ordered by accuracy) were becoming very similar in terms of their pairwise diversity. This initial list is the basis for the other schemes, so reordering the list might provide some benefit, but ultimately too similar classifiers ended up being ensembled. To summarize, if a cutoff point was chosen to reduce the pool to a manageable size, then the likelihood of them being too similar (and thus affecting ensemble performance) increased.

7.5 More Complex Ensembles

At this point, the next question became clear: how to winnow the set of potential classifiers even before selecting them using a scheme to prevent ones that are too similar from being within the pool. The next step, of ensembling classifiers that use combined discrete and combined continuous+discrete feature sets, was briefly investigated to verify this (combined continuous was not included because their accuracy was not comparable). The number of classifiers with an accuracy greater than 0.78 (the maximum being 0.795) was 184; the top 24 alone had an accuracy greater than 0.79. Thus the next step in creating effective ensembles that have a sufficient gain in accuracy becomes how to manage the classifier pool. Winnowing has only been briefly examined due to time limitations and is discussed in the next chapter. In the process of the investigation, some preliminary theories have been developed on the relationship between classifiers' accuracy, their pairwise diversities, how classifiers cooperate or conflict, and the effect on ensemble accuracy. Figure 7.11 was instrumental in this based on the analysis of the downward spike.

7.6 Conclusions

Ensemblement by combining multiple classifiers with a majority voting rule (the topic of Question 5) does sufficiently increase accuracy beyond that of any of the individual classifiers such that it is worth the effort. Therefore the basic idea of using multiple feature sets in parallel, including combined feature sets as the feature set, to enhance classification accuracy shows promise. This is contrary to the results of using ensemble methods on a single classifier with a single feature set (however complex) to construct an ensemble. But there are at least two

important factors: the number of potentially useful classifiers and their diversity. This was established through a methodical analysis of the ensembles by increasing the complexity of the classifiers and the input data in a controlled fashion. As the complexity increased, these two factors became limitations in searching the entire space of possible ensembles.

Schemes therefore were developed to more efficiently investigate possible ensembles through a sampling process. This process is based upon how classifier are ranked and the way they are selected. But schemes were still not sufficient as the pool size increased (and the most accurate classifiers became more similar), so this lead to a preliminary investigation (discussed in the next chapter) on how to winnow the classifier pool. But it is clear for this dataset at least, if not for all text, that classification systems should involve analyzing the text in multiple simultaneous ways to create feature sets in order to generate diverse classifiers. Ensemble methods generally do not do well given how they try to improve accuracy by manipulating only one feature set's data to diversify the classifiers.

Figure 7.13 is a critical difference diagram of the some of the ensembles developed. The names reflect the types of feature sets used. To properly compare them, the ensemble size was set to 9 (so the simplest ensemble of individual continuous feature sets could be compared) and the selection scheme was set to the first (one based on ordering classifiers by their accuracy). This is not a thorough comparison; to do so require more analysis across all the variables and more ensembling. But it provides some insight into how the ensembles differ based on which feature sets are used. Comprehensively addressing the problem of diversity would have a positive effect on accuracy for the ensembles where schemes were used.

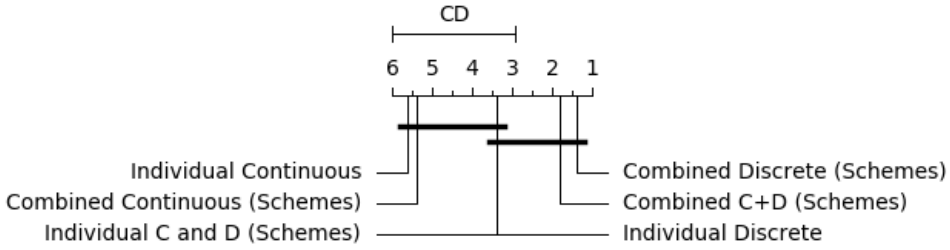

Figure 7.13: CD Diagram Of Ensembles (alpha=0.05, test=Nemenyi)

Chapter 8

Accuracy And Diversity

8.1 Overview

This chapter details the preliminary investigation into how a classifier pool might be winnowed to reduce the size before assembling ensembles. The intent of this step is to make the classifier selection and sampling process described in the previous chapter more efficient by eliminating classifiers that are too similar. The algorithm and the rationale behind it is first outlined, followed by examples of the analysis that led to its development.

8.2 The Winnowing Process

As part of the investigation into creating custom ensembles, the pairwise diversities between classifiers initially selected were calculated and saved. It was uncertain of what use these measures might be, so the disagreement measure was used as it is a straightforward calculation. The overall diversity of each ensemble was calculated as well using the entropy measure E as it was analogous to the disagreement measure. When it became apparent that winnowing the pool was necessary for efficiency, the question that arose was "how to determine how similar a classifier was to another?". Individual pairwise measures evaluate the relationship between two classifiers, but how could the relationship between a classifier and all others in an ensemble or a pool (as a whole) be evaluated? Based on this, treating all the pairwise measures for one classifier as a vector becomes an obvious idea. Then two classifiers could be labeled as 'similar' if their vectors are similar enough as defined by an appropriate distance measure.

Clustering was the first idea for evaluating the closeness of all the vectors. Then all the classifiers within a cluster with a low enough cohesiveness could be reduced to a single representative. But that is an entire research project in of itself and deadlines were approaching. To gain an initial understanding of what be might possible, a short circuit was devised. The

intent of the winnowing is to eliminate classifiers, so all that matters is how close two classifiers are in terms of their diversity vector. Looping over the list of classifiers, the distance between two vectors could be calculated and specific classifiers marked for winnowing.

The software was adapted to calculate all these different vectors (the distance measure chosen was the cosine similarity) and the matrix of vectors for the classifiers (the vector of cosine similarities, not pairwise diversity) displayed for analysis. Comparing these matrices to the graphs of the results of the selection schemes resulted in some preliminary understanding of why the ensemble accuracy, for each scheme, varied as the ensemble size increased and how the classifier pool was ordered. The next section is a detailed analysis of several figures from Chapter 7 to explain this further, but a simple example now follows.

Table 8.1: Example Diversity Vectors

Classifier	Diversity Vector		
One	0	0.54	0.23
Two	0.54	0	0.26
Three	0.23	0.26	0

Table 8.2: Example Similarity Vectors

Classifier	Similarity Vector		
One	1	0.998	0.948
Two	0.998	1	0.921
Three	0.948	0.921	1

An important thing to note is that calculation of the similarity vectors involves removing two of the numbers of the diversity vector first, i.e., the 0 which is the pairwise measure of a classifier with itself and the measure that corresponds to the other classifier. Otherwise, the positions of the two classifiers in the list become factors; the zeros are at different positions. Two copies of the same classifier should have a similarity measure of 1 because their diversity vectors are equivalent. A variant of this would move the two values to the front of the list or move only the non-zero value and eliminating the zero. Exactly what might be useful needs to be researched, but the essential point is to ensure the diversity measures at a specific position within two vectors correspond to the same classifier.

So from the similarity vectors, we can see that classifiers One and Two are similar in that they have roughly the same difference to Three despite having a high pairwise diversity. Two and Three, and One and Three, however, are less similar because of the different difference they have to the third. If a majority voting rule is being used to calculate the overall ensemble

decision, classifiers One or Two can not be removed, but given an initial pool of four classifiers, this approach could possibly provide guidance on what classifier to remove based on a ranking of the similarities. Or Three might be the candidate to remove as it is much more similar to One and Two than they are to each other. Thus the decision Three makes is more likely to be the decision One or Two makes, but One and Two will disagree more often. Exactly what to do depends on the overall composition of the ensemble and how different subsets of classifiers within the ensemble agree and disagree; further research is needed to elucidate this.

A concise description of the algorithm would be as follows:

- Classifiers $c_1 ... c_k$
- Classifier ordering scheme CS
- Diversity measure D
- Similarity function S
- Vector transformation function T

1. Order the classifiers using CS
2. For all i in 1 ... k, calculate $D_{ij}(c_i, c_j)$ for all j in 1 ... k
3. For all i in 1... k,
    - For all j in 1 ... k,
        - $V_i = T(\{D_{i1}, D_{i2} ... D_{ik}\})$
        - $V_j = T(\{D_{j1}, D_{j2} ... D_{jk}\})$
        - calculate $S_{ij}(V_i, V_j)$

8.3 The Dynamics Between Accuracy And Diversity

Besides being a possible technique by which classifier pools can be winnowed, similarity vectors may also have the ability to provide an explanation or insight into how ensemble accuracy varies as classifiers are added (or removed). An example is based on an analysis of the line plots in Figure 8.1. Tables 8.3 to 8.6 present a portion of the matrix of similarity vectors for the By Accuracy scheme, up to the ensemble of size 7 and the specific seed of

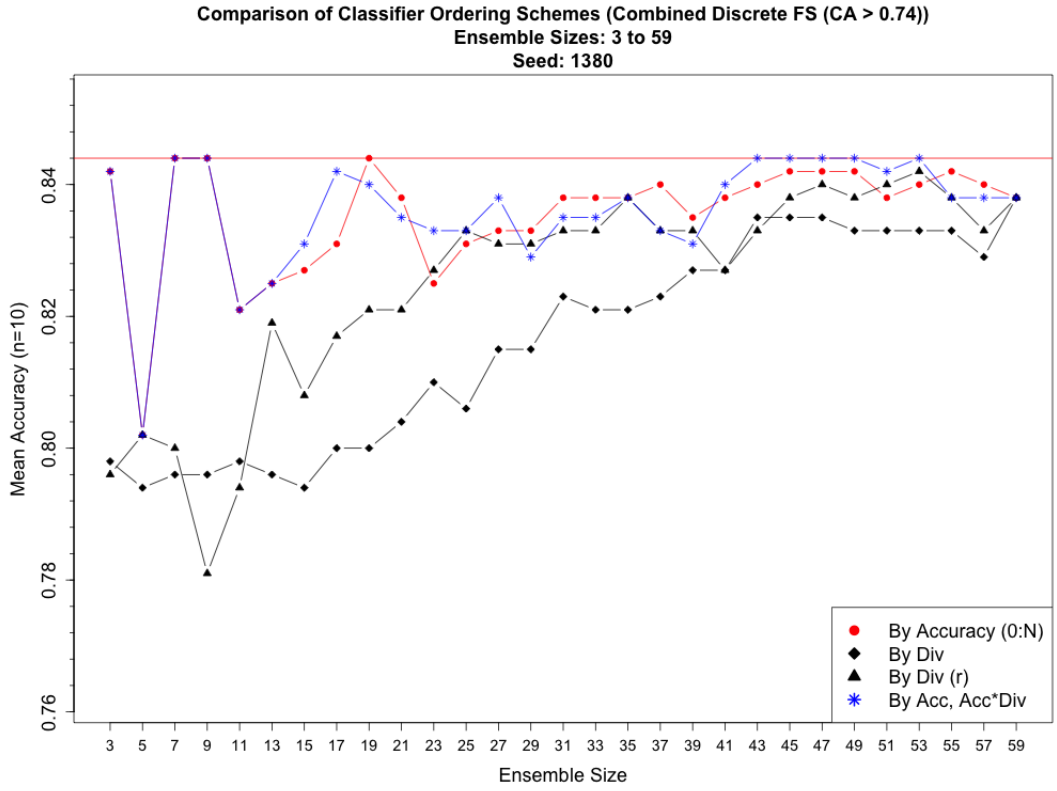

Figure 8.1: Comparison Of Schemes (Combined Discrete Feature Sets)

1380. This is one of the 10 runs that were averaged to create the graph in Figure 7.11; all of the line plots for the 10 seeds show this downward spike to some extent.

As seen in Figure 8.1, the By Accuracy scheme (and the By Acc, Acc*Div one as they overlap) has a noticeable spike in ensemble accuracy at the beginning, (about 4%). Examining how the similarity vectors change provides some insight into what is happening. Table 8.3 is a list of the classifiers used and how they were ranked. The similarity vectors are displayed in Tables 8.4 to 8.6; the pairwise diversity vectors are shown as well only to display how they change between different Ns and thus the similarity vectors do.

Based on Table 8.4, it is easy to see how all the classifiers are fairly well in agreement, but just enough diversity to increase the ensemble's accuracy from that of the highest classifier (0.7894) to just over 0.84. If one classifier decides incorrectly, the other two likely override it. But when two more classifiers are added (4 and 5, Table 8.5), the ensemble in effect starts dividing into voting blocks or factions. Examining the similarity vector for classifier 3 in Table 8.5 reveals the first two classifiers are now less similar to the other three. The two most accurate classifiers are now getting overridden presumably (determining how the classifiers voted for each test sample would provide evidence for this). Thus the ensemble accuracy drops drop down to only just above 80%. There is still some benefit to the ensemble, but worse decisions are being made.

Table 8.3: The First 7 Classifiers

Number	Classifier	Feature Set	Accuracy
1	SVC-SVM	lexicon-posbigram-tagbigram	0.7894
2	Logistic Regression	lexicon-pos-tagbigram	0.7852
3	MultiBayes	lexicon-tag-posbigram	0.7846
4	MultiBayes	lexicon-posbigram-tagbigram	0.7841
5	MultiBayes	lexicon-pos-posbigram-tagbigram	0.7829
6	Logistic Regression	lexicon-pos-tag-tagbigram	0.7818
7	Logistic Regression	lexicon-pos-posbigram-tagbigram	0.7810

Table 8.4: Diversity and Similarity Vectors (N=3)

Classifier	Vector		
1	0	0.181	0.148
2	0.181	0	0.217
3	0.148	0.217	0
1	1	0.982	0.996
2	1.0	1	0.995
3	1.0	1.0	1

Table 8.5: Diversity and Similarity Vectors (N=5)

Classifier	Vector				
1	0	0.181	0.148	0.142	0.148
2	0.181	0	0.217	0.202	0.200
3	0.148	0.217	0	0.040	0.042
4	0.142	0.202	0.040	0	0.015
5	0.148	0.200	0.042	0.015	0
1	1	0.988	0.873	0.831	0.845
2	0.999	1	0.837	0.777	0.778
3	0.833	0.806	1	0.995	0.995
4	0.781	0.727	0.995	1	1.0
5	0.794	0.723	0.995	1.0	1

However, when classifiers 6 and 7 are added, ensemble accuracy rises again and quite dramatically. Reading Table 8.6, it seems classifiers 1, 3, 4, and 5 now have formed the dominant faction, and because they are the highest (except for 2), overall accuracy increases again. 2, 6, and 7 seem to have formed a smaller faction based on their similarity vectors and what values are above 0.9. Exactly how to read these tables is unclear, as well as what similarity values are significant and what are not. But in general, the view that factions

compete within an ensemble and alliances change as new members are added is intuitive[1]. Thus this provides some possible insight into how classifier accuracy and the inter-diversity interact to influence the ensemble accuracy.

Table 8.6: Similarity Vectors (N=7)

Classifier	Vector						
1	1	0.822	0.923	0.901	0.909	0.826	0.859
2	0.777	1	0.629	0.586	0.591	1.0	0.995
3	0.911	0.479	1	0.998	0.998	0.634	0.662
4	0.886	0.436	0.998	1	1.0	0.59	0.635
5	0.894	0.447	0.998	1.0	1	0.596	0.64
6	0.782	1.0	0.485	0.442	0.453	1	0.996
7	0.824	0.995	0.538	0.502	0.512	0.996	1

8.4 Conclusions

This research into winnowing is only preliminary, so therefore a conclusive answer to Question 6 posed in Chapter 3 is not possible. There are many variables to examine (the strengths and limitations of using different diversity measures, the value of different similarity or distance functions, and the utility of different vector transformations). Multiple diverse data sets also must be analyzed to gauge how useful these concepts of diversity and similarity vectors are. Another important question is how do the overall ensemble diversity measures relate to these topics? But it is clear there looks to be some value in defining groups of classifiers as a unit (using a particular method) and that one unit can be compared to another to gain possible insights. Besides pairwise diversity and overall ensemble diversity, diversity between a classifier and a set of classifiers (treated as a unit) looks to be useful. The concept of different factions of classifiers (based on their similarities) that override one another based on the input data, along with the idea that factions can change, is a useful metaphor for describing ensembles.

Besides providing insight into how an initial pool of classifiers could be reduced to an essential set based on their diversity or similarity vectors, these techniques also could be a way to analyze the internals of an ensemble. The similarity vectors of ensembles created using ensemble methods could be analyzed, presumably providing an explanation as to why sometimes an ensemble method results in no performance improvement. Also, exactly how an ensemble's accuracy changes as its size (or composition) varies along with the similarity

[1]Any number of quotes from HBO's Games Of Thrones would go here

vectors is an obvious question as well as if there is an ideal set of similarity vectors or characteristics of them. An end-to-end analysis of how individual test data points result in different factions making the ultimate decision could be useful in pruning ensembles to an optimal size or figuring out which classifiers need replaced to improve performance.

These preliminary results also shed light on how the ideal amount of diversity lies within a range, i.e., not too much or too little. Exactly what this range is, how to determine it, and how it might vary for different types of datasets and classifiers (and even the type of voting rules) are open questions. A thorough investigation will aid in the better understanding of the relationship between classifier accuracy, diversity, and ensemble accuracy.

Chapter 9

Discussion

9.1 Overview

This chapter discusses the research organized by a particular aspect: the data, the methodology, the experimental process, and the results. Various issues related to each aspect are examined; this includes a discussion of how they might influence the conclusions of this study.

9.2 Data

An important thing to note about the dataset used (the Ott dataset, from combining those in Ott et al. (2011, 2013)) is that it is not entirely representative of real-world reviews, as previously mentioned in Chapter 2. The larger difference in the word distribution for fake and authentic reviews creates a situation where n-gram based features make it very easy to distinguish between the two classes. This accounts for the numerous studies that use this dataset and n-gram features which achieve accuracies around 90%. During this research, a hypothesis was devised that could possibly explain the reason for this larger difference. The Ott dataset was generated in part by hiring workers through the Amazon Mechanical Turk service. Thus reviews generated by these people were known to be fake. There were requirements imposed on the AMT workers, e.g., they only had 30 minutes to write a review and they were to pretend they worked for the hotel's marketing department, among other basic requirements. The fundamental issue here is that any research that involves humans who generate data for an experiment must take into account psychological, sociological, and economic type factors; in a sense, the research is an experiment in psychology as much as the main topic.

The generation of the Ott dataset apparently did not compensate well enough for some important factors. AMT workers are typically paid a flat fee, not by the hour; the fee offered

was one dollar. Therefore it was in their best interest to complete the work as quickly as possible; 12% of the reviews were ostensibly written in under one minute, although Ott et al have a hypothesis that explains this. From a writing standpoint, a very low fee and a time limit is a recipe for lazy cliched writing that relies upon the first things that come to mind. For a hotel review, that would be rude staff, dirty rooms, small beds, and other such stereotypical topics associated with hotels. Thus the fake reviews that were generated most likely have a smaller variance in the words used compared to the actual reviews from TripAdvisor that were labeled as authentic. Cliched writing all sounds the same. This would account for the larger difference in word distribution. An topic model based analysis of the Ott dataset could provide some insight into this idea; if the known fake reviews discuss a smaller set of topics than real-world reviews, then this hypothesis about the need to better control psychosocial factors and economic incentives should be kept in mind when creating new datasets from scratch. There also was no guarantee these fake reviewers had any training in marketing, which could be considered a positive. But it also could be considered a negative in that non-marketers theoretically would not write as well or convincingly; other studies have focused on using actual marketers to generate reviews which makes more immediate sense if one is trying to generate realistic fake reviews under controlled conditions.

This hypothesis is supported by the results of Banerjee and Chua (2014c) who conduct sociological studies of actual reviewers to analyze their process. Reading their paper reveals spam reviewers put a lot of effort into crafting reviews that are as realistic as possible. They read other reviews, model their structure, plagiarize sentences from them, and so on; the focus on 'marketing' is evident. Expecting an AMT worker to go to this much effort for a token amount is unrealistic. It is for this reason the Ott dataset should not be considered a 'gold-standard' as the authors claim; the subsequent analysis of this dataset by others bears this out. Li et al. (2014) address these concerns in part by enhancing the base Ott dataset with reviews written by marketers and subject matter experts. The fact the base Ott dataset is present despite its known limitations does not invalidate the results, but it does imply there are caveats to any conclusions.

9.3 Methodology

The caveats associated with the Ott dataset are the reasons this study focused on not using bag-of-words or any vocabulary based features. Instead, the focus was on determining if there was any utility to using multiple feature sets based on more abstract analyses of the

text. Also investigated was how the classifiers might be ensembled. This does not preclude unknown statistical biases, such as the demonstrated word distribution issue, from influencing the results; more research is needed to compare the feature sets from real-world reviews to the Ott dataset's. But in general, the questions posed in Chapter 3 were answered satisfactorily: customized ensemblement of individual classifiers using a majority voting rule was shown to have promise. Once the accuracy and diversity relationship is better understood, this will lead to efficient creation of effective ensembles. The results were positive enough that it is likely this set of techniques and feature sets would be applicable to classification of real-world reviews. The Ott dataset is useful in that it is a known quantity and is well understood, so therefore new techniques and different methodologies can be tested used it. But results from using it should not be declared as definitive proof.

Another point to make about the methodology is that the polarity of a review is certainly a factor in how well reviews can be classified. There is enough of a difference between fake (or authentic) positive and negative reviews that implies classifiers are somewhat confused sometimes, e.g., there is possibly two distinct subsets of feature data corresponding to each type and the classifiers can not unify things. It could account for the leveling off of accuracy of individual classifiers at about 80%. In this study, the review polarity was simply used as an additional feature in the feature set to see if that approach was sufficient. The results when using a feature set where polarity was not present were slightly less accurate. Follow-on research should address this aspect in a more sophisticated way. One reason would be to confirm this idea that some type of XOR situation is present, and two, to see how it could be handled in a better manner. The training process used in this study randomly selected reviews and did not consider their polarity, so undoubtedly there was a class imbalance in that respect in the training data. This approach is more congruent to how a real world classification system might deal with data. In contrast, Ott et al. (2013) trained classifiers only on positive reviews, only on negative ones, and both in three phases. For test data, they used only the positive or negative reviews and cross-validation. Thus class (polarity) imbalance was not an issue, which could account for the higher accuracies achieved (as well as the fact unigrams and bigrams were used as features). But their results also support the conclusion polarity is an important factor.

Some other methodological improvements that could improve results would be using PCA or linear discriminant analysis to guide the processes of feature selection and feature extraction (or creation). Dimensionality reduction, especially when multiple feature sets are combined, would aid in reducing correlated features and possibly improve performance. As

for feature creation, the exploratory data analysis showed very clearly how fake and authentic reviews overlapped in terms of the distribution of values for many features, so improving the separability would only improve classifier performance. More analysis of the individual features would be beneficial. Hyperparameter optimization was also not fully explored and may add some benefit if building a system for production.

9.4 Experiments

An important distinction between this study and others is that k-fold cross-validation was not used, but Monte Carlo. As mentioned, it was decided that 10 fold cross validation would not be sufficient as that would leave only 160 samples to serve as a test set; overfitting was a concern. 5 fold would leave 320 samples for testing, but using only 5 folds might not be statistically sufficient. The question of what might result from a Monte Carlo approach was also of interest. Results were generally comparable to other studies; this is examined in the next section. But one important result seen was that particular random seeds seemed to be always associated with the better (or worse) results (in terms of classifier accuracies). This assumption needs to be supported by a detailed analysis of the data, but if true, it might provide some insight into hidden correlations within the data. For instance, the right random seed might have resulted in a set of reviews being used for training data that were all distributed in the right sort of fashion along a particular set of individual features. This then lead to the test data being more accurately classified. If these relationships could be determined, it might lead to insights into how to best create new features that are more discriminative.

Also, as always, more time and more computer cores would have been useful because they would have allowed for more experiments. Experiments were carried out in a thorough fashion by subdividing the set of feature sets by the type of the data (continuous, discrete, and then a mix), unlike other studies. This comprehensive approach provided a better understanding of classification results and the value of feature sets relative to each other. It was only near the end when the need for more time was apparent. Exactly how important classifier diversity is and the impact of the best classifiers having insufficient inter-diversity was not obvious at the start of this study. Improving ensemble accuracy into the mid or upper 80% range looks possible, but it is going to require more complex ensembles, perhaps ones that involve using ensembles of ensembles to gain benefit from sufficiently diverse classifiers. Investigating other feature sets like topic models or linguistic frames is also another approach to increasing diversity.

9.5 Results

The most obvious topic to discuss about the results is the poor performance of classifiers when using the sentiment analysis based feature set. This study blindly used the output of several sentiment analysis tools which were shown to be fairly incorrect in detecting negative sentiment. A more sophisticated way (through feature creation or a finer grained analysis) or using better tools that can handle the negative polarity reviews better would surely improve results. The initial hypothesis that excessive sentiment would be indicative of inauthentic reviews did not hold up, and as the results show, adding the sentiment based feature set to others tended to depress the accuracy. Peng and Zhong (2014) is a good example of a more sophisticated way to address sentiment; they calculate a score based on the individual sentiment associated with specific features of the product being reviewed and those features are weighted. For example, a phone may be expensive, so the sentiment and weight attached to the price are important factors, but if the phone's other features are all positively scored, the overall sentiment score is high. Thus the accuracy of around 84% is not unexpected; the official or true sentiment class was based on the overall review score of 1 to 5 stars. The sentiment scores are then used in a time series analysis to detect anomalies in the ratings and thus what reviews are most likely spam; sentiment is not directly used to classify reviews.

Another important consideration is that this study, as far as the literature review revealed, is the second one to use the Empath software in place of the more commonly used LIWC. Ott et al. (2011) report an accuracy of 76.8% when using a LIWC based feature set with a SVM; our results of 73.4% and a SVM are comparable, especially considering the SVM configurations were different as well as the polarities of the reviews in each dataset. Unfortunately, Ott et al. (2013) which also used the LIWC does not report results broken down by the feature set used, but only along polarities. Li et al. (2014) do, and their results associated with the LIWC range from 72% to 76%; the complexity of their methodology prevents a more refined comparison. But overall, it can be concluded the Empath software is an acceptable substitute for the LIWC and the extensibility of the Empath software is a potential benefit; it could be customized appropriately depending on the exact type or subject under review or other review characteristics.

As for a comparison to the few previous studies that investigated ensembles and review classification, two of them used n-grams as the features. So their results can not be compared because, as discussed, the Ott dataset has a distinct difference in the distribution of words between fakes and authentic reviews. Thus studies that use n-grams tend to report greater accuracies; also, only ensemble methods were used, not custom ensemblement. Banerjee

et al. (2015) however uses a combination of different feature sets more akin to this research. Besides stylometric, POS, and readability, the last set can be labeled as word 'categories', i.e., a lexical feature set. Some examples would be past tense words, casual words, and motion related words; the LIWC was used to acquire this type of data. Another difference is that review titles were also processed, unlike in this study.

The most important consideration however is that Banerjee et al. (2015) used their own dataset of authentic and human generated fake reviews, not the Ott dataset. So only a rough comparison of results can be made. The accuracies for single classifiers, all using the same feature set, ranged from 64% to 71% and the voting ensemble of all nine classifiers improved the accuracy to 74%. So these results are in line with the results of this study, showing hybrid ensembles can improve performance. Performing this study again using the Banerjee dataset would be interesting; it is an open question if their dataset has the problems with word distribution that the Ott dataset does. Another aspect of the Banerjee dataset is that it included reviews of moderate polarity (neither wholly negative or positive), so that was certainly a complicating factor that influenced results.

Chapter 10

Conclusion

10.1 Main Findings

The economic impact of fake reviews is a significant problem as well as the ever-increasing extent of it. Improving the classification accuracy of machine learning systems is crucial if this issue is going to be effectively addressed. This research investigated how feature level and decision level fusion could improve the classification of fake reviews by addressing five specific questions.

Question 1 involved establishing a baseline for performance when using a single classifier and a single feature set. The best accuracies per each feature set ranged from 0.610 to 0.749, while the AUC scores ranged from 0.673 to 0.827. Logistic Regression classifiers in general were the most accurate with SVM a close second, although more variable depending on the feature set. Feature sets with discrete individual features, such as POS, were associated with higher accuracies than continuous feature sets like readability measures; this was attributed to the discrete feature set data showing more of a distinction between fake and authentic reviews while the distribution of values (fake and authentic) for the continuous feature sets greatly overlapped.

Ensemble methods, the subject of Question 2, did not improve the accuracy of these base classifiers at all except in the case of Bagging and Decision Trees. An analysis of how the ensemble methods work, and what would result given the specific characteristics of this particular dataset, provided an explanation. One key factor was the only slight difference in the distribution of some features' values between fake and authentic reviews. The way Bagging leverages the variance of multiple Decision Trees, resulting in better accuracy, overcomes this problem that other types of classifiers can not. The other significant factor was the polarity of a review (negative or positive); there is sometimes a notable difference between the two review types in terms of feature value distributions. Thus classifiers that were accurate for positive reviews (authentic or fake) failed more often when faced with a

negative review and AdaBoost, because of how it works, created ensembles that were more likely to misclassify reviews that were confusing because of the polarity and the composition of the training dataset.

Question 3 addressed whether combining feature sets into one was beneficial when using a single classifier. It did result in a modest improvement compared to the best individual feature set and again Logistic Regression was the best classifier in general. Combining only discrete feature sets with each other resulted in greater overall accuracy compared to combining just continuous feature sets, but the lower accuracies associated with each of the continuous feature sets was certainly a factor. Combining both types of feature sets together in different combinations showed some improvement as well. But in the interest of efficiency, using only combinations of a few discrete feature sets is sufficient given the effort involved in more complex arrangements.

As for using ensemble methods with a single classifier and combined feature sets (Question 4), once again only Bagging and Decision Trees resulted in any improvement in performance. Combining feature sets into one essentially did not introduce any variance into the decisions made by the members of the ensemble. Thus the ability to override an initially incorrect decision made by the base classifier, which ensemble methods engender, was not present. Using Bagging with Decision Trees does engender this ability as previous explained in the answer to Question 2.

The final question addressed whether hybrid ensembles, composed of individual classifiers from Question 1 and 3, would improve performance. A methodical investigation revealed how ensemblement using a simple majority voting rule (and selecting the right classifiers) could result in reasonably large gains in overall accuracy. However, the problem of what classifiers to use quickly became an issue. This is due to the size of the potential classifier pool growing as classifiers became more complex (in terms of the feature sets used) while the same general level of accuracy is maintained. Schemes for ordering and selecting classifiers based on a variety of parameters were then devised. This allowed the entire set of possible ensembles to be sampled, alleviating the need to construct each one for evaluation.

Classifier selection schemes show some promise as a technique, but the problem of diversity became even more important as the classifier pool kept increasing and the top classifiers less diverse. A sixth question was then formulated. To address this question, the concepts of diversity and similarity vectors were created. Based on evaluating the similarity of two classifiers' diversity vectors (vectors of the pairwise diversity between a classifier and the others), these vectors allow for the evaluation of how one classifier relates to the others

as a whole; classifiers that are comparable in accuracy and have close enough similarity or diversity to the others can logically be eliminated from the pool. Analysis also revealed how similarity vectors might be used to understand the changes in an ensemble's accuracy as classifiers are added to it, as well as understand exactly how classifiers within an ensemble cooperate. This was preliminary research and there are many parameters to investigate before determining the utility of this technique, but it shows promise in furthering the understanding of how individual classifier accuracy, ensemble accuracy, and diversity are interrelated.

10.2 Suggestions for Further Work

It is a truism that answering one question only creates more questions. The following list is extensive, so the major ideas are stately as briefly as possible; minor ones such as investigating hyperparameter optimization are not.

- How does adding more feature sets (topic models, linguistic frames, or grammar based features), or improving the existing ones, aid in improving performance?
- Would using a classic Mixture Of Experts model, with a gating network, improve performance over using a voting rule? How would feature sets clash?
- Does the methodology used in this research work with real world review datasets, not just the somewhat artificial Ott dataset?
- Does treating this as a four class problem improve results, in that negative and positive fake reviews are better distinguished and the results combined?
- If viewed as a four class problem, can ensembles be created that, in effect, act as a multiplexor (e.g., 3 classes, authentic, fake positive, and fake negative reviews)?
- Would a two stage process of classifying by polarity, then by authenticity, improve results?
- Would ensembling the Naive Bayes classifiers with high sensitivity and low specificity, in a ECOC way, work?
- Does an analysis of the similarity or diversity vectors of ensembles created through ensemble methods reflect their observed behavior?
- How can using pool winnowing and similarity vectors improve the selection of more effective classifiers?

- Would clustering of classifiers (based on their diversity vectors) in order to find a single representative one aid in winnowing the pool?
- Are different patterns revealed if the first 10,000 ensemble accuracies (or overall diversities) are graphed using different pool ordering schemes?
- What improvement might result from ensembling the different types of complex classifiers together and also with ensembles of the simpler classifiers?

Bibliography

Ahsan, M. I., Nahian, T., Kafi, A. A., Hossain, M. I., and Shah, F. M. (2016). An Ensemble Approach to detect Review Spam using hybrid Machine Learning Technique. In *Computer and Information Technology (ICCIT), 2016 19th International Conference on*, pages 388–394. IEEE.

Alyahyan, S. and Wang, W. (2017). Feature Level Ensemble Method for Multi-media Dataset. Unpublished.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *LREC*, volume 10, pages 2200–2204.

Ballenger, B. (2011). 11 Tips to Avoid Fake Reviews. http://moneytalksnews.com/11-tips-to-avoid-fake-reviews/, Accessed: 2017-03-20.

Bambauer-Sachse, S. and Mangold, S. (2013). Do consumers still believe what is said in online product reviews? A persuasion knowledge approach. *Journal of Retailing and Consumer Services*, 20(4):373–381.

Banerjee, S. and Chua, A. Y. (2014a). Applauses in Hotel Reviews: Genuine or Deceptive? In *Science and Information Conference (SAI), 2014*, pages 938–942. IEEE.

Banerjee, S. and Chua, A. Y. (2014b). A Study of Manipulative and Authentic Negative Reviews. In *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, page 76. ACM.

Banerjee, S. and Chua, A. Y. (2014c). Understanding the Process of Writing Fake Online Reviews. In *Digital Information Management (ICDIM), 2014 Ninth International Conference on*, pages 68–73. IEEE.

Banerjee, S., Chua, A. Y., and Kim, J.-J. (2015). Using Supervised Learning to Classify Authentic and Fake Online Reviews. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*, page 88. ACM.

Burgoon, J. K., Blair, J., Qin, T., and Nunamaker, J. F. (2003). Detecting Deception through Linguistic Analysis. In *International Conference on Intelligence and Security Informatics*, pages 91–101. Springer.

Chen, R. Y., Guo, J. Y., and Deng, X. L. (2014). Detecting Fake Reviews of Hype About Restaurants by Sentiment Analysis. In *International Conference on Web-Age Information Management*, pages 22–30. Springer.

Christopher, S. L. and Rahulnath, H. (2016). Review authenticity verification using supervised learning and reviewer personality traits. In *Emerging Technological Trends (ICETT), International Conference on*, pages 1–7. IEEE.

Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., and Al Najada, H. (2015). Survey of review spam detection using machine learning techniques. *Journal of Big Data*, 2(1):23.

Davis, J. and Goadrich, M. (2006). The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM.

Ezpeleta, E., Zurutuza, U., and Hidalgo, J. M. G. (2016). Using Personality Recognition Techniques to Improve Bayesian Spam Filtering. *Procesamiento del Lenguaje Natural*, 57:125–132.

Fast, E., Chen, B., and Bernstein, M. S. (2016). Empath: Understanding Topic Signals in Large-Scale Text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4647–4657. ACM.

Feldman, R. (2013). Techniques and Applications for Sentiment Analysis. *Communications of the ACM*, 56(4):82–89.

Feng, S., Banerjee, R., and Choi, Y. (2012). Syntactic Stylometry for Deception Detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 171–175. Association for Computational Linguistics.

Fogg, B. and Tseng, H. (1999). The elements of computer credibility. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 80–87. ACM.

Gara, T. (2013). A Tough Day for Local Search: Fake Reviews, and a Bad Review. https://blogs.wsj.com/corporate-intelligence/2013/09/23/a-tough-day-for-local-search-fake-reviews-and-a-bad-review/, Accessed: 2017-03-15.

Gilbert, C. H. E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International AAAI Conference on Weblogs and Social Media*.

Goldberg, L. R. (1990). An Alternative Description of Personality: The Big-Five Factor Structure. *Journal of Personality and Social Psychology*, 59(6):1216.

Gonçalves, P., Dalip, D. H., Costa, H., Gonçalves, M. A., and Benevenuto, F. (2016). On the Combination of "Off-The-Shelf" Sentiment Analysis Methods. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 1158–1165. ACM.

Heredia, B., Khoshgoftaar, T. M., Prusa, J., and Crawford, M. (2016). An Investigation of Ensemble Techniques for Detection of Spam Reviews. In *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*, pages 127–133. IEEE.

Heydari, A., ali Tavakoli, M., Salim, N., and Heydari, Z. (2015). Detection of review spam: A survey. *Expert Systems with Applications*, 42(7):3634–3642.

Hutto, C. J. and Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *Eighth international AAAI conference on weblogs and social media*.

Janan, D. and Wray, D. (2012). Readability: The limitations of an approach through formulae. Paper presented at the British Educational Research Association Annual Conference.

Jensen, M. L., Averbeck, J. M., Zhang, Z., and Wright, K. B. (2013). Credibility of Anonymous Online Product Reviews: A Language Expectancy Perspective. *Journal of Management Information Systems*, 30(1):293–324.

Jindal, N. and Liu, B. (2007). Analyzing and Detecting Review Spam. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 547–552. IEEE.

Jindal, N. and Liu, B. (2008). Opinion Spam and Analysis. In *Proceedings of the First ACM International Conference on Web Search and Data Mining*, pages 219–230. ACM.

Kamerer, D. (2014). Understanding the yelp review filter: An exploratory study. *First Monday*, 19(9).

King, R. A., Racherla, P., and Bush, V. D. (2014). What We Know and Don't Know About Online Word-of-Mouth: A Review and Synthesis of the Literature. *Journal of Interactive Marketing*, 28(3):167–183.

Koven, J., Siadati, H., and Lin, C.-Y. (2014). Finding Valuable Yelp Comments by Personality, Content, Geo, and Anomaly Analysis. In *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*, pages 1215–1218. IEEE.

Lau, R. Y., Liao, S. S., and Xu, K. (2010). An empirical study of online consumer review spam: A design science approach. In *ICIS*, volume 2010, pages 103–123.

Li, J., Ott, M., Cardie, C., and Hovy, E. H. (2014). Towards a General Rule for Identifying Deceptive Opinion Spam. In *ACL (1)*, pages 1566–1576.

Ma, Y. and Li, F. (2012). Detecting Review Spam: Challenges and Opportunities. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, pages 651–654. IEEE.

Mairesse, F., Walker, M. A., Mehl, M. R., and Moore, R. K. (2007). Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text. *Journal of artificial intelligence research*, 30:457–500.

McGill, R., Tukey, J. W., and Larsen, W. A. (1978). Variations of box plots. *The American Statistician*, 32(1):12–16.

Morran, C. (2016). Is Amazon Doing Anything To Fight Latest Wave Of Fake, Paid-For Reviews? https://consumerist.com/2016/02/08/is-amazon-doing-anything-to-fight-latest-wave-of-fake-compensated-reviews/, Accessed: 2017-03-5.

Mukherjee, A. and Venkataraman, V. (2014). Opinion Spam Detection: An Unsupervised Approach Using Generative Models. *Techincal Report, UH*.

Mukherjee, A., Venkataraman, V., Liu, B., and Glance, N. (2013a). Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews. *Technical Report UIC-CS-2013–03, University of Illinois at Chicago, Tech. Rep.*

Mukherjee, A., Venkataraman, V., Liu, B., and Glance, N. S. (2013b). What Yelp Fake Review Filter Might Be Doing? In *ICWSM*.

Northrup, L. (2016). Amazon Sues 1,114 Individual Reviewers For Hire. https://consumerist.com/2015/10/16/amazon-sues-1114-individual-reviewers-for-hire/, Accessed: 2017-03-28.

Olson, R. S., Bartley, N., Urbanowicz, R. J., and Moore, J. H. (2016). Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. In *Proceedings of the*

Genetic and Evolutionary Computation Conference 2016, GECCO '16, pages 485–492, New York, NY, USA. ACM.

Ong, T., Mannino, M., and Gregg, D. (2014). Linguistic characteristics of shill reviews. *Electronic Commerce Research and Applications*, 13(2):69–78.

Ott, M., Cardie, C., and Hancock, J. (2012). Estimating the Prevalence of Deception in Online Review Communities. In *Proceedings of the 21st international conference on World Wide Web*, pages 201–210. ACM.

Ott, M., Cardie, C., and Hancock, J. T. (2013). Negative Deceptive Opinion Spam. In *HLT-NAACL*, pages 497–501.

Ott, M., Choi, Y., Cardie, C., and Hancock, J. T. (2011). Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319. Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peng, Q. and Zhong, M. (2014). Detecting Spam Review through Sentiment Analysis. *JSW*, 9(8):2065–2072.

Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). Linguistic Inquiry and Word Count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.

Petrov, S. and Klein, D. (2007). Improved Inference for Unlexicalized Parsing. In *HLT-NAACL*, volume 7, pages 404–411.

Popken, B. (2010). 30 Ways You Can Spot Fake Online Reviews. https://consumerist.com/2010/04/14/how-you-spot-fake-online-reviews/, Accessed: 2017-03-20.

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., and Stein, B. (2017). A Stylometric Inquiry into Hyperpartisan and Fake News. *arXiv preprint arXiv:1702.05638*.

Ramyaa, C. H. and Rasheed, K. (2004). Using Machine Learning Techniques for Stylometry. In *Proceedings of International Conference on Machine Learning*.

ReviewMeta (2016a). Analysis of 7 million Amazon reviews: customers who receive free or discounted item much more likely to write positive review. https://reviewmeta.com/blog/analysis-of-7-million-amazon-reviews-customers-who-receive-free-or-discounted-item-much-more-likely-to-write-positive-review/, Accessed: 2017-03-28.

ReviewMeta (2016b). The Fundamental Problem with Review Clubs. https://reviewmeta.com/blog/the-fundamental-problem-with-review-clubs/, Accessed: 2017-03-28.

Ribeiro, F. N., Araújo, M., Gonçalves, P., Gonçalves, M. A., and Benevenuto, F. (2016). SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):1–29.

Saito, T. and Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PloS one*, 10(3):e0118432.

Segal, D. (2011). A Rave, a Pan, or Just a Fake? http://www.nytimes.com /2011/05/22/your-money/22haggler.html, Accessed: 2017-03-25.

Severance, C. (2016). Couple Fights Off $ 1 Million Lawsuit Over Bad Yelp Review. http://dfw.cbslocal.com/2016/05/04/yelp-review-could-cost-couple-1-million/, Accessed: 2017-03-23.

Shojaee, S., Murad, M. A. A., Azman, A. B., Sharef, N. M., and Nadali, S. (2013). Detecting Deceptive Reviews Using Lexical and Syntactic Features. In *Intelligent Systems Design and Applications (ISDA), 2013 13th International Conference on*, pages 53–58. IEEE.

Shrestha, P., Mukherjee, A., and Solorio, T. (2016). Large Scale Authorship Attribution of Online Reviews. http://www2.cs.uh.edu/~arjun/papers_new/Shrestha et al. CICLING 16.pdf.

Smedt, T. D. and Daelemans, W. (2012). Pattern for Python. *Journal of Machine Learning Research*, 13(Jun):2063–2067.

Tausczik, Y. R. and Pennebaker, J. W. (2010). The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, 29(1):24–54.

Vasquez, C. (2014). *The Discourse of Online Consumer Reviews*. Bloomsbury Publishing.

Villaroel Ordenes, F., Ludwig, S., Grewal, D., de Ruyter, K., and Wetzels, M. (2016). Analyzing Online Reviews Through the Lens of Speech Act Theory: Implications for Consumer Sentiment Analysis. *Journal of Consumer Research*.

Xia, R., Zong, C., and Li, S. (2011). Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences*, 181(6):1138–1152.

Xu, Q. and Zhao, H. (2012). Using Deep Linguistic Features for Finding Deceptive Opinion Spam. In *COLING (Posters)*, pages 1341–1350.

Yoo, K.-H. and Gretzel, U. (2009). Comparison of Deceptive and Truthful Travel Reviews. *Information and communication technologies in tourism 2009*, pages 37–47.

Zheng, R., Li, J., Chen, H., and Huang, Z. (2006). A Framework for Authorship Identification of Online Messages: Writing-Style Features and Classification Techniques. *Journal of the Association for Information Science and Technology*, 57(3):378–393.