Spring 5-22-2021

Sentiment Classification Bias In User Generated Content

Alpana Deshpande
Syracuse University

Abstract

Interactive websites generate terabytes of data on a daily basis. This data can be used in multiple analytical applications to teach computers more about human behavior. Text classification is such an application. Multiple freely available user-generated text data can be used to teach computers to identify the sentiments behind a user's on-screen interactions without the need of any human intervention. Sentiment analysis is an interesting problem, solving which would theoretically get a computer closer to passing the Turing test. Through this thesis, we test the ability of a classifier to accurately identify user sentiments. However, we do not focus on standard classification settings and the aim is to train the classifier in such a way that it would also be effective in identifying sentiment behind user generated text generated from a completely new social media platform. To be able to do this, we must first identify behavioral *bias* based on user interactions in two different social media sites as well as websites that accept user reviews. This bias must then be mitigated in order to obtain an unbiased classifier that can then be used to identify user sentiments on any social media platform. For the research in this thesis, such user-generated text is obtained from the social media sites Reddit and Twitter. We also obtain product review data related to both books and wine. Various natural language processing techniques are then employed to process the data and extract similar and dissimilar trends. Vectorized user text would be used to train sentiment classifiers. Finally, classification bias would be identified and mitigated in order to obtain classifiers that can identify human sentiments in real-time with an improved accuracy with limited dependency on source information.

SENTIMENT CLASSIFICATION BIAS IN USER GENERATED CONTENT

by

Alpana Deshpande

B.E., University of Mumbai, 2016

Thesis
Submitted in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science

Syracuse University
May 2021

Contents

Abstract i

List of Figures viii

List of Tables ix

1 Introduction 1

1.1 Contributions 2

1.2 Road Map 3

2 Literature Review 4

2.1 Bias 4

2.2 Text Classification 5

2.3 Sentiment Analysis 6

2.3.1 Machine Learning-based Approaches 7

2.3.2 Other Approaches 9

2.3.3 Applications of Sentiment Classification 9

2.4 Parts of Speech 10

2.5 Cross-Domain Classification 12

3 Datasets 14

3.1 Datasets 14

3.1.1 Social Media Datasets 15

Reddit Dataset 15

Twitter Dataset 16

3.1.2 User Reviews Dataset 16
Wine Reviews Dataset 16
Book Reviews Dataset 17
3.2 Noise in Data 18
3.2.1 Unusable Features in Text 18
Stop words 18
Spelling mistakes 19
Special characters and numbers 19
Mixture of Languages 19
3.2.2 Clarity of Speech 20
Synonyms 20
Variations of words 20
Negations 20
Sarcasm 21
Vagueness 21
Hyperboles 21
3.2.3 Domain Specific Data Issues 21
Social Media Dataset 21
Reviews Dataset 22
3.3 Data Cleaning 22
Removing stop-words 23
Reducing variations in text 23
Removing numbers and special characters 23
Removing words from a different language 23
Fixing spelling mistakes 23
Adding parts of speech tags 24

4 Methodology and Results **25**
4.1 Sentiment Classification 25

4.1.1 Rule-Based . . . 26
4.1.2 Machine Learning-Based . . . 27
Decision Trees . . . 29
SVM . . . 29
Random Forest . . . 29
Naive Bayes . . . 30
XGBoost . . . 30
4.1.3 Hybrid Approach . . . 30
4.1.4 Comparison of Sentiment Analysis approaches . . . 31
Results on Social Media Datasets . . . 33
Results on Reviews Datasets . . . 33
4.2 Cross-Domain Classification . . . 33
4.2.1 Social Media . . . 34
4.2.2 Reviews . . . 36
4.3 Bias in Cross-Classification . . . 36
4.3.1 Analyzing drops in accuracy . . . 36
4.3.2 Experiments to Mitigate Bias . . . 38
Using subsets of features . . . 38
Amplifying relevant features . . . 44
Incorporating rule-based features . . . 44
Increasing the weightage of common words . . . 48
4.3.3 Evaluating and Comparing Bias Reduction methods . . . 50

5 Conclusions and Future Work **52**

5.1 Observations . . . 52
5.1.1 Data . . . 52
5.1.2 Sentiment Classification . . . 53
5.1.3 Cross-Domain Classification . . . 53
5.1.4 Bias-Reduction Methods . . . 54

5.2 Limitations 55
5.3 Applications 56
5.3.1 Detection of hate speech in new social media sites . . . 57
5.3.2 Detection of popularity in newly launched products . . 57
5.3.3 Comparing the behaviours of users 57
5.3.4 Applications beyond sentiment analysis 58

Bibliography **59**

List of Figures

3.1 Distribution of Reddit data 15
3.2 Distribution of Twitter data 16
3.3 Distribution of Wine Reviews data 17
3.4 Distribution of Book Reviews data 18

4.1 Comparison of Sentiment Analysis approaches 32
4.2 Cross-Domain Classification 34
4.3 Comparison of Bias Reduction approaches 51

List of Tables

4.1 Rule based sentiment classification: Social Media 27
4.2 Rule based sentiment classification: Reviews Dataset 27
4.3 Machine Learning Approach - Social Media Dataset 28
4.4 Machine Learning Approach - Reviews Dataset 28
4.5 Hybrid Approach - Social Media Dataset 31
4.6 Hybrid Approach - Reviews Dataset 31
4.7 Comparison of Sentiment Analysis - Social Media Dataset . . 32
4.8 Comparison of Sentiment Analysis - Reviews Dataset 32
4.9 Cross-Domain - Social Media Dataset 35
4.10 Cross-Domain Drop Rates - Social Media Dataset 35
4.11 Hybrid Cross-Domain - Social Media Dataset 35
4.12 Hybrid Cross-Domain Drop Rates - Social Media Dataset . . . 35
4.13 Cross-Domain - Reviews Dataset 36
4.14 Cross-Domain Drop Rates - Reviews Dataset 36
4.15 Hybrid Cross-Domain - Reviews Dataset 37
4.16 Hybrid Cross-Domain Drop Rates - Reviews Dataset 37
4.17 Social Media: Feature subset - Adjectives 39
4.18 Drop Rates of Social Media Feature subset - Adjectives 40
4.19 Social Media: Feature subset - Adverbs 40
4.20 Drop Rates of Social Media Feature subset - Adverbs 40
4.21 Social Media: Feature subset - Adjectives and Adverbs 40
4.22 Drop Rates of Social Media Feature subset - Adjectives and Adverbs 41

4.23 Social Media: Feature subset - Nouns and Verbs 41
4.24 Drop Rates of Social Media Feature subset - Nouns and Verbs 41
4.25 Reviews: Feature subset - Adjectives 41
4.26 Drop Rates of Reviews Feature subset - Adjectives 42
4.27 Reviews: Feature subset - Adverbs 42
4.28 Drop Rates of Reviews Feature subset - Adverbs 42
4.29 Reviews: Feature subset - Adjectives and Adverbs 42
4.30 Drop Rates of Reviews Feature subset - Adjectives and Adverbs 43
4.31 Reviews: Feature subset - Nouns and Verbs 43
4.32 Drop Rates of Reviews Feature subset - Nouns and Verbs . . . 43
4.33 Parts-Of-Speech (POS) Feature Amplification: Social Media . . 45
4.34 Drop Rates of POS Feature Amplification: Social Media 45
4.35 Parts-Of-Speech (POS) Feature Amplification: Reviews 45
4.36 Drop Rates of POS Feature Amplification: Social Media 45
4.37 Incorporating rule based features: Social Media 46
4.38 Drop Rates of Incorporating rule based features: Social Media 46
4.39 Incorporating rule based features: Reviews 47
4.40 Drop Rates of Incorporating rule based features: Reviews . . . 47
4.41 Incorporating rule based features: Reviews with Social Media 47
4.42 Drop Rates of Incorporating rule based features: Reviews . . . 48
4.43 Increasing the weightage of common words: Social Media . . 49
4.44 Drop Rates of Increasing the weightage of common words: Social Media 49
4.45 Increasing the weightage of common words: Reviews 49
4.46 Drop Rates of Increasing the weightage of common words: Reviews 50
4.47 Comparing Bias Reduction Methods 50

Chapter 1

Introduction

Millions of users generate large volumes of data online per day. For various applications, it is important to be able to process this data in real time and understand the general opinions of users without having to manually sort through this data. Various rule-based, machine learning-based, and a combination of the two techniques exist in performing sentiment analysis. One of the drawbacks of sentiment analysis however, especially for machine learning-based approaches, is that the usability of a sentiment classifier is largely dependent on the type of training data being fed to the classifier.

Users behave in varying ways across various different online platforms. This difference may arise due to a variety of reasons such as the subjects being discussed, the anonymity offered by a social media platform, the level of informality expected from a platform, and the amount of expertise held by an average user of a platform. As a result, when a sentiment classifier is introduced to a completely new testing data, it produces biased results due to the unfamiliarity of a new domain.

There are multiple scenarios that require an unbiased classifier to be able to perform well on unseen training data. For example, there may not exist sufficient training data for new online websites to successfully train their own sentiment classifiers. In this case, having an unbiased classifier trained on a different set of training data can be a useful alternative. In addition to this, we can also consider the use for sentiment classifiers in understanding

the reception of users to a newly launched product. Different kinds of products have different jargons in their reviews that denote sentiments. In this case, if a method exists to make a sentiment classifier unbiased then it can be successfully used to test the sentiment of a newly launched products without having to worry about the jargons,

In this thesis, we will start with exploring the various issues in training data that need to be overcome while using a sentiment classifier. We will then proceed to test out the various methods for training a sentiment classifier. After that, we will analyze the bias that occurs when a sentiment classifier is *cross-trained* on a different set of training data. We will then propose various methods to mitigate this bias and compare the results to denote the best possible solution,

1.1 Contributions

Sentiment Analysis has many real-world applications. It is highly sought out as a solution for detecting hate speech, understanding customer reception to various products and weeding out malicious users in various forums.

A major issue in utilizing sentiment analysis however is the need for a sufficient amount of training data. While this may be readily available in cases where there is an abundance of existing data, it is harder in cases of new websites or products. In this project, we explore methods to make a sentiment classifier useful for text obtained from completely unseen datasets. To do so, we first aim to reduce the classification bias that exists when a classifier is used to test text obtained from such unseen datasets.

While we explore reduction of bias in terms of sentiment classification, the same findings and methodologies can be modified and extended to solve the reduction of prediction bias in other cross-classification problems.

1.2 Road Map

In chapter 3, we will explore various datasets for sentiment analysis. We will evaluate their suitability for sentiment analysis while exploring the various kinds of noise in the dataset that will impact the performance of our classifiers. We will then suggest certain methods to reduce this noise and make our data more suitable for our problem statement. In Chapter 4, we will explore various methods to perform sentiment analysis and test them on our data. We will compare these results and determine the best approach in terms of accuracy. After this, we will detect the presence of *bias* when performing *cross-classification* within the datasets present in each domain of user-generated data. We will further propose various methods to reduce this *bias* and determine the best approach to do so. Finally, in Chapter 5 we will look into the applications and future work for our project.

Chapter 2

Literature Review

This research revolves around observing and mitigating classification bias in text obtained from social media data. This involves researching multiple subdomains of data science related to natural language processing, classification, and bias.

Before diving into the research details, we first look into existing work across these domains. The scope of this literature review involves analyzing methodologies to implement the various parts of this research. We start with defining what bias is in terms of data science, We then proceed to explore research in text classification and then move on to sentiment analysis and finally cross-domain classification.

2.1 Bias

In the current day and age, data science is heavily used across multiple domains for recommendations, classification and predictions. Multiple machine learning algorithms have been created and used depending on the problem statement. While these algorithms are not discriminatory by themselves, bias can seep into many applications due reasons such as unfair representation in training data, insufficient training data, or faulty feature selection.

Eirini Ntoutsi and colleagues [8] discuss the history of bias in data mining along with ways to mitigate bias depending on the use cases. A system is said to be *biased* when it has a higher probability of coming to one conclusion over others. While not inherently undesirable, various machine learning problems need to do away with bias to ensure that end-users are not being discriminated against.

Bias mitigation techniques focus on one of the three approaches – collecting data on bias awareness , modifying the machine learning algorithms to mitigate bias, and finally adjusting the results post-learning to account for bias. Through experiments, we will determine the presence of bias in a text classification problem (i.e, sentiment classifiation) by training on a Reddit dataset and testing on a Twitter dataset. We will then explore methods to identify the presence of bias in the outcomes followed by testing ways to mitigate this bias.

2.2 Text Classification

Text classification is not a new problem. Social media websites, online retail stores, and blogs are among numerous sources that generate a large amount of user text. Given the volume of text generated, it is often imperative to devise methods to classify text for analysis without any human involvement. Before implementing and testing one strategy for text classification, it is worth mentioning that there is no “best" strategy for text classification. In order to determine what strategy works the best, often the structure of the input data needs to be considered. Common strategies to process and classify text include, but are not limited to nearest neighbor methods such as KNN, Multi-class SVMs, Web-based categorization techniques, semantic labeling, and random walks.

2.3 Sentiment Analysis

Sentiment classification is a form of text classification problem discussed earlier. The rise in end-user generated, opinionated online content has led to the need of classifying text on the basis of the sentiment behind them. Sentiment analysis allows a classifier to gauge the underlying sentiment in some text. This is particularly useful in use cases involving reviews of various products and services.

The work by Walaa Medhat [15] provides a systematic study of various sentiment analysis algorithms and situations where their use are appropriate.

A strategic training of a sentiment analysis classifier includes processing the text, identifying the sentiment, feature selection, and finally classification of the sentiment.

There are many issues with data extracted from user generated content that can negatively impact a sentment analysis classifier. These issues are both semantic as well as syntactic. Under semantic issues, we consider the paper by Iti Chaturvedi [11] that discusses the issue in sentiment analysis related to differentiating between opinions and facts. This paper discusses the methodologies to analyze and remove data instances with neutral sentiments prior to training classifiers. In this paper, this sub-task is referred to as detection of subjectivity in sentiment analysis. They seek to prevent forcefully classifying data instances into positive and negative sentiments when they are neutral or fact-based.

Feature selection is an essential precursor to training the classifier. It is important to extract the correct subset of features from a given text in order to accurately represent the sentiment-related factors. Some popular options to select such features in text are detecting the polarity of objectives in text, identifying the presence of negations in text, and noting the frequencies of using certain terms in speech.

Feature selection is followed by selecting a classification algorithm. The algorithm is responsible for training the classifier. Sentiment classifiers can be roughly categorized into three broad categories. These are the machine learning-based approaches, lexicon-based approaches, and a hybrid of the previous two approaches.

For our project, we have considered sentiment analysis approaches for social media and product reviews datasets. A survey by Lin Yue [17] focuses solely on sentiment analysis performed on social media data. This paper provides an extensive study of the current status of sentiment analysis algorithms. They discuss sentiment analysis approaches under granularity oriented sentiment analysis, document level sentiment analysis as well as sentiment analysis based on individual words and sentences. In therms of methodologies, they broadly classify sentiment analysis approaches in terms of supervised learning, unsupervised learning and finally semi-supervised learning approaches. For the datasets, they consider Twitter as well as Facebook datasets in order to test and analyze various approaches.

2.3.1 Machine Learning-based Approaches

Various existing machine learning algorithms can be redesigned in order to fit a sentiment classification problem. In this case, the training file consists of a set of records along with their target sentiment class. Machine learning algorithms can further be broken down into supervised and unsupervised learning algorithms. Supervised machine learning algorithms are appropriate when we have training data that already contain the target sentiment class for each record. Traditional machine learning classifers such as Naive Bayes classifiers, Bayesian networks, Neural networks, Decision tree classifiers, and Support Vector Machines all have comparable accuracies in sentiment classification. For the scope of this project, since we have tweets and

comments with their associated targets, we can initially consider supervised learning algorithms for training the classifiers.

Support Vector Machines are especially useful in classification of text documents. This is due in part to the sparsity of the features in a text document and the relatedness of the features. Chen [4] focuses on utilizing support vector machines to classify sentiment polarity of user reviews.

A lot of research has been done into utilizing machine learning for classifying sentiments for social media data. Twitter provides one of the best sources to extract text classification data from as metrics such as likes and retweets provide a good metric for labeling the data. A paper by Yung-MingLi [18] focuses more on classifying tweets using support vector machines while also adding the feature of user credibility when training the classifier. A survey by Anastasia Giachanou and Fabio Crestani [9] analyzes and compares various methods to perform sentiment classification on Twitter data. In this survey, they discuss various Twitter specific data challenges that need to be dealt with to perform sentiment analysis. These include length of text, relevance of topics, correctness of grammar, sparsity of data, stop words and multilingual content. They go ahead to discuss various features present in this text based dataset. These include semantic features such as the different kinds of negations and opinion words present in the text as well as syntactical features such as term frequencies, bigrams, n grams and parts of speech. Machine learning approaches are discussed that use SVMs to predict the sentiments of the data instances as positive, negative and neutral Tweets. They further discuss evaluation metrics such as F measure, precision, recall and accuracy in order to evaluate the classifiers.

2.3.2 Other Approaches

Sentiment classification algorithms are not limited to using machine learning approaches. In the case where target classes are not available, a lexicon-based approach can be used to determine the sentiment behind classes. This can involve incorporating dictionaries to identify positive and negative words, statistical approaches to identify patterns or corpus-based approaches that make use of opinion words along with connectives and conjunctions. As the scope of this project is to identify and mitigate bias, our focus is more towards a machine learning-based approach. We also look into a more hybrid learning model that involves lexicon based segregation before utilizing machine learning for training.

The work of Dalibor [3], shows that it is apparent that we have various challenges when it comes to analysis of user generated text. These include, but are not limited to, presence of sarcasm in text, ironic sentences, fake news, indistinguishable facts and opinions and unstructured data. The algorithm selected needs to do away with challenges that harm the problem settings.

2.3.3 Applications of Sentiment Classification

The work of [2] studies sentiment analysis in the movie reviews domain. Movie reviews are some of the more convenient datasets to perform sentiment analysis on due to widely available databases for training and a clear sentiment class in the form of the 'movie ratings' provided by the users along with their text-based reviews. The authors employ three primary machine learning methods for their study. These are namely Naive Bayes classifier, maximum entropy classifiers, and support vector machines. These machine learning models employ standard bag-of-words features. Such an implementation has a provably higher accuracy than random baselines and classification models with selected unigrams in predicting sentiments across the

movie-reviews domain.

We note that the work of [7] best resembles our approach for sentiment classification. This approach involves using word2vec to convert Chinese words into their corresponding vectors. Support Vector Machines are then employed to train the classifier into detecting sentiments.

Another similar line of research is the work of [12], which deals with sentiment analysis on social media data obtained from Twitter. The obtained tweets are preprocessed and cleaned to include parts-of-speech tags as well as reduce words to their origins. The sentiment analysis problem is then further broken down into two sub-problems, namely the target-dependent classification problem and the target-independent classification problem. The target-dependent problem deals with sentiment classification in tweets with the correct sentiment class provided. This proved to display a greater accuracy as compared to a target-independent classifier. The latter focuses more on the content and lexicons in the tweets rather than their subject. The authors further improved accuracy rates by employing a graph-based optimization framework to assign sentiments to tweets based on retweets.

While we discuss text base sentiment analysis problems in this project, it is worth looking into sentiment analysis for user generated content that involves input data in forms other than that of text. A survey by Alessandro Ortis [1] discusses sentiment analysis in terms of user responses to various images. Features in this case are features pertaining to Valence, Arousal and Control. They discuss classifying the polarity of images in terms of five fine-grained sentiments - anger, fear, disgust, surprise and sadness.

2.4 Parts of Speech

Any classification problem involves feature engineering and often a way to generate the best possible subset of features that are relevant to the problem

at hand. Sentiment analysis deals with the detection of the emotions behind a user's textual inputs. From analyzing the English language, we can hypothesize that various *parts of speech* play an important role in concluding sentiments. With this in mind, we take a look into 'parts of speech' as a possible way fir feature engineering.

The use of parts of speech tags as features for sentiment analysis has been explored before. The study by [5] employs the use of parts of speech tags as a method to weigh certain lexicons prior to training a machine learning classifier for sentiment analysis. The study initially trains a sentiment analysis classifier without any special weighing. After this, a classifier is retrained based on the following subsets of features: nouns, verbs, adjectives, and adverbs. After testing the classifier on these subsets, they are retrained based on all permutations and combinations of the four subsets and the optimal subgroup of POS tags is selected. These subsets are assigned higher weights when training a classifier. Their results show that adjectives and adverbs seem to have high relevance in determining the sentiment behind a given text. This is later tested and incorporated into our project to improve accuracy.

Similarly the work of [16] discusses Parts of Speech in sentiment analysis for Twitter as a method to reduce the high dimensionality of a text corpus by reducing the number of features in a text. This paper explores this goal through a proposed method that helps to determine how related a parts of speech feature is to the overall sentiment of a given text (using χ^2). This dependency is then used to create a composite set of features that is weighted based on the relevance of the feature.

2.5 Cross-Domain Classification

The final stage in detection of bias during classification is testing a classifier on completely new text. Cross-classification in text-based applications involves training a classifier over one dataset and then testing its accuracy on another dataset obtained from an unseen domain. Theoretically, this would lead to lower accuracies as user behaviors across online platforms typically vary in terms of language and format of the generated text. There is also the issue of the kinds of content generated. Reddit, being anonymous, typically has larger amounts of controversial comments and 'troll' comments as compared to Twitter, where the identity of the poster is generally known. Our objective is to spot the bias in languages used across Reddit and Twitter by employing sentiment classification techniques. We can further analyze the potential for removing such a bias.

Various state of the art algorithms exist to perform cross-domain classification of text documents. The most popular out of these are the *spectral feature alignment* and *structural correspondence learning* algorithms. For instance, [6] provides an interesting approach towards training classifiers across multiple domains. Their work expands into the previously discussed concept of target-dependent classifiers. The dataset involves reviews taken for two separate products or 'targets'. For the sake of the experiment, four different targets are taken - books, DVDs, electronics, and kitchen appliances. Cross-domain classification is a challenging problem when it comes to text-based domains as the features of the two training domain do not align with each other. This problem can be partially solved by creating a *sentiment specific thesaurus*. Here, lemmatized sentences can be broken down to create feature vectors that contain words that express similar sentiments to each other.

Another approach to cross-domain sentiment classification is covered by the work of [10]. The paper also considers the reviews domain in sentiment

classification. Different categories of products may have different wordings for equally positive and equally negative reviews. With this notion, training a classifier on one domain will give inaccurate results when training on a completely new domain. The address this problem using a Topical Correspondence Transfer (TCT) approach, where different domains are represented using a term-document matrix and their relationships are explored.

In other approaches of cross-domain classification, we explore the research paper by Paola Zola [19] that discusses cross-domain classification in terms of two completely different domains of user generated data. In this case, data obtained from websites such as Amazon and Tripadvisor are used to train a machine learning based sentiment classifier, This classifier is then used to test on datasets from social media such as Facebook and Twitter. Various evaluation metrics such as F score, accuracy and ROC are then used to test the efficiency of cross-classifying.

Chapter 3

Datasets

In this chapter we will analyze the datasets that have been used for our experiments. For this project, we will obtain user-generated text data from two different domains and observe the similarities in bias trends and sentiment analysis accuracies across both the domains. The datasets we have picked are Reddit datasets and Twitter datasets obtained from the social media domain as well as the wine reviews and book reviews datasets obtained from the product reviews domain.

In the following sections, we will analyze the breakdown of this data, the noise present in it and methods to mitigate this noise in order to make the data more suitable for sentiment analysis.

3.1 Datasets

Our data has been sourced from social media posts and review datasets. We start with smaller subsets of data and proceed to using larger datasets to measure the change in accuracy and compare the eventual similarities between the datasets.

The data is manually labeled with sentiments corresponding to each tweets, or comments. Value -1 denotes a negative sentiment, +1 a positive sentiment, and 0 a neutral sentiment.

The datasets for both these domains for this project have been sourced from Kaggle.

3.1.1 Social Media Datasets

We run our first set of experiments on data obtained from social media websites. This data has been extracted from English language comments obtained from Reddit as well as English language tweets that are obtained from Twitter.

Reddit Dataset

The Reddit dataset is a subset of approximately 100,000 comments taken from Reddit and labeled with 0,1, or -1 depending on the correct sentiment class. To accurately depict a comparison between datasets, we obtain a distribution of the Parts-Of-Speech tags in the text corpus. The distribution is shown in Figure 3.1.

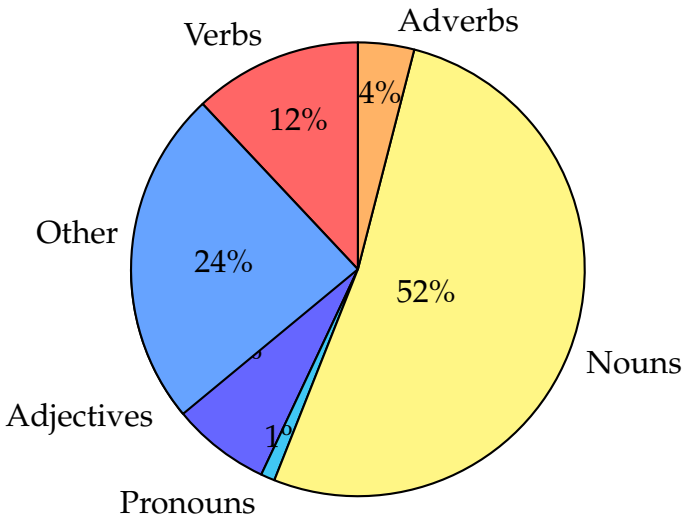

FIGURE 3.1: Distribution of Reddit data

Twitter Dataset

The Twitter dataset is a subset of approximately 100,000 English tweets taken from Twitter and labeled with 0,1 or -1 depending on the correct sentiment class. We find that the Parts-Of-Speech distribution for this text corpus is similar to that of Reddit (see Figure 3.1). The distribution is shown in Figure 3.2

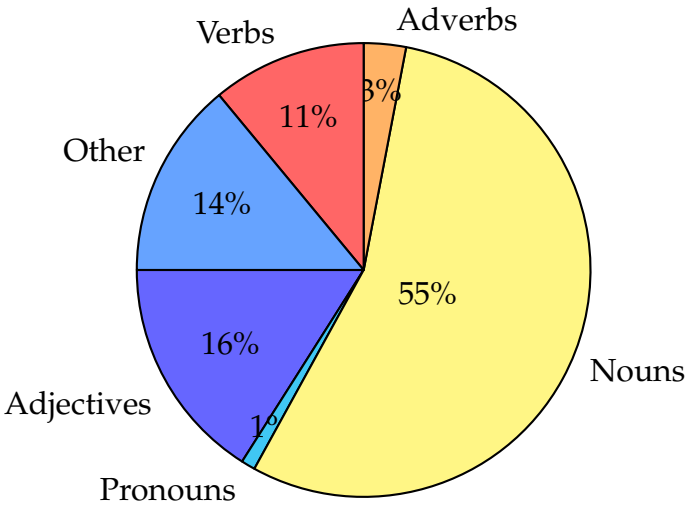

FIGURE 3.2: Distribution of Twitter data

3.1.2 User Reviews Dataset

Most online retail websites provide the option for end-users to leave text-based reviews as well as ratings. This data is highly suitable for sentiment analysis as the sentiment class can be inferred from the user ratings instead of manually assigning the sentiment class. For demonstration, we obtain reviews data for two kinds of products: books and wine.

Wine Reviews Dataset

The wine reviews dataset is a subset of 280,000 reviews of wines. Users assign each product a rating from 1-100 and also leave a text-based review. Through the ratings, we check the distribution and assign values of -1,0 and

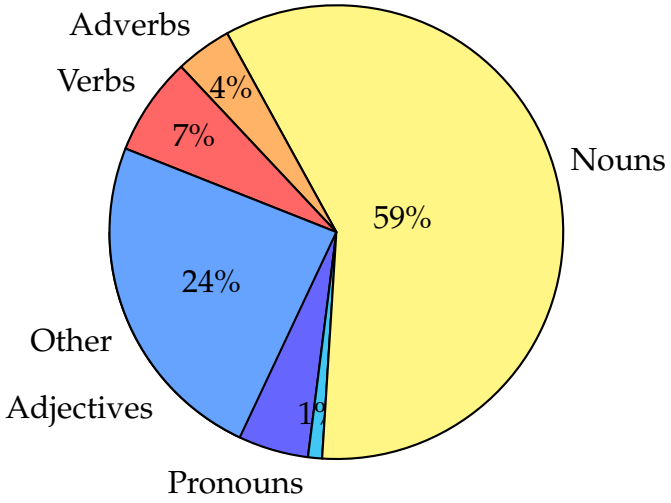

FIGURE 3.3: Distribution of Wine Reviews data

1 sentiment classes to each review. As with social media datasets, we also calculate the Parts-of-Speech distribution for this dataset. We can see that this (Figure 3.3) too follows a similar distibution to Figures 3.1 and 3.2, with a majority of the text being Nouns.

Book Reviews Dataset

The book reviews dataset is a subset of 280,000 reviews of books obtained from Amazon. Users assign each product a rating from 1-5 and also leave a text-based review. Through the ratings, we again check the distribution and assign values of -1,0 and 1 sentiment classes to each review. We also calculate the Parts-of-Speech distribution for this dataset, as shown in Figure 3.4

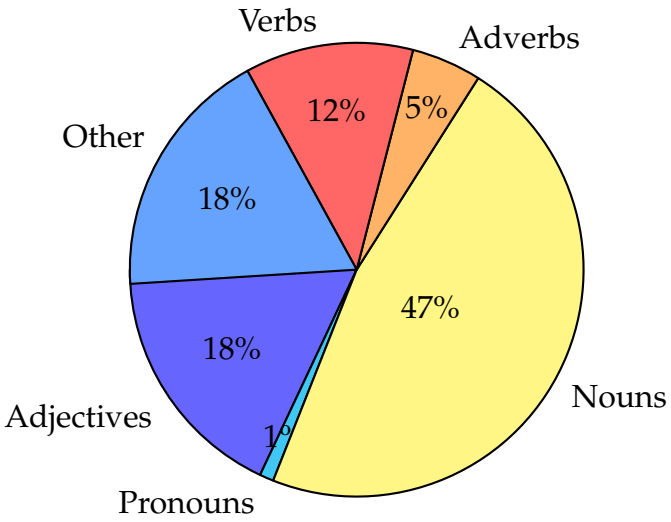

FIGURE 3.4: Distribution of Book Reviews data

3.2 Noise in Data

One of the main reasons that text classification problems are challenging is that user-generated text is highly unreliable. This unreliability comes from the various inconsistencies with which users choose to interact with one another online. As we are using text-based user-generated data to conduct our study, the data cleaning approach will be more focused on the noise surrounding user behaviours on these platforms.

We can classify noise in user-generated text into two categories: issues related to clarity of speech, and issues related to unusable features in text.

3.2.1 Unusable Features in Text

Stop words

Stop words are words that make topical and grammatical sense when used in a sentence but otherwise add no impact on determining the overall sentiment behind it. This includes words such as `I`, `me`, `you` and `your`. The goal of cleaning the data should be to make the training data as meaningful as

possible. As these stop words do not contribute to the underlying sentiment of a sentence, they are classified as noise for sentiment classification. Each data instance should have a majority of words that can indeed contribute to determining the sentiment class. This would ensure that unusable or noisy features do not impact the data instance averages or shift the data distribution.

Spelling mistakes

Websites that allow user-generated text provide an informal environment for users to interact with each other. Due to this, end users are under no pressure to proof-read text before posting them on a website. Data scraped from social media sites is hence often ridden with spelling mistakes or “typos". These features are meaningless when converted to vectors and add noise to the training data.

Special characters and numbers

Numbers and special characters are unreliable indicators of sentiment in social media data. An exception to the rule is when special characters combine to form emoticons. Exclamation marks may add heightened polarity to the sentence and question marks can add ambiguity. However, for most parts it is more useful to have training data that does not account for this possibility.

Mixture of Languages

Social media texts can be a mixture of multiple languages. For the scope of our research, we will be dealing with only the English language and so, the presence of other languages in the text will be considered as noise.

3.2.2 Clarity of Speech

Synonyms

In a text corpus, we often have multiple words that have similar meanings. This is an issue when using text for any kind of classification problem. If similar words can be mapped to the same vectors, then we can reduce the number of features that determine a certain outcome and subsequently improve the efficiency of the classifier.

For instance, words such as "beautiful":"pretty" imply the same positive sentiment whereas words like "sad":"upset" imply the same negative sentiment. If these words can both be reduced to the same n-dimensional vector, then we can increase the meaningfulness of these words and ensure that the classifier learns how to identify the sentiment behind them more accurately.

Variations of words

Various parts of speech such as adjectives and adverbs have different forms of their word that amount to the same meaning. For instance, superlatives and comparatives combine the same root word with a prefix or a suffix. "happy":"happier" and "bright":"brighter" convey the same information when it comes to determining the polarity of a sentiment. It is more meaningful for a classifier to use the same vectors for such words that have the same roots.

Negations

Negations include words such as 'not', 'non-' and 'un-'. It is important to appropriately account for the impact of negations on the polarity. For instance, ignoring these negations leads to opposite sentiments. The weights that a negative sentiment holds should be appropriate. For instance, given the text "Not great", if the words "Not" and "great" both account for opposite

and equal weights, then the aggregate sentiment would turn out to be 0 or 'neutral', even though we can clearly see that it is a negative sentiment.

Sarcasm

Sarcasm is one of the most challenging properties of text to deal with in text processing. Although certain features strongly suggest one polarity, the actual context behind a sentence may imply the opposite sentiment.

Vagueness

Not all user-generated text contains the presence of highly polarized statements. Many generated statements only contain vaguely positive or negative wordings. This can add unnecessary features to a classifier in the form of vectors that do not always contribute to the polarity of a sentiment.

Hyperboles

Hyperboles or exaggerations can denote a sentiment opposite to the literal meaning. For instance, "This is just great." may mean that things are in fact, not great according to the context in which the sentence is being spoken. While a human mind can easily infer the correct sentiment behind the words, they have the potential to confuse a classifier.

3.2.3 Domain Specific Data Issues

Apart from the issues listed above, there are some data issues specific to the context of the user-generated text.

Social Media Dataset

Social media data is generally loaded with slang words, misspellings, and informal jargons. This makes it hard for a machine to understand words out of

context. Another issue with social media datasets for sentiment classification is the label for each comment or tweet. The correct label cannot be inferred from the number of upvotes for the Reddit dataset as a negative opinion may also be a popular one. Similarly for Twitter, a tweet having a high number of retweets does not automatically mean that the tweet carries a positive sentiment as a number of factors could lead up to that. As a result, we have to rely on human judgement to get the 'correct' polarity values for a given Reddit comment or Twitter tweet.

Reviews Dataset

For the reviews dataset, we include reviews for wine as well as reviews for books. An upside to using a product reviews dataset for sentiment analysis is the fact that we can infer the sentiment class based on the ratings provided by the users. In case of product reviews however, the words used to describe one product might denote a positive sentiment just for that specific product. This would make an improperly trained classifier produce incorrect results.

3.3 Data Cleaning

In the previous section, we have discussed multiple issues with the training data that impact the accuracy of a sentiment classifier. While some of those issues need to be handled by making adjustments to a classifier or penalizing the outputs of the classifier, a reasonable number of these issues can be resolved by cleaning the data before training the classifier.

The following steps have been taken to clean the data to make it appropriate for a classifier.

Removing stop-words

For this issue, we make use of the nltk 'stopwords' library. This library provides a list of stop words that are irrelevant to sentiment analysis. Every instance of the input data is processed and the words matching the members of this set are removed.

Reducing variations in text

The PortStemmer class in the nltk package takes a word and reduces them to their root words. Preprocessing text in such a way ensures that words with the same roots get converted to the same vectors, thus reducing the number of features in a meaningful way.

Removing numbers and special characters

We use Regex pattern matching to remove special characters and numbers present in a text to make them more suitable for a sentiment classifier.

Removing words from a different language

We make use of a dictionary to ensure that the words that are being fed to a classifier are a part of the English language and can be reliably used in a classifier. Data instances containing no English words are skipped while vectorizing the words.

Fixing spelling mistakes

Spelling mistakes can be fixed by taking words that are misspelled and using a dictionary to find the closest matching words. These words are then replaced with their correct spellings. The same words may be misspelt in multiple places in different ways and so fixing all spelling mistakes ensures a reduction in noisy data.

Adding parts of speech tags

The final stage of data cleaning is to add tags to the input data determining what kind of words are present in a text. This will be useful later during sentiment classification to perform feature engineering to determine the most relevant parts of speech in terms of determining polarity.

Chapter 4

Methodology and Results

In this chapter, we will discuss our approach using various sentiment analysis technologies on our datasets. We will then discuss the bias associated with cross-classification and propose various methods to mitigate this bias.

4.1 Sentiment Classification

In order to test the bias trends in cross-training using classification, we consider sentiment analysis, a subproblem of text classification. Sentiment analysis is a complicated text classification problem due to the various nuances in text that determine its overall sentiment. For instance, the presence of sarcasm, negations in text, and figures of speech all make sentiment classification more complicated than a regular context-based text classification problem.

For demonstration purposes, we can test out various sentiment analysis approaches to understand the best approach for our dataset. Sentiment analysis methods can broadly be classified into rule-based and machine learning-based approaches or conversely, a hybrid of the previous two.

4.1.1 Rule-Based

Rule based sentiment analysis approaches deal with individual subsets of text rather than the entire training set as a whole. These approaches are unsupervised and do not need a sentiment class in order to make predictions. The objective of this research does not focus on unsupervised learning and we focus more on the challenges regarding machine learning algorithms.

For our experiments, we consider the VADER rule-based approach. This approach combines a cumulative scoring of sentiment specific words with considerations for hard-coded language features such as negations and words that denote extremeness of a sentiment. Each analyzed text receives a score denoting the positivity, neutrality, and negativity present in the sentence along with a cumulative score that combines the aforementioned values with the hard-coded values for various sentiment features. This cumulative score is then normalized between -1 and 1 values, following which we can bin them into -1, 0 and 1 values denoting negative, neutral, and positive values of sentiment.

Rule-based approaches are cost effective and practical for sentiment analysis. In problems that involve text along with a rating such as in the case of movie reviews and product reviews, the 'target' value for a sentiment is already available in the form of the rating (we assume that a high rating would be accompanied by a positive text). However, in almost every other case, it can be tedious to obtain an accurate metric for determining the correct sentiment of a given text. In these cases, a rule-based approach would provide a quick and inexpensive way for sentiment analysis.

From the results of the experiments performed in tables 4.1 and 4.2, we can see that machine learning methods largely outperform rule-based ones for multiple datasets. For our social media datasets, we can see that VADER

provides a comparable accuracy. The reviews datasets however have a significant performance gain when switching to a machine learning approach. Rule based algorithms tend to stagnate in terms of their accuracy and are demonstratively inferior to machine learning and hybrid approaches.

TABLE 4.1: Rule based sentiment classification: Social Media

Social Media Dataset	**Vader Performance**
Reddit	42.8784267
Twitter	44.34813569

TABLE 4.2: Rule based sentiment classification: Reviews Dataset

Reviews	**Vader**
Wine	43.57478257
Books	50.73638043

4.1.2 Machine Learning-Based

One of the reasons text classification is challenging is the sparseness of the features. In an unprocessed text corpus, the number of features is equal to the number of words in the vocabulary. This leads to unreliable results when it comes to training a classifier.

In order to train a classifier correctly, we first explore methods for dimensionality reduction. Dimensionality reduction methods convert words into n-dimensional vectors. We can do this in multiple ways. One approach is to use the standard Bag-of-Words method. In this method, each word in the vocabulary is a feature. The value of the feature refers to the frequency of the occurrence of that word in the give sentence. While bag of words method has comparable accuracy in small datasets, it is undesirable in a text corpus with a larger vocabulary due to the sparseness of the features. Another issue with this approach is the loss of sequence of words.

A more recent approach to vectorize words is by using the WORD2VEC or DOC2VEC algorithms. This approach mitigates the sparseness problem in the bag-of-words approach and converts each word to an n-dimensional vector. Each instance in the dataset is then converted to an average of the vectors of the words present in that sentence.

After the dataset is vectorized using one of the two approaches listed above, it can be used to train the classifier. We have used multiple classification techniques (Decision Trees, SVM, Random Forest, Naive Bayes, and XGBoost) and compared the classification performance. Each classifier has its own set of advantages and trade-offs. We use multiple classifiers to track bias and accuracy rates to ensure incorporating their advantages into our final results. For completeness we briefly review these classification methods.

TABLE 4.3: Machine Learning Approach - Social Media Dataset

Algorithms	**Reddit(R)**	**Twitter(T)**
Decision Trees	49.06546275	48.45204723
SVM	53.69751693	36.38149414
Random Forest	61.21896163	60.97071883
Naive Bayes	50.96162528	47.31845061
XGBoost	61.89616253	61.50068548

TABLE 4.4: Machine Learning Approach - Reviews Dataset

Algorithms	**Wine(W)**	**Books(B)**
Decision Trees	85.41253812	72.04613687
SVM	77.40385186	53.30303426
Random Forest	90.51156388	83.27894531
Naive Bayes	56.46087165	50.34309057
XGBoost	83.43558282	84.99839803

We can see the results of our machine learning approach in tables 4.3 for Social Media datasets and table 4.4 for product reviews datasets. It is clear that the machine learning approach outperforms the rule-based approach in

all instances. It can also be seen that XGBoost and Random Forest outperform all the other classifiers in terms of accuracy scores for sentiment analysis.

Decision Trees

A decision tree is classification method used for a dataset, which allows determine the best possible features from multiple features in data. The algorithm determines a set of sequential *best-splits* in order to help the classifier determine the correct target class. In problem statements with large number of features, this helps the machine determine the most relevant features. A downside of using decision tree at times is their tendency to overfit the data. Pruning of the tree needs to be done sometimes to ensure that this does not occur.

SVM

Support Vector Machines(SVMs) Support Vector Machines is a supervised machine learning algorithms. The method is initially designed and sometimes best suited for two-class classification problems such as our current sentiment classification problem. In our case, the two classes are *positive* and *negative*. SVMs process the input data and learns a decision boundary that separates the data points.

Random Forest

Random Forrest is an algorithm that involves the construction of multiple decision tree classifiers. The output class of the classifier is a mean of the results of individual decision trees. Random forest often yields a higher accuracy than that of decision trees and can often overcome overfitting.

Naive Bayes

Naive Bayes classifiers are simple probabilistic classifiers. These classifiers assume an independence of all the features given the class variable in the input data. These classifiers are highly scalable, making them suitable for text processing problems with a large number of features.

XGBoost

XGBoost is a recent machine learning algorithm. XGBoost[13] [14] provides an implementation of decision trees that are gradient boosted for heightened accuracies and performance. This method often outperforms random forest in terms of both speed and accuracy. We can see in our implementation that this method continually outperforms the other classification algorithms in terms of performance.

4.1.3 Hybrid Approach

For this approach, we make use of VADER's sentiment scores as a feature in addition to the machine leaning methods discussed earlier. From training a classifier with the rule-based approach discussed earlier, we have the positive, negative and compound scores listed for each data instance in the dataset. We take the N dimensional feature vector and convert it to $N + 2$ by adding 2 more dimensions- one with the negative score and one with the positive score for every data instance. This $N + 2$ dimensional vector is then passed to the classifier for training.

The results of this experiment are demonstrated in table 4.5 for social media datasets and table 4.6 for the product reviews dataset. We can see that this method outperforms the rule-based and machine learning methods for sentiment analysis.

TABLE 4.5: Hybrid Approach - Social Media Dataset

Algorithms	**Reddit(R)**	**Twitter(T)**
Decision Trees	55.19638826	52.93732479
SVM	60.00902935	36.98717031
Random Forest	66.65462754	65.03857093
Naive Bayes	55.25959368	50.99343169
XGBoost	67.11512415	65.10404944

TABLE 4.6: Hybrid Approach - Reviews Dataset

Algorithms	**Wine(W)**	**Books(B)**
Decision Trees	85.44813756	73.35738273
SVM	77.41571834	56.53308967
Random Forest	90.46528462	83.66223256
Naive Bayes	74.40400612	75.57285424
XGBoost	83.7797107	85.51696313

4.1.4 Comparison of Sentiment Analysis approaches

We train each of the four datasets using all the three aforementioned sentiment analysis approaches. We first go over the rule-based approach using the VADER algorithm. Next, we train using the WORD2VEC features obtained on our datasets and the five classifiers. This allows us to obtain the "best" classifier among them. Finally, we compare the previous two methods by incorporating VADER scores along with the WORD2VEC vectors prior to training the classifier.

We can see that in the case of the social media datasets(Table 4.7), the accuracy is significantly higher using a hybrid approach. In the case of the reviews datasets(Table 4.8, we find that the machine learning and hybrid approaches both have comparable accuracies but both outperform the rule-based approach. Figure 4.1 helps to illustrate these results.

TABLE 4.7: Comparison of Sentiment Analysis - Social Media Dataset

Social Media	Vader	ML	Hybrid
Reddit	42.8784267	61.89616253	67.11512415
Twitter	44.34813569	61.50068548	65.10404944

TABLE 4.8: Comparison of Sentiment Analysis - Reviews Dataset

Reviews	Vader	ML	Hybrid
Wine	43.57478257	83.43558282	83.7797107
Books	50.73638043	84.99839803	85.51696313

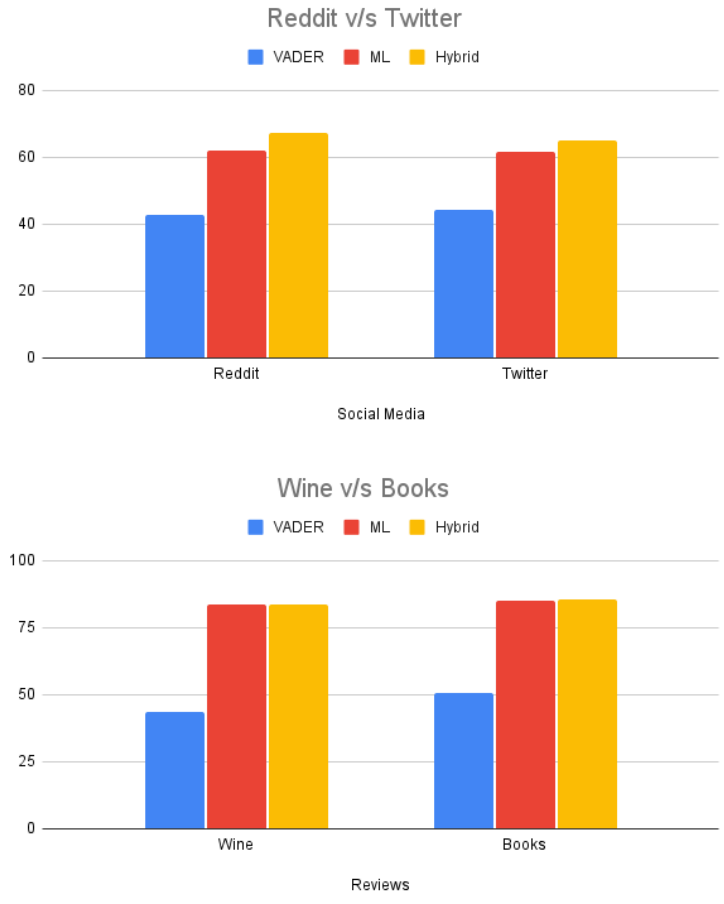

FIGURE 4.1: Comparison of Sentiment Analysis approaches

Results on Social Media Datasets

We train sentiment classifiers for our two social media datasets - Reddit and Twitter. We observe that a hybrid approach performs the best for both the datasets with the machine learning approach coming as a close second. The accuracy for a sentiment analysis classifier for social media datasets varies from 40-45 for VADER, 60-62 for Machine learning, and 65-70 for a hybrid approach, incorporating the previous two approaches. A lower accuracy of a sentiment analyzer can be attributed to the varying subjects of text present in the training data obtained from a social media domain.

Results on Reviews Datasets

For the reviews dataset, we obtain the training and testing data from two different subsets of reviews - reviews of wines and books. For this approach, we can see that the machine learning approach and the hybrid approach have a similar accuracy, with the hybrid approach slightly outperforming the other. In this case as well, we can see that VADER does not have high accuracy. For the reviews dataset, we can see that a sentiment analyzer outperforms the results obtained on social media data. This is due to the fact that the reviews dataset has a clearer depiction of the 'correct' class based on the ratings provided. It also helps that in both these datasets the subject discussed is consistent, providing a more reliable training data.

4.2 Cross-Domain Classification

After training and testing a classifier on dataset of one domain at a time, we run a few more experiments. The goal of this next set of experiments is to check the performance of one classifier on completely unseen data.

Users generally behave differently on different online platforms. Due to this, classifiers perform notably worse when testing on data from a different platform, even though the language and source of data is the same. We use the same set of classifiers trained in the previous section and perform cross-classification in order to spot this drop in accuracy.

Figure 4.2 explores the steps of cross-classifying using datasets obtained from two completely dissimilar text domains.

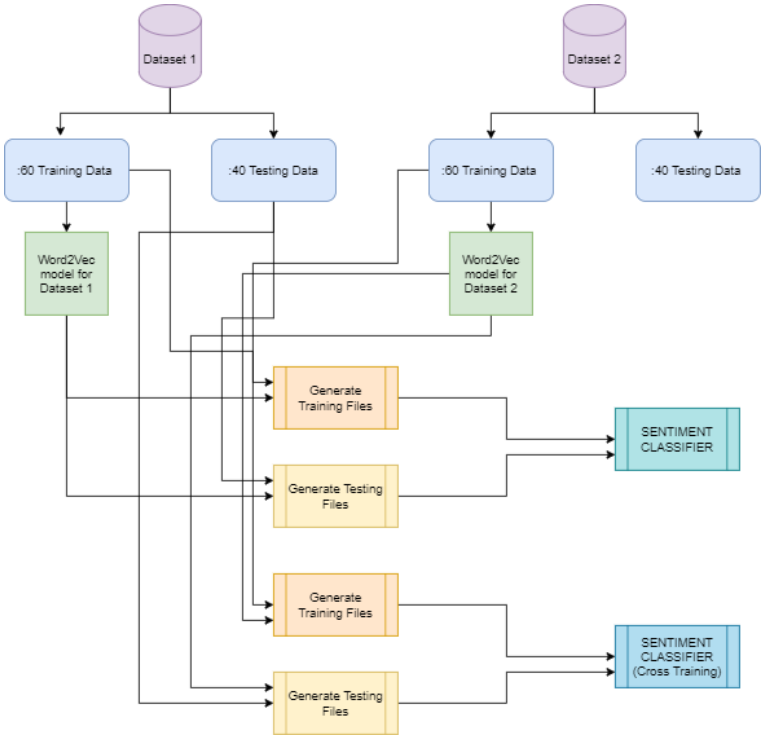

FIGURE 4.2: Cross-Domain Classification

4.2.1 Social Media

While social media sites have a lower accuracy for a sentiment classifier, we can see that a cross-trained classifier is not as biased in this case, providing an average drop in accuracy of around 10% in cases of machine learning and hybrid approaches(Table 4.9 and 4.11). This suggests that for this domain, users often behave (i.e., write) in a similar manner, reducing the presence of behavioral bias.

TABLE 4.9: Cross-Domain - Social Media Dataset

Algorithm	Reddit(R)	Twitter(T)	Cross-Training R(data)T(model)	Cross-Training T(data)R(model)
Decision Trees	49.06546275	48.45204723	45.54401806	42.21931207
SVM	53.69751693	36.38149414	36.93002257	49.86392748
Random Forest	61.21896163	60.97071883	60.06320542	53.18696159
Naive Bayes	50.96162528	47.31845061	45.93227991	42.15997217
XGBoost	61.89616253	61.50068548	61.76975169	52.73475067

TABLE 4.10: Cross-Domain Drop Rates - Social Media Dataset

Algorithm	R(data)T(model)	T(data)R(model)
Decision Trees	0.07177033493	0.1286371891
SVM	0.3122582815	-0.3705849269
Random Forest	0.01887905605	0.1276638588
Naive Bayes	0.09868887314	0.1090162162
XGBoost	0.002042304887	0.1425339367
Average	0.1007277701	0.02745325477

TABLE 4.11: Hybrid Cross-Domain - Social Media Dataset

Algorithm	Reddit(R)	Twitter(T)	Cross-Training R(data)T(model)	Cross-Training T(data)R(model)
Decision Trees	55.19638826	52.93732479	49.95936795	48.99019869
SVM	60.00902935	36.98717031	38.78103837	55.42755417
Random Forest	66.65462754	65.03857093	66.35665914	59.36035686
Naive Bayes	55.25959368	50.99343169	50.50112867	48.1860408
XGBoost	67.11512415	65.10404944	67.69300226	59.35012584

TABLE 4.12: Hybrid Cross-Domain Drop Rates - Social Media Dataset

Algorithm	R(data)T(model)	T(data)R(model)
Decision Trees	0.09487976444	0.07456225117
SVM	0.3537466145	-0.4985616287
Random Forest	0.004470333243	0.0873053327
Naive Bayes	0.08611111111	0.05505397055
XGBoost	-0.008610251581	0.08838042556
Average	0.1061195143	-0.03865192974

4.2.2 Reviews

For the reviews dataset, it is apparent that the classification bias is much more evident when cross-training a classifier for both, machine learning and hybrid approaches(Table 4.13 and Table 4.15). This is due to the fact that for both the wine and books datasets, users are taking about completely different products. The jargons and keywords used in these datasets are vastly different from each other, rendering a classifier barely usable on the other dataset. In the next section, we will look into reducing this classification bias and explore various existing and proposed techniques to do so.

TABLE 4.13: Cross-Domain - Reviews Dataset

Algorithm	Wine(W)	Books(B)	Cross-Training W(data)B(model)	Cross-Training B(data)W(model)
Decision Trees	85.41253812	72.04613687	48.76291963	52.36558247
SVM	77.40385186	53.30303426	44.51828031	59.14727486
Random Forest	90.51156388	83.27894531	47.7056164	57.30678407
Naive Bayes	56.46087165	50.34309057	46.0773756	49.3098139
XGBoost	83.43558282	84.99839803	50.77547436	60.29713662

TABLE 4.14: Cross-Domain Drop Rates - Reviews Dataset

Algorithm	W(data)B(model)	B(data)W(model)
Decision Trees	0.429089444	0.2731659914
SVM	0.4248570421	-0.1096417997
Random Forest	0.4729334644	0.3118694785
Naive Bayes	0.183906053	0.02052469681
XGBoost	0.3914410058	0.2906085524
Average	0.3804454019	0.1573053839

4.3 Bias in Cross-Classification

4.3.1 Analyzing drops in accuracy

We can see from training sentiment classifiers on multiple domains of user-enerated text that a classifier does not perform well when working with test

TABLE 4.15: Hybrid Cross-Domain - Reviews Dataset

Algorithm	Wine(W)	Books(B)	Cross-Training W(data)B(model)	Cross-Training B(data)W(model)
Decision Trees	85.44813756	73.35738273	51.62748751	51.7414057
SVM	77.41571834	56.53308967	44.81019568	61.6511018
Random Forest	90.46528462	83.66223256	50.26758909	57.6449787
Naive Bayes	74.40400612	75.57285424	47.10873254	60.41936135
XGBoost	83.7797107	85.51696313	51.03772353	61.09575061

TABLE 4.16: Hybrid Cross-Domain Drop Rates - Reviews Dataset

Algorithm	W(data)B(model)	B(data)W(model)
Decision Trees	0.3958032441	0.2946666882
SVM	0.4211744509	-0.09053126509
Random Forest	0.4443438795	0.3109796746
Naive Bayes	0.3668522033	0.2005150269
XGBoost	0.3908104586	0.2855715594
Average	0.4037968473	0.2002403368

data taken from an unseen subdomain. From the initial sentiment cross-classification experiment, we can see that the drop rate is higher in cases of the reviews datasets(Table 4.14) as compared to social media datasets(Table 4.10). This would suggest that users interact in a more consistent manner in different social media platforms as opposed to reviews datasets. For the reviews datasets for instance, we deal with wine reviews and book reviews. Words like `delicious`, `aromatic`, `fruity`, and `acidic` are used to exclusively describe positive sentiments in reference to wine reviews whereas `pungent` and `unbalanced` denote negative sentiments. In contrast, `interesting`, `riveting` and `amusing` denote positive sentiments only for books whereas `boring` and `unreadable` suggest negative sentiments. Since adjectives used in such a context are vastly different, a classifier does not recognize them when test data from a different subdomain is introduced to it.

In this section, we will explore various experiments to potentially reduce this drop in frequency when switching to an unseen text domain. The goal

of these sets of experiments is to train classifiers that are more suitable to be used on multiple domains, regardless of what data they have originally been trained on.

4.3.2 Experiments to Mitigate Bias

We will now experiment with various hypothesized ways to reduce the bias of a classifier when testing on a completely new text subdomain. We will compare the drops in accuracy with our initial drops and determine the best approach to train a classifier for our needs. We will do this using four approaches listed: (1) using subsets of features; (2) amplifying relevant features; (3) incorporating rule-based features along with amplified features; and finally, (4) increasing the relevance of common words among datasets.

We will explore the advantages and trade-offs of all the proposed methods and discuss the motivation behind using each of them. Finally, we will select the best proposed bias reduction method for our problem settings.

Using subsets of features

From the initial parts-of-speech breakdown of the datasets discussed in the previous chapter, we can see that there are four major parts-of-speech in each of the datasets: Nouns, Verbs, Adjectives and Adverbs. For this experiment, we can consider each of these to be separate features. The goal is to determine which subsets of these features have the greatest impact on determining the classification accuracy.

From the way the English language is structured, we can hypothesize that the most relevant parts-of-speech are adjectives and adverbs. For instance, in both the sentences “This is a great book” and “He swims well”, the key lexicon that specifies the sentiment behind the sentence is either an adjective or an adverb.

With this method, we create various different feature subsets: one that includes only adjectives in a sentence, one that has only adverbs, one with both adjectives as well as adverbs, and finally one with only nouns and verbs.

We can see that the highest accuracy is obtained when the features are a subset of adjectives and adverbs(Tables 4.29, 4.21). In addition to that, when we use such a subset of features, the average drop rate when cross-training the classifier reduces(Tables 4.30, 4.22). This is due to the fact that when different text based datasets have different subjects, the presence of the nouns and verbs add unnecessary noise to the classifier as an unseen data will not have the same context behind the sentiment. On the other hand we can see that using a subset of nouns and verbs provide the least accuracy. (Tables 4.23, 4.31)

We also note that using just adjectives provides a similar accuracy(Table 4.25, 4.17) to using a dataset with all features included. This would suggest the correctness of our hypothesis that they have the highest impact on determining the polarity of a sentiment. Using a subset of adverbs provides an overall reduced accuracy(Tables 4.27, 4.19). This may be due to the fact that the breakdown of adverbs in the overall sentence is lower than that of other features.

TABLE 4.17: Social Media: Feature subset - Adjectives

Algorithm	**Reddit(R)**	**Twitter(T)**	**Cross-Training R(data)T(model)**	**Cross-Training T(data)R(model)**
Decision Trees	52.83069977	47.60287287	46.94356659	44.01997094
SVM	52.33408578	34.03859139	33.82392777	46.10709828
Random Forest	63.41309255	58.59712304	58.41083521	54.08729103
Naive Bayes	48.44243792	42.38914694	47.62076749	41.45607825
XGBoost	63.85553047	57.6947474	59.54853273	53.83356183

TABLE 4.18: Drop Rates of Social Media Feature subset - Adjectives

Algorithm	R(data)T(model)	T(data)R(model)
Decision Trees	0.1114339429	0.07526650619
SVM	0.3536922015	-0.3545536519
Random Forest	0.07888366795	0.07696336907
Naive Bayes	0.01696178938	0.02201197142
XGBoost	0.06744909502	0.06692438644
Average	0.1256841394	-0.02267748376

TABLE 4.19: Social Media: Feature subset - Adverbs

Algorithm	Reddit(R)	Twitter(T)	Cross-Training R(data)T(model)	Cross-Training T(data)R(model)
Decision Trees	54.35665914	50.15039594	50.8261851	44.10795768
SVM	46.21218962	33.55568742	25.10158014	44.82617503
Random Forest	61.39954853	58.21857543	59.72009029	51.04663297
Naive Bayes	55.20541761	44.87323771	53.751693	44.67270979
XGBoost	61.64334086	58.02623233	60.80361174	51.30854699

TABLE 4.20: Drop Rates of Social Media Feature subset - Adverbs

Algorithm	R(data)T(model)	T(data)R(model)
Decision Trees	0.06495016611	0.120486352
SVM	0.4568190699	-0.3358741387
Random Forest	0.02735294118	0.1231899339
Naive Bayes	0.02633300622	0.00446876425
XGBoost	0.01362238172	0.1157698004
Average	0.117815513	0.005608142374

TABLE 4.21: Social Media: Feature subset - Adjectives and Adverbs

Algorithm	Reddit(R)	Twitter(T)	Cross-Training R(data)T(model)	Cross-Training T(data)R(model)
Decision Trees	51.00677201	47.50260891	45.38148984	42.48736469
SVM	51.79232506	44.25733052	47.57562077	45.72650447
Random Forest	62.68171558	59.3235252	57.95936795	53.87653209
Naive Bayes	49.98645598	42.31957603	42.20316027	44.70544904
XGBoost	63.09706546	58.46411983	60.25282167	53.92973338

TABLE 4.22: Drop Rates of Social Media Feature subset - Adjectives and Adverbs

Algorithm	R(data)T(model)	T(data)R(model)
Decision Trees	0.1102850062	0.1055782899
SVM	0.08141562064	-0.03319617181
Random Forest	0.07533851916	0.09181843267
Naive Bayes	0.1557080925	-0.05637752635
XGBoost	0.04507727533	0.07755844883
Average	0.09356490276	0.03707629465

TABLE 4.23: Social Media: Feature subset - Nouns and Verbs

Algorithm	Reddit(R)	Twitter(T)	Cross-Training R(data)T(model)	Cross-Training T(data)R(model)
Decision Trees	47.73814898	47.30208099	43.04288939	40.06875243
SVM	51.66591422	33.92809642	33.48984199	49.10069366
Random Forest	59.09706546	58.12649629	57.30925508	50.73356387
Naive Bayes	50.69074492	47.25706452	46.86230248	42.47099507
XGBoost	58.95259594	57.71316323	58.42889391	50.38366311

TABLE 4.24: Drop Rates of Social Media Feature subset - Nouns and Verbs

Algorithm	R(data)T(model)	T(data)R(model)
Decision Trees	0.09835445432	0.1529177661
SVM	0.3518000699	-0.4471986008
Random Forest	0.03025210084	0.127186961
Naive Bayes	0.07552547203	0.1012773328
XGBoost	0.0088834431	0.1269987591
Average	0.112963108	0.01223644363

TABLE 4.25: Reviews: Feature subset - Adjectives

Algorithm	Wine(W)	Books(B)	Cross-Training W(data)B(model)	Cross-Training B(data)W(model)
Decision Trees	85.60240178	72.15530847	51.1385886	54.53833466
SVM	75.20380677	49.13315375	43.63778761	59.2350868
Random Forest	89.6809104	81.45625423	50.07179219	59.95775534
Naive Bayes	72.47095679	73.84153505	50.71614197	59.76551839
XGBoost	80.65052035	81.71375681	50.67342265	59.32527204

TABLE 4.26: Drop Rates of Reviews Feature subset - Adjectives

Algorithm	R(data)T(model)	T(data)R(model)
Decision Trees	0.4026033436	0.2441535375
SVM	0.419739645	-0.205603188
Random Forest	0.441667218	0.2639269273
Naive Bayes	0.3001866649	0.1906246485
XGBoost	0.3716913117	0.2739867269
Average	0.3871776366	0.1534177304

TABLE 4.27: Reviews: Feature subset - Adverbs

Algorithm	Wine(W)	Books(B)	Cross-Training W(data)B(model)	Cross-Training B(data)W(model)
Decision Trees	85.56205575	71.87051299	51.54086222	52.89957399
SVM	77.32197316	53.28048795	44.62270532	59.2623797
Random Forest	90.42849853	83.21842627	48.22062157	58.30831484
Naive Bayes	74.33755384	73.73355009	46.70764557	59.14134162
XGBoost	83.61951324	85.00789121	51.02823035	59.2101672

TABLE 4.28: Drop Rates of Reviews Feature subset - Adverbs

Algorithm	W(data)B(model)	B(data)W(model)
Decision Trees	0.3976201043	0.2639599775
SVM	0.4228974831	-0.1122717149
Random Forest	0.46675415	0.2993340843
Naive Bayes	0.3716816985	0.1979045964
XGBoost	0.3897569075	0.3034744615
Average	0.4097420687	0.190480281

TABLE 4.29: Reviews: Feature subset - Adjectives and Adverbs

Algorithm	Wine(W)	Books(B)	Cross-Training W(data)B(model)	Cross-Training B(data)W(model)
Decision Trees	85.55849581	73.3989154	50.88227267	54.75311792
SVM	75.13735449	47.28316977	52.89364075	57.97842674
Random Forest	89.76753569	82.5942495	47.78749511	61.97149672
Naive Bayes	71.07427229	69.79031933	48.08297042	57.10149399
XGBoost	80.85937036	83.00720295	50.87633943	61.06964436

TABLE 4.30: Drop Rates of Reviews Feature subset - Adjectives and Adverbs

Algorithm	R(data)T(model)	T(data)R(model)
Decision Trees	0.4052925757	0.2540336922
SVM	0.2960406829	-0.226195854
Random Forest	0.467652813	0.2496875135
Naive Bayes	0.3234827615	0.1818135446
XGBoost	0.3708046551	0.2642849995
Average	0.3726546976	0.1447247792

TABLE 4.31: Reviews: Feature subset - Nouns and Verbs

Algorithm	Wine(W)	Books(B)	Cross-Training W(data)B(model)	Cross-Training B(data)W(model)
Decision Trees	85.41728471	70.39906967	49.2185924	51.61799433
SVM	77.40385186	53.30303426	44.51828031	59.14727486
Random Forest	90.51156388	83.27894531	47.7056164	57.30678407
Naive Bayes	56.46087165	50.34309057	46.0773756	49.3098139
XGBoost	83.43558282	84.99839803	50.77547436	60.29713662

TABLE 4.32: Drop Rates of Reviews Feature subset - Nouns and Verbs

Algorithm	R(data)T(model)	T(data)R(model)
Decision Trees	0.4237865022	0.2667801638
SVM	0.4248570421	-0.1096417997
Random Forest	0.4729334644	0.3118694785
Naive Bayes	0.183906053	0.02052469681
XGBoost	0.3914410058	0.2906085524
Average	0.3793848135	0.1560282184

Amplifying relevant features

We can see from the results of the previous experiment that adjectives and adverbs have a higher impact in determining the polarity of a sentiment. Because of this, we propose a method to increase the impact that these features have when they appear in the training dataset.

We approach this problem by using NLTK's list of Parts-of-Speech tags. Parts of Speech is a predefined set of tags that can be used to identify the context of a word and determine the part of speech that it belongs to such as nouns, adjectives, verbs, adverbs, etc. Parts of Speech provide an additional context to a classifier and provides necessary weightage to words that are more relevant to the sentiment classification problem statement.

We modify our initial machine learning algorithm by implementing a one-hot encoder to encode the POS tags. We only include the tags for adjectives and adverbs as those are the features that we would like to have more weightage for. These tags are then concatenated with the initial 100-dimensional word vectors for each WORD2VEC model. We use k-fold cross validation to test the relevance of using POS tags in addition to the WORD2VEC model.

We find in every case that amplifying the adjectives and adverbs feature subsets assists in reducing the overall drop rates of the cross-classification(Tables 4.34,4.36), however reduces the accuracy of the classifier(Tables 4.33, 4.35) and . This method has a lower processing time than the previous method as subsets do not need to be formed. At the same time however, the vocabulary size is larger in this case and so the training can take longer.

Incorporating rule-based features

In this experiment, we use the amplification of feature subsets technique discussed in the previous section and combine them with a hybrid machine

TABLE 4.33: Parts-Of-Speech (POS) Feature Amplification: Social Media

Algorithm	Reddit(R)	Twitter(T)	Cross-Training R(data)T(model)	Cross-Training T(data)R(model)
Decision Trees	51.72911964	49.55085838	46.9255079	45.36432649
SVM	56.61399549	47.11178408	49.76975169	53.62484909
Random Forest	62.70880361	61.57230259	62.00451467	55.9472898
Naive Bayes	54.21218962	48.64029793	47.79232506	49.38716212
XGBoost	63.94582393	62.29665855	62.90744921	56.43223998

TABLE 4.34: Drop Rates of POS Feature Amplification: Social Media

Algorithm	R(data)T(model)	T(data)R(model)
Decision Trees	0.09286088323	0.08448959366
SVM	0.1208931419	-0.1382470466
Random Forest	0.01123110151	0.09135621947
Naive Bayes	0.1184210526	-0.01535484414
XGBoost	0.01623835075	0.0941369683
Average	0.07192890601	0.02327617815

TABLE 4.35: Parts-Of-Speech (POS) Feature Amplification: Reviews

Algorithm	Wine(W)	Books(B)	Cross-Training W(data)B(model)	Cross-Training B(data)W(model)
Decision Trees	82.2964009	56.36102574	49.67545182	51.15520167
SVM	67.11917504	51.48983636	54.88958242	53.52019081
Random Forest	86.79142291	66.61484971	49.48677481	54.29744515
Naive Bayes	61.59176941	56.63870133	53.9663704	52.05705403
XGBoost	76.35604182	68.42923426	49.0275421	53.95569057

TABLE 4.36: Drop Rates of POS Feature Amplification: Social Media

Algorithm	R(data)T(model)	T(data)R(model)
Decision Trees	0.3963836659	0.0923656729
SVM	0.1822071355	-0.03943214031
Random Forest	0.4298195242	0.1849047865
Naive Bayes	0.1238054871	0.08089252043
XGBoost	0.3579088055	0.2115111157
Average	0.2980249236	0.1060483911

learning approach. Adding VADER positive and negative scores gives an added boost to sentiment analysis algorithms, especially in terms of reducing the bias of the classifier when testing on a completely new domain.

In this approach, classifiers are trained in a similar method to the 'Amplifying Relevant Features' method. In addition to the word vectors and one-hot encoded vectors, VADER positive and negative scores are appended prior to training.

We can see that this method performs really well as compared to the other methods of reducing bias. It provides a considerably high accuracy for classification (Tables 4.37, 4.39) while reducing the bias during cross-classification (Tables 4.38, 4.40)

TABLE 4.37: Incorporating rule based features: Social Media

Algorithm	**Reddit(R)**	**Twitter(T)**	**Cross-Training R(data)T(model)**	**Cross-Training T(data)R(model)**
Decision Trees	51.72911964	49.55085838	46.9255079	45.36432649
SVM	56.61399549	47.11178408	49.76975169	53.62484909
Random Forest	62.70880361	61.57230259	62.00451467	55.9472898
Naive Bayes	54.21218962	48.64029793	47.79232506	49.38716212
XGBoost	63.94582393	62.29665855	62.90744921	56.43223998

TABLE 4.38: Drop Rates of Incorporating rule based features: Social Media

Algorithm	**R(data)T(model)**	**T(data)R(model)**
Decision Trees	0.09286088323	0.08448959366
SVM	0.1208931419	-0.1382470466
Random Forest	0.01123110151	0.09135621947
Naive Bayes	0.1184210526	-0.01535484414
XGBoost	0.01623835075	0.0941369683
Average	0.07192890601	0.02327617815

Among the proposed methods, this approach has proved to be the best one to reduce the presence of bias during cross-classification. In order to test our method further, we introduce a new testing dataset to check the bias trends. For this purpose, we will combine our social media datasets and then

TABLE 4.39: Incorporating rule based features: Reviews

Algorithm	Wine(W)	Books(B)	Cross-Training W(data)B(model)	Cross-Training B(data)W(model)
Decision Trees	85.55612251	73.13903953	50.55475786	52.25403757
SVM	77.36706578	56.5188499	44.86003489	61.74840693
Random Forest	90.47596445	83.65985926	50.55357122	57.37442299
Naive Bayes	74.41112601	75.5443747	47.15501181	60.48700027
XGBoost	83.67053909	85.50391	52.14249267	59.67889309

TABLE 4.40: Drop Rates of Incorporating rule based features: Reviews

Algorithm	W(data)B(model)	B(data)W(model)
Decision Trees	0.4091041485	0.2855520402
SVM	0.4201662628	-0.09252766172
Random Forest	0.4412486065	0.3141941249
Naive Bayes	0.3662908447	0.1993182746
XGBoost	0.3768117997	0.3020331691
Average	0.4027243324	0.2017139894

use this to train the *Word2Vec* model. After this, we will test this on a completely new domain which is our product reviews domain for our set of wine reviews. Table 4.41 Has the accuracy scores for this experiment. The table 4.42 lists the average bias when performing this experiment. We compare the trends with bias values from Table **??**

We can see that using this method, we obtain a reduced bias even in cases where the models are trained from a completely new domain. In this case, the social media domain.

TABLE 4.41: Incorporating rule based features: Reviews with Social Media

Algorithm	Wine(W)	Cross-Training
Decision Trees	81.4265880314699	41.3131444981073
SVM	66.7192747208411	43.542855786688186
Random Forest	86.00230209680673	45.216029239002744
Naive Bayes	62.93030817244366	57.445621862799776
XGBoost	75.44232298180869	47.776815274530975

TABLE 4.42: Drop Rates of Incorporating rule based features: Reviews

Algorithm	Drop in Accuracy
Decision Trees	0.4466619013
SVM	0.6252618536
Random Forest	0.3049689961
Naive Bayes	0.2415559207
XGBoost	0.1817108505
Bias	0.3600319044

Increasing the weightage of common words

In this method, we improve the relevance of the words that lead up to the same sentiment on both datasets. For instance, we find that the word “good" leads to a positive sentiment in both of the social media datasets and the word “ugly" leads to a negative sentiment.

We use this new subset of words, both for the positive as well as the negative sentiment and aim to reduce the bias of the classifier by assigning more weight to these common words.

Initially, the sentiment classifier was trained by converting the input text of the testing data into a WORD2VEC model. We perform this current set of experiments by modifying the testing data with which all the classifiers have been trained. We take our set of common words for both the positive as well as the negative sentiments. We then proceed to repeat and shuffle these words throughout the training data on all the sub-domains against their relevant sentiment classes. For our previous example, if we find that the word “good" implies a positive sentiment across both the social media datasets, then we increase the frequency of the lexicon in our WORD2VEC model by replacing the word with a different word of a similar POS tag. By doing this, we increase the overall Jaccard similarity of both the social media datasets and make it so that the models are more similar to begin with and that the bias that comes with cross-classification is reduced.

We then proceed to test the accuracy of the classifiers in identifying sentiments and also measure the drop in accuracy, or bias, when cross-classifying in all four subdomains.

TABLE 4.43: Increasing the weightage of common words: Social Media

Algorithm	Reddit(R)	Twitter(T)	Cross-Training R(data)T(model)	Cross-Training T(data)R(model)
Decision Trees	51.72911964	49.55085838	46.9255079	45.36432649
SVM	56.61399549	47.11178408	49.76975169	53.62484909
Random Forest	62.70880361	61.57230259	62.00451467	55.9472898
Naive Bayes	54.21218962	48.64029793	47.79232506	49.38716212
XGBoost	63.94582393	62.29665855	62.90744921	56.43223998

TABLE 4.44: Drop Rates of Increasing the weightage of common words: Social Media

Algorithm	R(data)T(model)	T(data)R(model)
Decision Trees	0.09286088323	0.08448959366
SVM	0.1208931419	-0.1382470466
Random Forest	0.01123110151	0.09135621947
Naive Bayes	0.1184210526	-0.01535484414
XGBoost	0.01623835075	0.0941369683
Average	0.07192890601	0.02327617815

TABLE 4.45: Increasing the weightage of common words: Reviews

Algorithm	Wine(W)	Books(B)	Cross-Training W(data)B(model)	Cross-Training B(data)W(model)
Decision Trees	51.72911964	49.55085838	46.9255079	45.36432649
SVM	56.61399549	47.11178408	49.76975169	53.62484909
Random Forest	62.70880361	61.57230259	62.00451467	55.9472898
Naive Bayes	54.21218962	48.64029793	47.79232506	49.38716212
XGBoost	63.94582393	62.29665855	62.90744921	56.43223998

We can see from our results that this method does not perform as well as the previously mentioned methods. This method provides the highest bias which is undesirable(Tables 4.44, 4.46) This is due to a couple of reasons.

TABLE 4.46: Drop Rates of Increasing the weightage of common words: Reviews

Algorithm	W(data)B(model)	B(data)W(model)
Decision Trees	0.09286088323	0.08448959366
SVM	0.1208931419	-0.1382470466
Random Forest	0.01123110151	0.09135621947
Naive Bayes	0.1184210526	-0.01535484414
XGBoost	0.01623835075	0.0941369683
Average	0.07192890601	0.02327617815

Firstly, while this method works to make the WORD2VEC models more consistent, in doing so we lose the legibility of the text and so the classifier is not able to learn the text as properly. Future work can be done in increasing the similarities of datasets while retaining the context of the data instance.

4.3.3 Evaluating and Comparing Bias Reduction methods

In the previous section, we have proposed various methods to mitigate classification bias in sentiment classifiers. In this section, we aim to evaluate these proposed bias reduction methodologies and their drawbacks in order to determine the best one for our settings.

TABLE 4.47: Comparing Bias Reduction Methods

Datasets	Original	Review Subsets	Amp. relevant features	Inc. rule based features	Increasing common word weights
R(data)T(model)	0.1007277701	0.09356490276	0.07192890601	0.07192890601	0.1318049096
T(data)R(model)	0.02745325477	0.03707629465	0.02327617815	0.02327617815	0.166325439
W(data)B(model)	0.3804454019	0.3726546976	0.2980249236	0.2980249236	0.3798952909
B(data)W(model)	0.1573053839	0.1447247792	0.1060483911	0.09604839105	0.1856189347

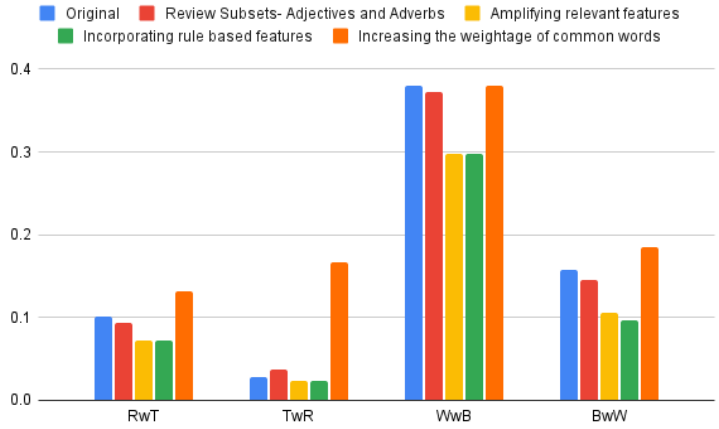

FIGURE 4.3: Comparison of Bias Reduction approaches

Figure 4.3 compares the bias after cross-classification for all our datasets as detailed in Table 4.47. The aim is to have the least possible bias while cross-classifying the sentiment analysis data while still having considerably high accuracy in terms of sentiment analysis.

We can see that using the *Incorporating rule based features* approach discussed in section 4.2 provides the least bias in all our training datasets. *Amplifying relevant features* method provides comparable bias but with a reduced accuracy. Overall, we see that all proposed methods other than the *Increasing Similarities* method have a positive impact in bias reduction.

The above observations are consistent in all our datasets obtained from both user domains. From our proposed methods, we conclude that the *Incorporating rule based features* method provides the best trade-off between mitigating bias and maintaining accuracy.

Chapter 5

Conclusions and Future Work

Through this thesis, we have explored the various issues that affect the performance of sentiment classifiers. We have proposed various methods to reduce this classification bias and compared results. While there is still a lot of research to be done in order to perfect the process of removing bias, our work can act as a good starting point. There are various applications of having an unbiased classifier irrespective of the training data. In this chapter, we will discuss the applications of such a classifier and the future work in this field.

5.1 Observations

In this section, we will summarize the various conclusions that we have arrived at through various experiments.

5.1.1 Data

In this thesis, we work with user-generated text from two main domains - Social Media and Product Reviews. When working with English text, we can see that even when text is obtained from different sources, the basic structure of the text in terms of the number of adjectives, nouns, verbs, adverbs, and other parts of speech remains consistent.

5.1.2 Sentiment Classification

We explore sentiment classification of text in the form of sentences. We check the results of sentiment classification using Rule-Based, Machine Learning and Hybrid methods.

- **Rule-Based Methods**
  - Rule-Based methods provide a lower accuracy for sentiment classification.
  - Sentiment classification results remain comparable despite the size and source of the datasets.
- **Machine Learning Methods**
  - Accuracies are higher than those of rule-based methods.
  - Product reviews datasets have a higher accuracy as compared to rule based methods. This is due to larger size of training data.
  - XGBoost and Random Forrest classifiers outperform other classifiers in terms of accuracies.
- **Hybrid Methods**
  - This method has the highest accuracy among the explored sentiment classification methods.
  - An advantage of using this method is the reduced dependency on training data. A trade-off however is needing larger training data and and increased number of computations.

5.1.3 Cross-Domain Classification

Cross-domain classification is the method of testing the accuracy of a classifier on data obtained from an unseen domain. We use the cross-classification algorithm on each dataset and observe the trends in bias.

- Machine Learning approaches provide a bias of 2-10% for Social Media Data and 15-40% for Product Reviews Data.
- Hybrid Approaches provide a bias of 3-10% for Social Media data and 20-40% for Product Reviews data.
- Social Media datasets are more similar to each other as compared to Product Reviews datasets as they provide less bias during cross-domain classification.

5.1.4 Bias-Reduction Methods

We propose and test out four methods for reduction in bias. The results and observations for each of those methods are listed as follows.

- **Using Feature Subsets**
  - Adjectives and adverbs have a high impact in both datasets in predicting the correct sentiment.
  - Adverbs have a positive impact but have a higher bias in some cases. This is due to the distribution of adverbs in data.
  - Nouns+Verbs have a considerable impact in predicting sentiments but only in the product reviews datasets. While their accuracies in sentiment analysis are high, the bias produced during cross-classification is higher while using this subset in both cases.
- **Amplifying Relevant Features**
  - This method provides a reduced bias than using the Adjectives+Adverbs subsets.
  - The accuracy for this approach is lower than that of the previous method.

  - The computation time is lower as feature subsets need not be calculated.

- **Incorporating rule-based features**

  - This method outperforms the previous two in terms of bias reduction.
  - It provides the advantages of a reduced bias through amplification of relevant feature subsets. At the same time, it also provides the improved accuracies from using a Hybrid approach.

- **Increasing Similarities**

  - This method does not perform as well as the previous methids for bias reduction.
  - Even though the model is more similar, the legibility is reduced and so the classifiers do not perform as well.

5.2 Limitations

In this section, we will discuss the various limitations in the application for our thesis caused by our scope and assumptions.

- **Limitations of Input Data**

  - For our scope, we have used comments from social media websites as well as reviews from product review websites. While *Word2Vec* is suitable in this instance, we would require different machine learning approaches such as *Doc2Vec* where we need to process larger user inputs.
  - We have only worked with text written in the English language for our approach. As a result, the assumptions made have been

proven just for English and cannot be generalized to other languages without prior testing.

- **Limitations of Sentiment Analysis approaches**
  - For our approach, we have considered classification bias only for the *Word2Vec* vectorizer along with a list of different classifiers. The results obtained have been tested only for this approach. Experiments need to be done to test if our hypotheses holds up for other approaches.
  - For our hybrid method, we have only used lexicon data obtained from the VADER approach as it is a suitable measure when cross-classifying.
- **Limitations of Bias**
  - The bias trends observed have mainly been tested on datasets obtained from the same general domain. Our methods may not hold up as well with unseen data obtained from a different domain such as news forums or online journals.
  - While the bias reduction experiments hold true for the datasets we have tested, further work is needed to test the stability of our model.

5.3 Applications

There are various applications of our findings on sentiment classification and other cross-domain classification problems. These are listed in this section.

5.3.1 Detection of hate speech in new social media sites

One of the major issues with social media sites is the issue of detecting and controlling hate speech. Various new social media sites emerge on a regular basis. For newer sites, it is difficult to analyze the sentiments behind user generated text due to the lack of substantial training data. Through an unbiased classifier trained on a different social media site, this issue can be overcome. Such a classifier would make for a considerably accurate sentiment classifier till additional training data can be obtained from the new social media site.

5.3.2 Detection of popularity in newly launched products

For product reviews, sentiment classification provides an important application to judge the general consensus and reception regarding a newly launched product. Through experiments we can see that a sentiment classifier is highly biased when testing on a brand new product review dataset. If we can successfully reduce this bias, we would be able to understand the sentiments of users irrespective of the fact that we do not have enough training data.

5.3.3 Comparing the behaviours of users

While there are many applications of reducing bias, we can also use the presence of bias to make various observations. A large amount of bias would suggest a large amount of behavioral bias in the ways that users interact in various domains. This could have applications in determining the most similar subsets of users and compare the likeness of various different social media platforms.

5.3.4 Applications beyond sentiment analysis

In this project, we have experimented with the detection and mitigation of classification bias in the case of sentiment analyzers. While this has many applications the same techniques of analyzing and mitigating bias can be used for various different classifiers as well. Classification bias based on training data exists in many different applications and so devising methods to overcome the limitations of our training data is important.

Bibliography

[1] Giovanni Maria Farinella Alessandro Ortis and Sebastiano Battiato. "Survey on Visual Sentiment Analysis". In: *Computer Vision and Pattern Recognition* (May 2020), p. 30. URL: https://arxiv.org/abs/2004.11639.

[2] Shivakumar Vaithyanathan Bo Pang Lillian Lee. "Thumbs up? Sentiment Classification using Machine Learning Techniques". In: *Conf. on Empirical Methods in Natural Language Processing (EMNLP)* (May 2002). URL: https://arxiv.org/abs/cs/0205070.

[3] Dalibor Bužić. "Sentiment Analysis of Text Documents". In: *Central European Conference on Information and Intelligent Systems* (Apr. 2019), pp. 215–221. URL: https://search-proquest-com.libezproxy2.syr.edu/docview/2366654646/abstract/96B44A2143344F84PQ/1?accountid=14214.

[4] You-De Tseng Chien Chin Chen. "Quality evaluation of product reviews using an information quality framework". In: *Decision Support Systems* 50.4 (Mar. 2010), pp. 755–768. URL: https://www-sciencedirect-com.libezproxy2.syr.edu/science/article/pii/S0167923610001478?via%3Dihub.

[5] Fei Song Chris Nichols. "Improving Sentiment Analysis with Parts of Speech weighing". In: *IEEE* (June 2009). URL: https://ieeexplore-ieee-org.libezproxy2.syr.edu/stamp/stamp.jsp?tp=&arnumber=5212278.

[6] David Weir Danushka Bollegala and John Carroll. "Using Multiple Sources to Construct a Sentiment Sensitive Thesaurus for Cross-Domain Sentiment Classification". In: *49th Annual Meeting of the Association for Computational Linguistics* (June 2011), 132–141. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.207.5144&rep=rep1&type=pdf.

[7] DongwenZhang. "Chinese comments sentiment classification based on word2vec and SVM". In: *Expert Systems with Applications* 42.4 (2015), pp. 1857–1863. URL: https://www-sciencedirect-com.libezproxy2.syr.edu/science/article/pii/S0957417414005508?via%3Dihub.

[8] Pavlos Fafalios Eirini Ntoutsi and Steffen Staab. "Bias in data-driven artificial intelligence systems—An introductory survey". In: *WIREs* 10.12 (Dec. 2020). URL: https://onlinelibrary.wiley.com/doi/full/10.1002/widm.1356.

[9] ANASTASIA GIACHANOU and FABIO CRESTANI. "Like It or Not: A Survey of Twitter Sentiment Analysis Methods". In: *ACM Computing Surveys* (June 2016), p. 41. URL: https://dl-acm-org.libezproxy2.syr.edu/doi/pdf/10.1145/2938640.

[10] Xiyue Guo Xinhui Tu Tingting He Guangyou Zhou Yin Zhou. "Cross-domain sentiment classification via topical correspondence transfer". In: *Neurocomputing* 159 (June 2015), pp. 298–305. URL: https://www.sciencedirect.com/science/article/abs/pii/S0925231214016701.

[11] Erik Cambria Iti Chaturvedi. "Distinguishing between facts and opinions for sentiment analysis: Survey and challenges". In: *IEEE* (Nov. 2018), pp. 65–77. URL: https://www-sciencedirect-com.libezproxy2.syr.edu/science/article/pii/S1566253517303901?via%3Dihub.

[12] Ming Zhou Xiaohua Liu Tiejun Zhao Long Jiang Mo Yu. "Target-dependent Twitter Sentiment Classification". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (June 2011), 151–160. URL: https://www.aclweb.org/anthology/P11-1016/.

[13] Carlos Guestrin Tianqi Chen. "XGBoost: A Scalable Tree Boosting System". In: *ACM* (2016). URL: https://dl.acm.org/doi/abs/10.1145/2939672.2939785.

[14] Tong He Tianqi Chen. "xgboost: eXtreme Gradient Boosting". In: (2021). URL: https://mran.microsoft.com/web/packages/xgboost/vignettes/xgboost.pdf.

[15] Hoda Korashy Walaa Medhat Ahmed Hassan Hoda Korashy Walaa Medhat Ahmed Hassan. "Sentiment analysis algorithms and applications: A survey". In: *AIN SHAMS ENGINEERING JOURNAL* 5.4 (Dec. 2014), 1093 – 1113. URL: https://doaj.org/article/4aee8aa7872040a3b53a333cd5303583.

[16] ByungJun Lee Yili Wang KyungTae Kim and Hee Yong Youn. "Word clustering based on POS feature for efcient twitter sentiment analysis". In: *Human-Centric computing and Information Sciences* (2018). URL: https://hcis-journal.springeropen.com/track/pdf/10.1186/s13673-018-0140-y.pdf.

[17] Lin Yue. "A survey of sentiment analysis in social media". In: *Knowledge and Information Systems* (Apr. 2018). URL: https://link-springer-com.libezproxy2.syr.edu/content/pdf/10.1007/s10115-018-1236-4.pdf.

[18] Tsung-YingLi Yung-MingLi. "Deriving market intelligence from microblogs". In: *Decision Support Systems* 55.1 (Apr. 2013), pp. 206–217. URL: https://www-sciencedirect-com.libezproxy2.syr.edu/science/article/pii/S0167923613000511?via%3Dihub.

[19] Paola Zola. “Social Media Cross-Source and Cross-Domain Sentiment-Classi¯cation”. In: *International Journal of Information Technology Decision Making* 18.5 (2018), pp. 1469–1499. URL: https://www-worldscientific-com.libezproxy2.syr.edu/doi/epdf/10.1142/S0219622019500305.

ALPANA DESHPANDE

(586)-272-4334 · alpanadeshpande09@gmail.com · https://www.linkedin.com/in/alpana-deshpande/

EDUCATION

Master of Science, Computer Science | Syracuse University — Aug 2019 - May 2021

Bachelor of Engineering, Computer Engineering | University of Mumbai — Nov 2012 - Aug 2016

WORK EXPERIENCE

Data Scientist, Smarta Nov 2020 - Present

- Developed an analytics tool using Python that analyzed potential student housing markets and predicted profitability of various markets in Upstate New York and Boston.
- Collected data on student housing preferences and trends by creating and distributing surveys to Syracuse University students
- Analyzed available student housing data to create and test market hypotheses and understand the factors contributing to making sales.
- Used Tableau to generate insightful reports on available student housing data to present to investors.

Backend Developer, Smarta Jun 2020 – Nov 2020

- Led a team of three developers to create the backend for the Smarta student housing web application using React, JavaScript and Firebase.
- Incorporated Google Maps API and user chat APIs and redesigned them to fit seamlessly with the UI of the Smarta website.
- Designed the structure of the Smarta database using Firebase and connected to the database from React.
- Collaborated with marketing and social media teams to plan unique student housing features that improved uniqueness and quality.
- Analyzed market trends related to the student housing market in order to keep up with the contemporary student housing market.
- Helped to grow the development team by interviewing and recruiting new student software development interns for the Fall semester.
- Organized and led meetings with third party vendors and stakeholders in order to gather resources for planning and designing the system.

Software Engineer, Larsen & Toubro InfoTech Aug 2016 – Apr2019

- Identified production environment system defects of the Barclays retail banking application to improve the system quality and performance.
- Proposed and implemented independent automation projects for user log retrieval to improve the efficiency in identifying production issues.
- Optimized retrieval of failure logs and daily health check reports for UNIX production servers in order to fast track detection of system failures.
- Regularly coordinated with multiple South African and UK onshore and offshore teams for the overall success of the project.
- Oversaw and implemented various routine and emergency change requests and batch deployments using ServiceNow and ServiceFirst.
- Implemented and deployed code fixes in Java in the production and testing environments in compliance with client SLAs.
- Maintained the databases for various applications by creating and running SQL queries.
- Provided system insights and solutions to stakeholders during unexpected system downtimes and critical failures.
- Extracted logs and provided reports to enable the investigation of potential fraudulent activities.
- Trained new employees in various banking applications through regular knowledge transfer calls, enabling seamless growth of the team.