UNIVERSITY OF CALIFORNIA
Los Angeles

Sentiment Analysis of the Undergraduate STEM Community at UCLA Using the Bruinwalk Course Reviews

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Kaixin Wang

2022

ABSTRACT OF THE THESIS

Sentiment Analysis of the Undergraduate STEM Community at UCLA Using the Bruinwalk Course Reviews

by

Kaixin Wang
Master of Science in Statistics
University of California, Los Angeles, 2022
Professor Mark Handcock, Co-Chair
Professor Robert Gould, Co-Chair

With the rapid advancement of technologies in the recent few decades, more and more students are entering the Science, Technology, Engineering, and Mathematics (STEM) field in college. As one of the leading universities in the world, University of California, Los Angeles (UCLA) has a strong group of undergraduate programs in STEM. With the size of the STEM community expanding rapidly, it is important that we examine the sentiment of the community through some statistical analyses. Among various approaches, sentiment analysis of the course reviews could help us understand the feedback from the student community, which could also provide us with many interesting insights. In this paper, we will look at the methodologies and results from applying the sentiment analysis pipeline on a corpus with around 7000 course reviews collected from the UCLA Bruinwalk website, with the goal of analyzing if students were happy (positive sentiment) or unhappy (negative sentiment) towards the STEM courses they have taken. As we shall see, because the reviews obtained from the Bruinwalk website were not initially annotated, the sentiment analysis pipeline consisted of two main components, review annotation and sentiment analysis modeling. Throughout the study, various data visualization techniques were utilized to help us obtain a better understanding of structure of corpus, including the features and the sentiment annotation. Multiple NLP model architectures, such as CNN+LSTM, Transformer, and the state-of-the-art BERT and DistilBERT architecture, were established and compared to optimize the sentiment prediction performance. The results from the sentiment analysis showed that around 60% of the reviews collected contained a positive sentiment, the sentiment of the reviews was positively associated with the student grades, together with several other interesting findings.

The thesis of Kaixin Wang is approved.

Nicolas Christou

Mathieu Bauchy

Robert Gould, Committee Co-Chair

Mark Handcock, Committee Co-Chair

University of California, Los Angeles

2022

To my dad and mom...
who've been giving me so much love
and support along the way

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Background

Science, Technology, Engineering, and Mathematics (STEM) is a term that describes a group of academic disciplines (Kirtibas Singh and Renuga Devi, 2021), which also has implications for workforce development, national security concerns and immigration policy (Gonzalez and Kuenzi, 2017). In STEM, the word *science* refers to two of the three major branches of science: natural sciences, which includes biology, physics, and chemistry, and formal sciences, which includes mathematics, statistics, and engineering. The third major branch of science, social sciences, such as psychology, sociology, and political science, is instead grouped with humanities and arts, which collectively form the counterpart of STEM – Humanities, Arts, and Social Sciences (HASS). With the rapid advancement of technologies in the recent years, STEM is playing a more important in higher education, reflected in students' studies, research, and future careers.

As one of the leading universities in the world, University of California, Los Angeles (UCLA) together with its professional schools offers more than 5000 courses in a wide range of academic programs. Each academic department at UCLA typically offers two types of courses, the lower-division courses for General

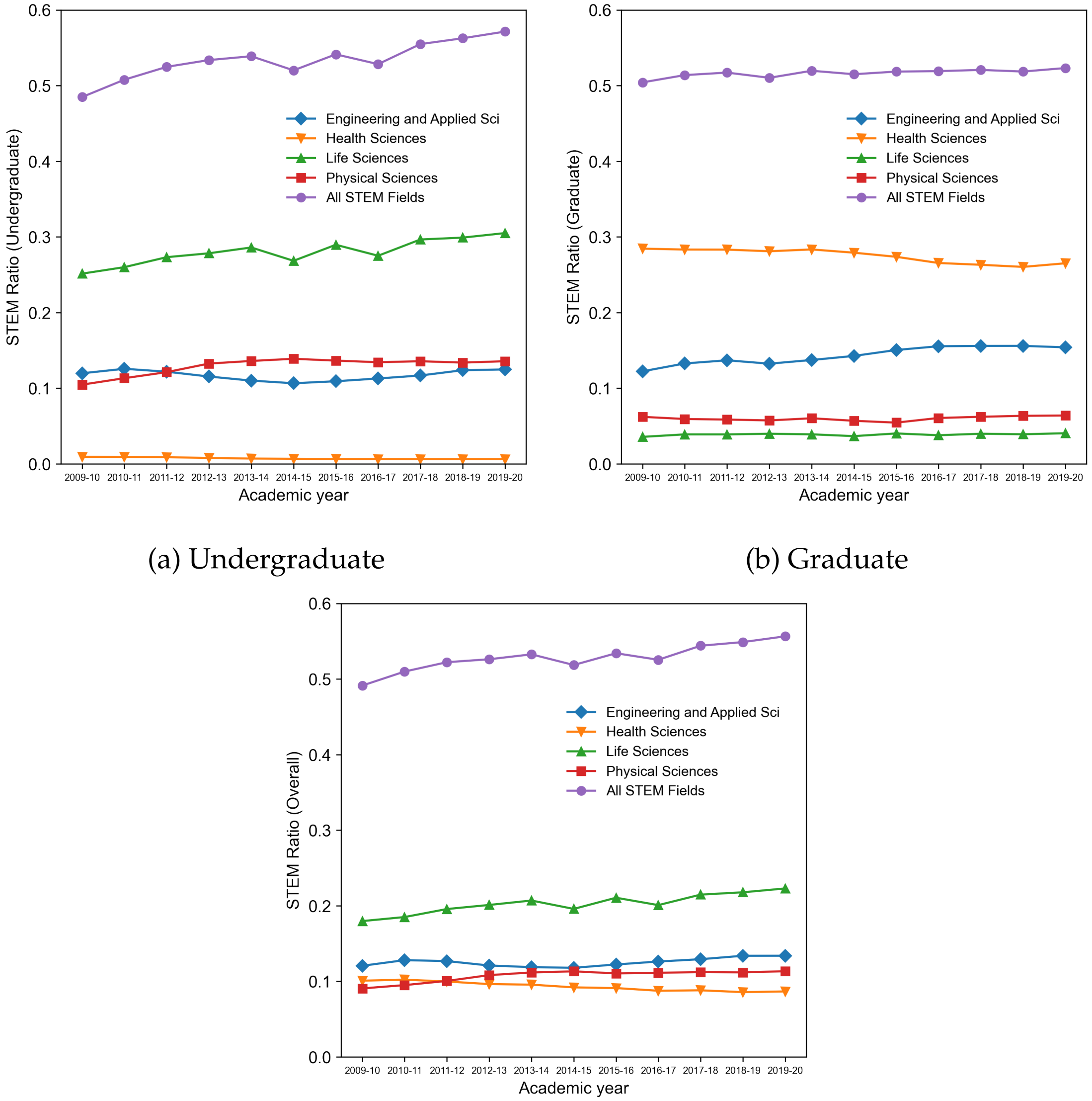
Education (GE) and major preparation purposes, as well as the upper-division, more advanced major coursework. Based on Higher Education Research Institute (HERI) at UCLA, UCLA's STEM is defined to include disciplines in the Life Sciences, Physical Sciences, Engineering, Mathematics, Computer Science, and the Health Sciences. As mentioned in the ″STEM programs and initiatives″ at UCLA, ″current STEM programs include those that strengthen undergraduate science education at UCLA by broadening access to research opportunities as well as better preparing undergraduates, especially those from underrepresented groups, for college success. Programs also promote workforce training and preparing future faculty for successful careers in STEM [1].″

A variety of statistics are recorded and calculated at UCLA each year, such as the admission rate, enrollment rate and the graduation rate. Based on the official facts and figures provided on the UCLA website, the proportion of students who enrolled in a STEM major (shortened as the STEM ratio) has been increasing in each academic year during the past decade [2]. Figure 1.1 shows the STEM ratio in four major STEM fields at UCLA from academic year 2009-2010 to 2019-2020, using the information from multiple annual reports.

From Figure 1.1, we observe the STEM ratio in Life Sciences and Physical Sciences has been increasing in the undergraduate community in the past decade, as well as the overall ratio across all four STEM fields; in the graduate community, the STEM ratio in Engineering and Applied Sciences has been rising, although the ratio in the other three fields is relatively stable; lastly, in the overall community,

[1] `https://ceils.ucla.edu/stem-programs-initiatives/`.

[2] `https://www.ucla.edu/about/facts-and-figures`.



(a) Undergraduate

(b) Graduate

(c) All students

Figure 1.1: STEM ratio of undergraduate students (Figure 1.1a), graduate students (Figure 1.1b) and the overall community (Figure 1.1c) at UCLA from 2009-2020.

the STEM ratio in most of the STEM fields has increased, as well as the overall ratio over all STEM fields. These results show that a larger proportion of UCLA students have chosen to major in STEM in the recent decade, which further reinforces the importance to examine the STEM community at UCLA to sustain or further improve the teaching and studying ecosystem.

1.2 Objectives

As STEM is becoming more and more important in higher education, it is important that we diagnose the healthiness of the STEM community at UCLA to create a more welcoming and diverse community. The objective of this study was to conduct sentiment analysis on course reviews, aiming to see if students had a good experience taking the STEM courses at UCLA. Note that this project focused on analyzing the undergraduate STEM community at UCLA only, since the undergraduates take up 80% of the entire student population, and most of the reviews came from the undergraduate community. In the following Chapter 2, we will look at the candidate dataset that were available for the study, the dataset selection process, the way the dataset was curated, and various visualization results from the exploratory data analysis. The prior work done in sentiment analysis, as well as the state-of-the-art models in NLP will then be introduced (Chapter 3), followed by the methodologies used in implementing the sentiment analysis on UCLA course reviews (Chapter 4). In Chapter 5, we will look at the results obtained from applying the sentiment analysis pipeline from using various visualization techniques. Lastly, we will summarize the results and discuss the possible improvements and future directions in Chapter 6. The flowchart in Figure 1.2 describes the overall workflow of the sentiment analysis pipeline developed in this study.

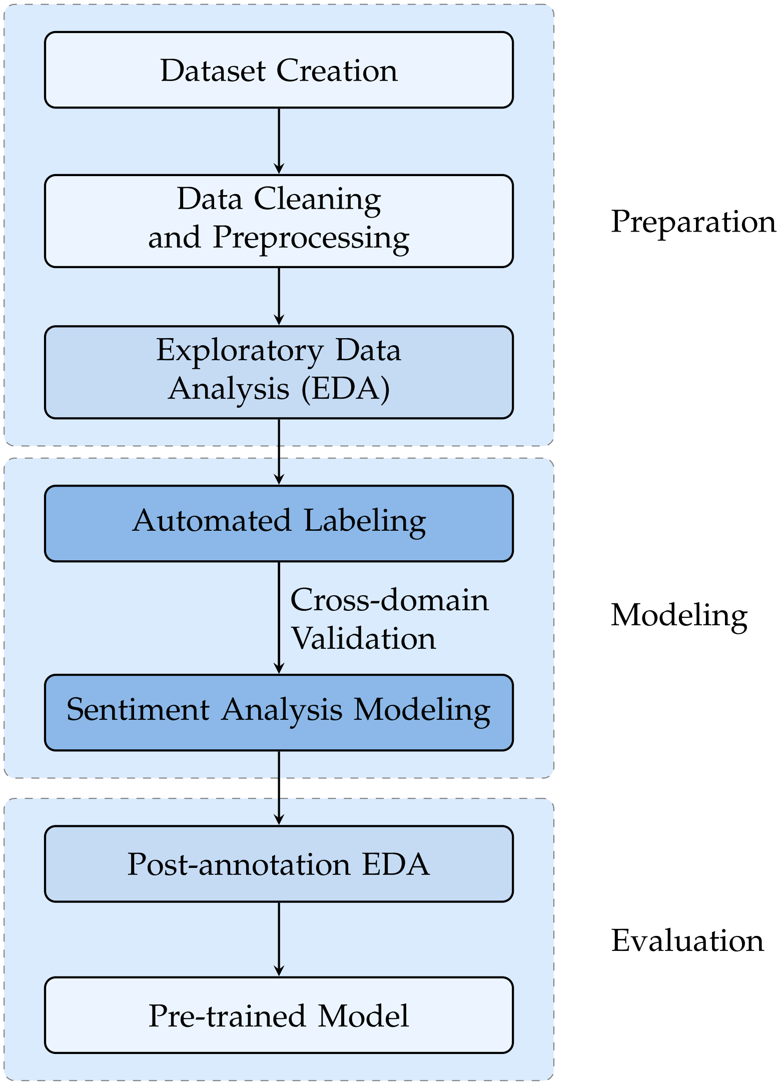


Figure 1.2: Flowchart for the sentiment analysis pipeline on the undergraduate STEM community at UCLA.

CHAPTER 2

Dataset

2.1 Choosing the Dataset

Choosing the right dataset is essential to any type of data analysis. Three major data sources were considered for collecting the course reviews: the official UCLA course evaluation database [1], the RateMyProfessors [2] review website, and the Bruinwalk review website [3].

The official UCLA course evaluation system is built to help improve the existing courses by sending out online course evaluation forms to students at the end of each term. Most of the surveys are designed by UCLA's Center for the Advancement of Teaching (CAT) and they typically contain a set of multiple-choice questions and short-answer questions. Students can provide their feedback on the content of the courses, as well as their feedback to the instructors and the teaching assistants. This database contains quite formal and rich student reviews, but because the evaluation database is highly confidential and is the internal use at UCLA CAT and the College Faculty Executive Committee, we don't have direct

[1] `https://teaching.ucla.edu/eip/`.

[2] `https://www.ratemyprofessors.com`.

[3] `https://www.Bruinwalk.com`.

access to it.

The RateMyProfessors website (RMP) is a public online review website, where students can write reviews for courses and colleges or universities in the United States, Canada, and United Kingdom. RMP is currently the largest online review website for professor ratings, which includes information for over 8000 schools, 1.7 million professors, totaling over 19 million ratings. One drawback of RMP, however, is that the reviews are fully categorized based on the instructor, meaning it's very difficult to locate reviews based on the course numbers. As we shall see the next, in the Bruinwalk review database, each review corresponds to a course and an instructor, which makes it much easier to look up the reviews.

The Bruinwalk.com website (shortened as Bruinwalk) is a review website operated by the Daily Bruin (student media at UCLA) and founded in 1998. Bruinwalk is built for students, run by students, and it provides anonymous professor reviews and reviews of apartments near UCLA. Its goal is to supplement the Daily Bruin's mission of creating a comprehensive record of life at UCLA by allowing students to share reviews and advice about academic study and housing. One of the biggest advantages the Bruinwalk dataset has, is that all reviews are for UCLA courses only, and the reviews are written by students verified to have studied at UCLA. Furthermore, each review corresponds to a course and an instructor, which means visitors to the website can search for reviews both by the course and by the instructor. Some Bruinwalk reviews also feature the grade distribution of the class in the past, which are directly synchronized from the UCLA registrar.

By comparison, the Bruinwalk database is the best fit to this study – the database is designed for UCLA courses only, and it contains a rich set of features which supplements the analysis. In fact, many institutions also have their own course review website, and some have been used for similar type of research, referred to as the institution-based course review analysis. Since no such prior work has been done on the Bruinwalk dataset, it will be interesting to see the findings through the analysis.

2.2 Creating the Dataset

Among all the steps in creating the dataset, the most important one was to decide from which STEM courses the reviews should be selected from, as well as deciding the features of interests. Based on the four major STEM fields shown in Figure 1.1, 8 major/specialization were selected: Biology from Life Sciences; Chemistry, Mathematics, Physics, Program in Computing, and Statistics from Physical Sciences; Computer Science and Electrical Engineering from Engineering and Applied Sciences. The assumption was that courses from these major/specialization could represent the entire STEM community, considering that they took up a large proportion of the STEM population in the recent years. To refine the selection of the reviews, most of the lower-division preparation courses and upper-division major courses in each major/specialization were selected. Table A.1 in the appendix shows the courses selected from each major/specialization in detail. From the Bruinwalk review interface, in addition to the name of the course and the review content, other features selected were the instructor of the course, grade received by the reviewer, time the course was taken, time the review was written, etc. Table A.2 lists the full set of features selected for each review.

After determining the scope of the reviews, the next step was to collect the reviews. Because the Bruinwalk database doesn't have an associated application programming interface (API), the reviews were directly collected from the website using web-scrapping. The implementation was through a simple Python script, which enabled automated web-scrapping. At the end of this step, a dataset with 7547 course reviews and 9 features was created. Notice that all reviews collected were undergraduate-level courses – there were almost no reviews for graduate-level courses on Bruinwalk, further supporting the point mentioned in the first chapter. Three features in the dataset contained missing values, the Year, Quarter and Grade columns. Those missing information won't have a big impact on the analysis, as the sentiment analysis modeling would only depend on the review content. However, the supplementary features were selected to help us better understand and visualize the structure of the data.

2.3 Data Cleaning and Preprocessing

The next important step in preparing the dataset, was to apply data cleaning and preprocessing to the raw data collected. A three-step process was applied as follows, (1) removing comments that were not relevant to the rating of the courses, e.g., there were around 170 reviews that students wrote to selling their textbooks; (2) removing comments that were non-English – this is because translating non-English utterances into English would require using extra model architectures (which are typically different from those used in sentiment analysis), and because there were only less than 10 such reviews in the curated dataset, directly removing them would be more efficient, and (3) removing excessive whitespaces and symbols (e.g., extra spaces and emojis). After the data cleaning and preprocessing step, the Bruinwalk review corpus contained 7355 observations. In the next section, we will look at miscellaneous properties of the dataset through using various visualization techniques in the exploratory data analysis.

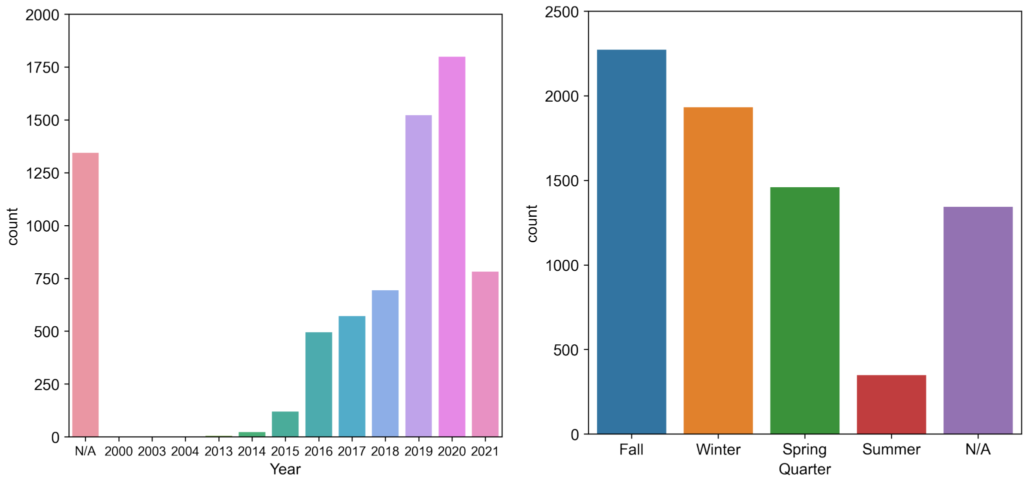
2.4 Exploratory Data Analysis (EDA)

Before implementing sentiment analysis pipeline on the Bruinwalk review corpus, it would be beneficial to have a better understanding of the structure of the data collected. In this chapter, we will look at how to visualize the Bruinwalk review corpus using various exploratory data analysis (EDA) techniques, including the time-series plots, countplots based on the features, and the word frequency plots.

2.4.1 Time-series Plots

As we saw earlier, there are three time-series related features in the corpus, Year (academic year the reviewer took the course), Quarter (academic quarter associated with the Year variable), and Date (date the reviewer wrote the review on Bruinwalk in the format of day, month, and year). Since there are missing values in the Year and Quarter fields, those values were filled in as ”N/A” for clearer visualization results. We first look at the distribution based on the Year and Quarter columns – Figure 2.1 shows the distribution of the reviews based on the academic year and quarter the courses were taken.

From Figure 2.1a, we observe most of the reviews collected were for courses taken from academic year 2015 to 2021, with a higher weight on most recent years, 2019 and 2020. From Figure 2.1b, we observe a decreasing number of reviews



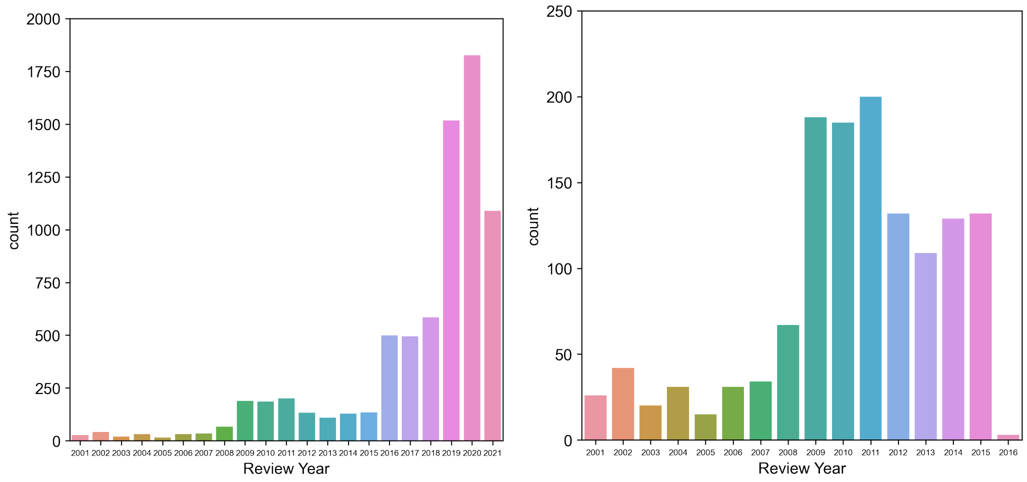
(a) Academic year.

(b) Academic quarter.

Figure 2.1: Number of reviews based on the academic year (Figure 2.1a) and academic quarter (Figure 2.1b) the reviewer took the course.

when going from the Fall quarter to the Summer quarter. From both plots, we see that around 20% of the reviews didn't come with the Year and Quarter values. This indicates it would be more informative to look at the Date variable, which has more complete information. Based on this feature, we can visualize the distribution of the day, month, or year each review was written. Figure 2.2 shows the distributions based on the the year each review was written.

From 2.2a, we observe the distribution based on the review year has a larger spread, especially for reviews written between year 2000 and 2010. We saw in Figure 2.1 that round 1500 reviews had missing Year and Quarter values, and hence it would be interesting to see in which years those reviews were written. Based on the results in Figure 2.2b, we see the reviews with the missing values were all written before 2015. One explanation for this observation, is that the



(a) Review year.

(b) Reviews with missing Year value.

Figure 2.2: Distribution of the review year (Figure 2.2a) and the distribution of the review year for reviews that contained missing Year and Quarter values (Figure 2.2b).

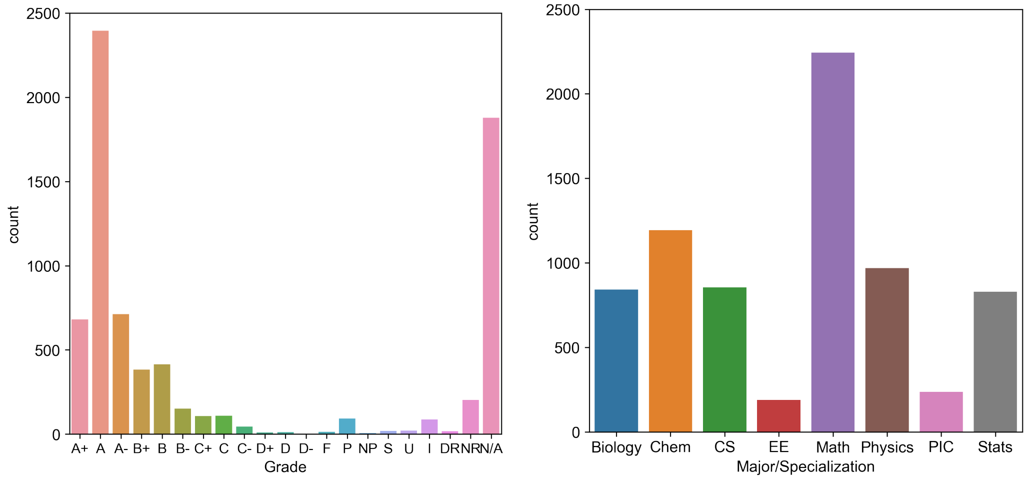
Bruinwalk website made some significant improvements to its interface in 2016, after which the reviewers were allowed to add the Year and Quarter information to their reviews. Overall speaking, we see a peak of the course reviews in the most recent years (2019 to 2021) based on both the time the course was taken and the review year. The result indicates that more students are actively giving their feedback to the community.

2.4.2 Countplots

In addition to the time-series features, there are several other interesting features, such as the grade the reviewer received, the major/specialization of the course, etc. Similar as before, we can visualize the distribution of the reviews based on each feature.

Figure 2.3a shows the countplot of the reviews based on the grade the reviewer received in the course, from which we observe most of the reviews came from students who received a grade of ”B” or better. This suggests an interesting research question for the study, which is whether there is an association between the sentiment of the review and the grade the reviewer received. This question will be answered after applying the sentiment analysis pipeline. Figure 2.3b shows the distribution of the reviews based on the major/specialization they belong to, from which we see around 30% of the reviews were written for the Mathematics major, around 16% were for the Chemistry major, and around 11% were for the Statistics major. This result aligns with our expectation, since the number of reviews associated with each major/specialization are typically influenced by two factors, the number of students in the department, and the number of courses (and submajors) in the department. For example, the Mathematics department at UCLA provides a group of majors, minors, and specializations, and it offers a large range of courses. This indicates it would also be interesting to examine the sentiment towards each major/specialization through the sentiment analysis pipeline.

We looked at the distribution of the reviews based on a single variable, now we can take one step further and nest multiple features. Figure 2.4 shows the review distribution for individual courses, and Figure 2.5 shows the distribution of the reviews based on the grade, both of which are grouped by the major/specialization. We observe most of the course reviews were for lower-level major preparation courses (i.e., courses with number below or equal to 99), and some courses received many more reviews than the others. We can also see the grade distribution varies among different major/specialization, for example, unlike in the other ma-



(a) Grade received.

(b) Major/specialization.

Figure 2.3: Distribution of the reviews based on the grade the reviewer received (Figure 2.3a) and the major/specialization of the course (Figure 2.3b).

jors/specializations, course reviews in the Chemistry, Electrical Engineering and Physics major are less dominated with the grade ”A” or better, and instead they contained many missing values in grade.

The remaining two non-text variables in the Bruinwalk corpus are the Upvote and Downvote variable, which represents the number of up-votes and down-votes received by each review, respectively. Figure 2.6 shows the distribution based on the Upvote and Downvote variables, where we see many of the reviews didn’t receive any upvotes or downvotes. This is within our expectation, as most of the reviews might not be highly polarized in sentiment and hence did not receive any upvotes or downvotes. However, it would be interesting to analyze the distribution based on the upvotes for reviews that received at least one downvote, and vice versa. From Figure 2.7, we observe most of the reviews that received at

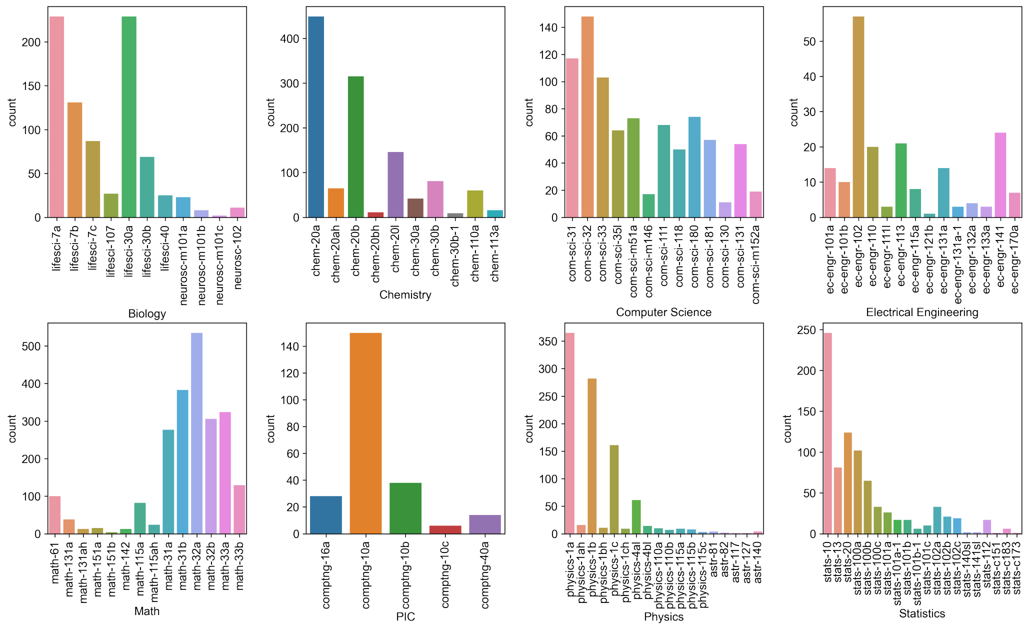


Figure 2.4: Distribution of the reviews based on the courses, grouped by the major/specialization.

least one upvote didn't get any downvotes; however, there were quite a few reviews that received downvotes but at the same time got some upvotes.

The observations from the distributions based on the upvotes and downvotes suggest that the Bruinwalk reviews could be highly subjective – students who took the same course in the same term might still have very different point of views. This leads us to the next part in the EDA, word frequency analysis, from which we could gain a better understanding of the content of the reviews.

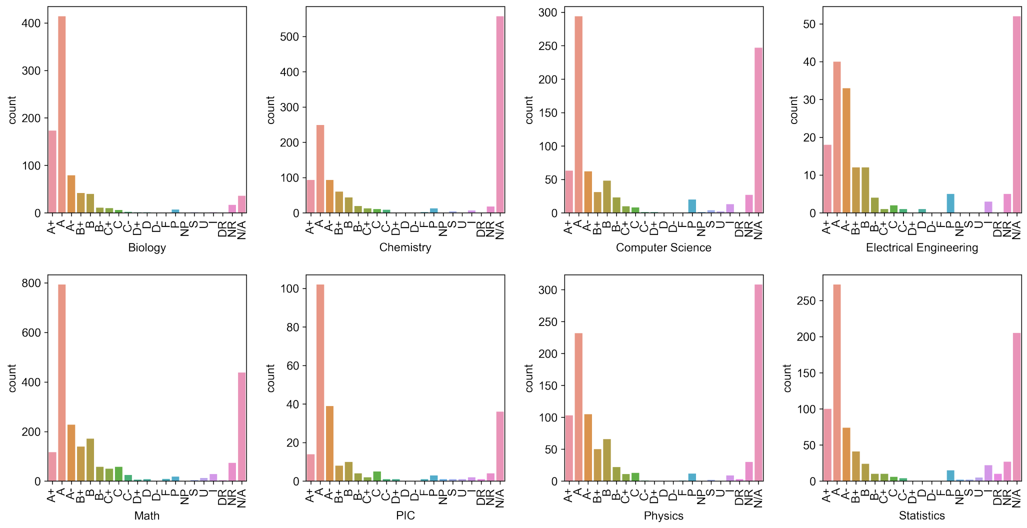
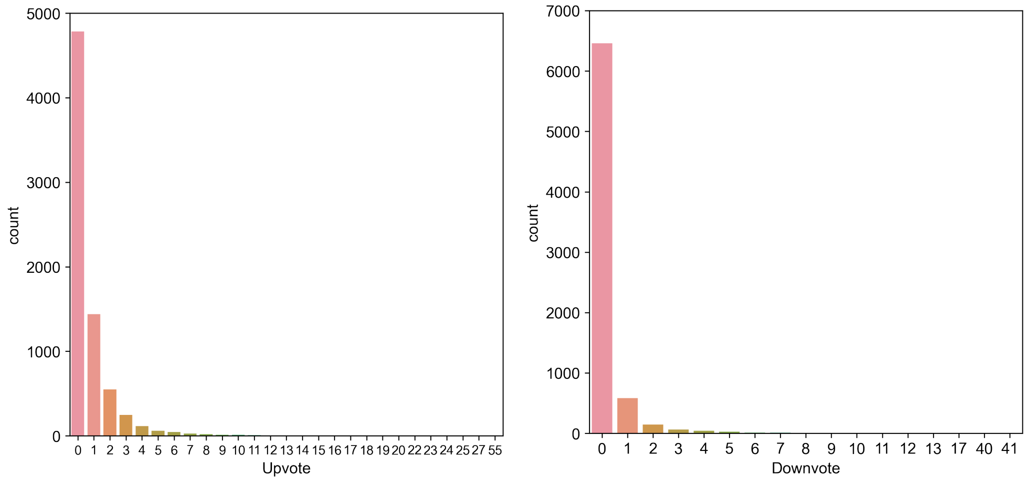


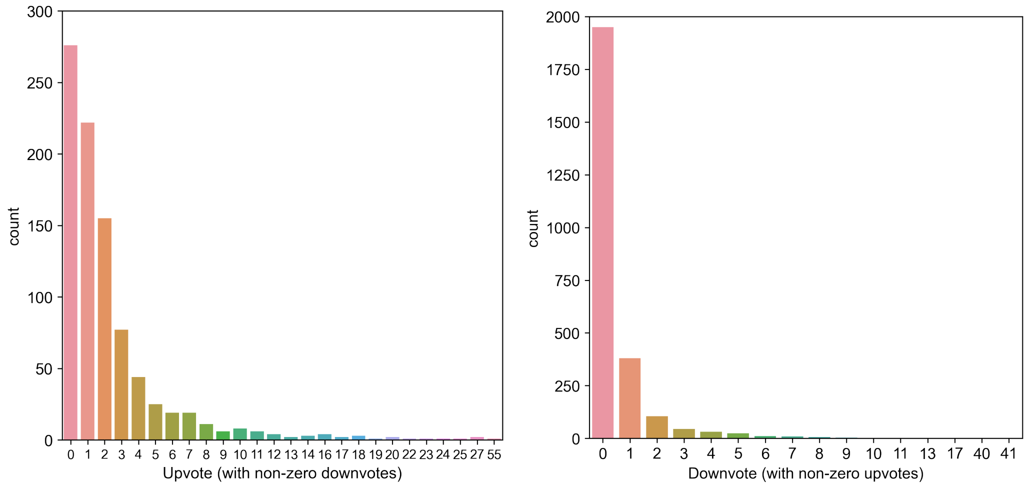
Figure 2.5: Distribution of the grades, grouped by the major/specialization.



(a) Upvotes.

(b) Downvotes.

Figure 2.6: Distribution of the reviews based on the number of upvotes (Figure 2.6a) and the downvotes (Figure 2.6b) received on Bruinwalk.



(a) Upvotes.

(b) Downvotes.

Figure 2.7: Distribution of the upvotes for reviews with non-zero downvotes (Figure 2.7a) and downvotes for reviews with non-zero upvotes (Figure 2.7b)

2.4.3 Word Frequency Plots

The last and the most important feature in the Bruinwalk corpus is the content of the review, since ultimately the sentiment analysis models will be built on this feature only. One common approach to visualize the word distribution in a collection of utterances is through the word frequency plot. Figure 2.8 shows the frequency of the top 20 most common words in the Bruinwalk review corpus. We see from this plot that ”class” is the word that appeared the most frequently, and words such as ”lecture”, ”exam”, ”homework” and ”test” are also very common. Notice that the typical stopwords such as ”and”, ”or”, ”I”, ”the” (i.e., words without actual meanings) were removed from the corpus when creating the frequency plot to better highlight the more unique words. Also, different forms of the same word were combined into a single word (e.g., ”lecture” and ”lectures”) to avoid repeated words or phrases.

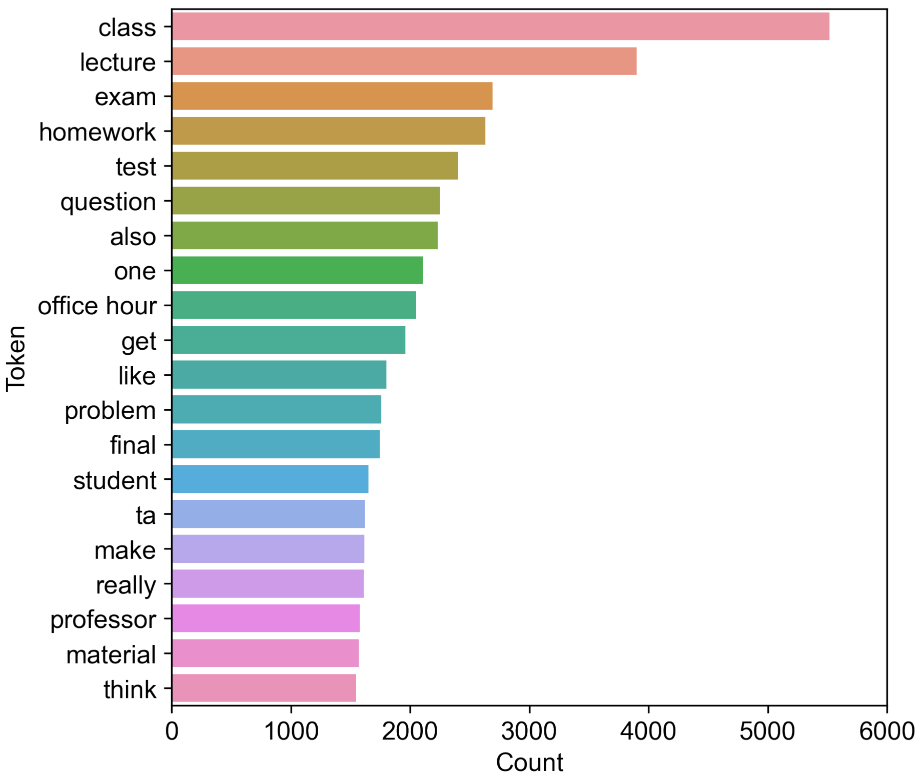


Figure 2.8: Top 20 words (excluding the stopwords) in the Bruinwalk review corpus.

In addition to the traditional word frequency plot, a novel method to visualize the word importance, is through the word cloud. A word cloud (also called a tag cloud) displays the top words from the utterances provided in the corpus, where the color and font size of the words reflect their importance. Figure 2.9 shows the word cloud of all reviews in the Bruinwalk corpus, from which we can easily see that the most common words in the corpus are ”lecture’, ”exam”, ”homework”,

”test”, ”question”, and ”office hour”. Notice that the word ”class” was treated as a stopword in the Bruinwalk corpus and was removed from the cloud map. This is based on the observation from Figure 2.8, that ”class” is dominating the importance ranking, and it would be more interesting to highlight the weights of the other important tokens in the word cloud.

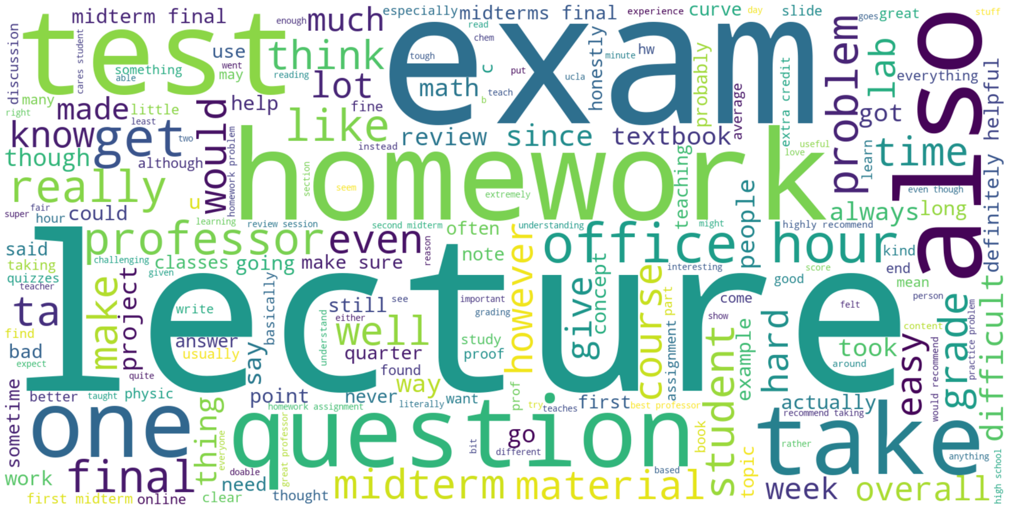


Figure 2.9: Word cloud of all reviews in the Bruinwalk corpus.

CHAPTER 3

Prior Work

In this chapter, we will look at the prior work done in sentiment analysis and natural language processing, as well as the common evaluation metrics used in machine learning classification problems. This chapter will us better understand the logic behind the sentiment analysis pipeline, which will be introduced in Chapter 4.

Natural Language Processing (NLP) is a rising branch of machine learning that is being advanced rapidly. Various topics are being studied in NLP, such as text classification, language translation, language generation, etc. NLP problems are very challenging in that natural languages are highly subjective and people with different culture or background could interpret the same piece of utterance very differently. Sentiment analysis is one of the most well studied text classification problems. It aims to classify the sentiment of the utterances as positive or negative through machine learning modeling. The early-stage NLP models typically consist of two components, the word embedding and the model architecture, while the more recent models use the attention mechanisms to replace the traditional word embedding. Various types of word embeddings and model architectures have been developed and applied to different corpora in NLP. The IMDb review corpus is one of the classical corpora where sentiment analysis has been studied on.

The Internet Movie Database (IMDb) is an online database that consists of millions of reviews for movies, television series, video games, etc. The IMDb review corpus is built for binary sentiment analysis classification, which contains 25000 highly polar movie reviews for training, and 25000 reviews for testing (Maas et al., 2011). Each review comes with the review content and the associated sentiment label. Various types of model architectures have achieved good performance on this corpus. Based on the leaderboard provided by PapersWithCode [1], the current top ten models achieved an accuracy over 95% on the testing set (see Table B.1 for the top models in detail). Among the top models, we will focus on some well-studied and widely used architectures – CNN+LSTM, Transformer, BERT, and DistilBERT, all of which have achieved quite good model performance as shown in Table 3.1.

Rank	**Architecture**	**Accuracy**
5	BERT large finetune UDA	95.8
6	BERT_large+ITPT	95.79
8	BERT_base+ITPT	95.63
9	BERT large	95.49
17	DistilBERT	92.82
19	BP-Transformer + GloVe	92.12
23	CNN+LSTM	88.9

Table 3.1: Top models in sentiment analysis on the IMDb review corpus.

From the table, we see the Transformer, BERT, and DistilBERT models achieved scores above 92%, although the CNN+LSTM model achieved a slightly lower

[1] `https://paperswithcode.com/sota/sentiment-analysis-on-imdb`.

score of 88%. In the following sections of the chapter, we will look at some established work in sentiment analysis and NLP. We will begin with several widely applied word embedding mechanisms, followed by the introduction different model architectures, including the fundamental CNN+LSTM architecture, the novel Transformer model, and state-of-the-art of NLP, BERT and DistilBERT.

3.1 Word Embeddings

Word embedding is a word representation method used in text analysis, which converts words or phrases into high-dimensional numeric vectors so that similar words have similar vector representations and are closer in the vector space. Word embedding models are typically learnt through language modeling and feature learning methods, and the state-of-the-art word embeddings include word2vec, GLoVe, fastText, etc. Other useful methods related to word embeddings are Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE), which have been used to reduce the dimensionality in the word vector spaces and to visualize word embeddings, respectively.

To better understand the idea of embedding the words, suppose we have the following words in the vocabulary: ″man″, ″woman″, ″boy″, ″girl″, ″king″, ″queen″, ″prince″, ″princess″, and ″monarch″ [2]. The simplest method to vectorize the words is the one-hot embedding, where the dimension of each word vector is the size of the vocabulary. Table 3.2 shows the vector representation of each word in the vocabulary using one-hot embedding.

[2] Word embedding examples based on https://www.shanelynn.ie/get-busy-with-word-embeddings-introduction.

Word	**Vector Representation**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
man	1	0	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0	0
girl	0	0	0	1	0	0	0	0	0
king	0	0	0	0	1	0	0	0	0
queen	0	0	0	0	0	1	0	0	0
prince	0	0	0	0	0	0	1	0	0
princess	0	0	0	0	0	0	0	1	0
monarch	0	0	0	0	0	0	0	0	1

Table 3.2: One-hot word embedding for the example vocabulary.

However, the one-hot mechanism is inefficient as it forms very high-dimensional vectors when the vocabulary is large. Hence, a better solution is to vectorize the words base on their semantic similarity. For instance, Table 3.3 shows one possible way to encode the words in the example vocabulary using three features, femininity, youth and royalty, where each feature varies between 0 and 1.

Comparing to the one-hot representation, we see the second approach not only reduces the dimensionality of the vectors, but it also captures the semantic similarity and relationships of the words. The semantic similarity of two words is reflected by the distance (Euclidean distance in Table 3.3's example) between the two vectors. For example, the distance between the vectors of ″girl″ and ″princess″ is smaller than that between ″girl″ and ″prince″. The semantic relationships are also maintained, meaning the vector arithmetics is meaningful. For example, the change in the word vector when moving from ″man″ to ″woman″,

Word	**Vector Representation**		
	Femininity	**Youth**	**Royalty**
man	0	0	0
woman	1	0	0
boy	0	1	0
girl	1	1	0
king	0	0	1
queen	1	0	1
prince	0	1	1
princess	1	1	1
monarch	0.5	0.5	1

Table 3.3: Word embedding example base on the word semantic similarity.

as well as moving from ″boy″ to ″girl″, can be represented as `[1, 0, 0]`.

3.1.1 word2vec

The word2vec word embedding algorithm uses a set of artificial neural network (ANN) model to learn the vector representation of words from a large input corpus. The vector representation of the words is chosen based on the cosine similarity, a metric that measures the semantic similarity of the word vectors (Mikolov et al., 2013). Pre-trained word2vec word embedding models have been applied to various language modeling tasks in NLP, such as the text classification problems, and achieved good model performance. The pre-trained word2vec embedding weights are available in sizes varying from 10 to 1000 dimensions, among which the 300-dimension version is the most frequently used.

3.1.2 GloVe

GloVe (Global Vectors) is another commonly used word embedding method in NLP. It is an unsupervised learning algorithm for obtaining vector representations of words. The model training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations indicate linear sub-structures of the word vector space (Pennington et al., 2014). Similar to word2vec, the words are mapped into the vector space using a similarity measure (Euclidean distance or cosine similarity) so that the distance between each pair of the vectors reflect their similarity in the semantic space.

3.1.3 Limitations

Although word embedding methods have proven to be efficient in many NLP tasks, one limitation of this type of algorithm is that words with multiple meanings are conflated into a single representation (i.e., a single vector representation in the semantic space). In other words, polysemy (a word or phrase that has multiple related but different meanings) and homonymy (words that have similar spelling and pronunciation, but with different meanings) are not handled properly. This is mainly because the word embedding models didn't take into consideration of the context of the text during their training process. As we shall see in the next section, architectures such as Transformer and BERT were trained using techniques that contextualize the text, which resulted in more meaningful and efficient word representations.

3.2 Model Architectures

In this section, we will introduce three types of model architectures in NLP, CNN+LSTM, Transformer and BERT (DistilBERT), all of which are general-purpose language modeling.

3.2.1 CNN, LSTM and CNN+LSTM

CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory) are two fundamental model architectures in deep learning. Unlike traditional neural networks, CNN uses convolutional layers to calculate the connection weights between the neurons, making the computation much more efficient than in the fully-connected (or dense) layers (Kim, 2014). LSTM is one type of the Recurrent Neural Network (RNN) models – it is an architecture that captures the spatial, temporal, and spatio-temporal dependencies of the data (Amato et al., 2020). The CNN+LSTM architecture combines the convolutional layers and the LSTM layers, which has proven to be better than CNN or LSTM alone in many NLP tasks, such as in sentiment analysis (Camacho-Collados and Pilehvar, 2017) and politeness analysis (Niu and Bansal, 2018).

Figure 3.1 illustrates the architecture of the CNN+LSTM model developed for politeness analysis. Politeness analysis is a text classification problem in NLP, and it can be thought as a sibling of the sentiment analysis. Instead of detecting utterances as positive or negative, politeness analysis aims to classify the utterances as polite or impolite. As we see from the architecture, the input text is first passed to the word embedding layer to generate the high-dimensional nu-

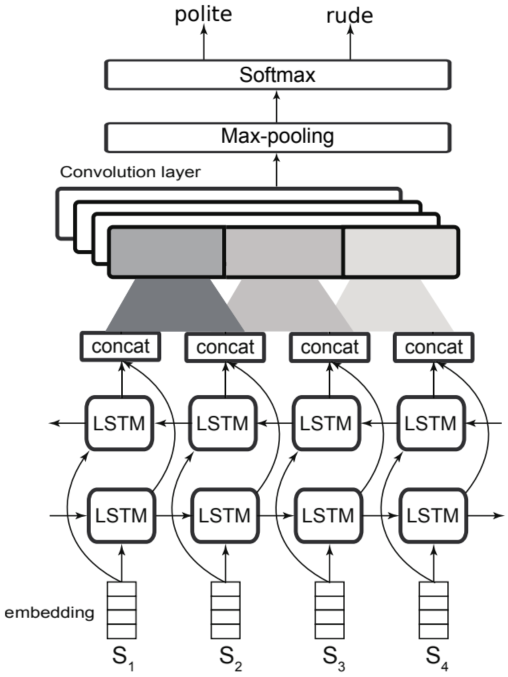


Figure 3.1: Architecture of CNN+LSTM from politeness analysis (Niu and Bansal, 2018).

meric vectors, which then go into the bi-directional LSTM layers. The weights are then passed to the convolutional layers, followed by sub-sampling operation (e.g., max-pooling), an operation used to reduce the computational space in the fully-connected layers that follow. This sequential convolutional recurrent structure is called an encoder-decoder structure. Lastly, the weights generated from the fully-connected layers are passed to the softmax layer to compute the probability of each class, which results in the final prediction using the softmax method. This CNN+LSTM architecture serves as the benchmark in many NLP modeling tasks. However, the architecture is largely limited by the word embedding layer, which lacks the contextualization of the text input. As we shall see next, the Transformer and the Transformer-based BERT and DistilBERT models overcame this issue by introducing the attention mechanism and reformulating the architecture.

3.2.2 Transformer

The traditional CNN+LSTM architecture is a complex convolutional and recurrent neural network in an encoder-decoder configuration. People later developed a mechanism named attention, to connect the encoder and decoder layers more efficiently. However, the CNN+LSTM with the attention mechanism is still not the optimal, since the computation in the recurrent and convolutional layers could be highly expensive when the data is large and complex. A much simpler network architecture, the Transformer, which based solely on the attention mechanisms and dispensing with recurrence and convolutions entirely, was then introduced in the well-known *Attention is All You Need* paper (Vaswani et al., 2017).

Attention is a mechanism frequently used in sequence modeling, which allows the modeling of dependencies without basing on the distance in the input or the output sequences. An attention is essentially a function that maps a query and a set of key-value pairs to an output, where the query, keys, values, and output are vectors. The output is computed as the sum of the values, weighted by the values and a function of the keys. The Transformer model used the multi-head self-attention mechanism to entirely replace the traditional convolution and recurrence structures. Self-attention is a special type of attention mechanism that relates different positions of a single sequence to compute the representation of the sequence (Vaswani et al., 2017). In multi-head attention, the calculation for multiple attention heads can be performed in parallel, which enables a faster and more efficient computation process.

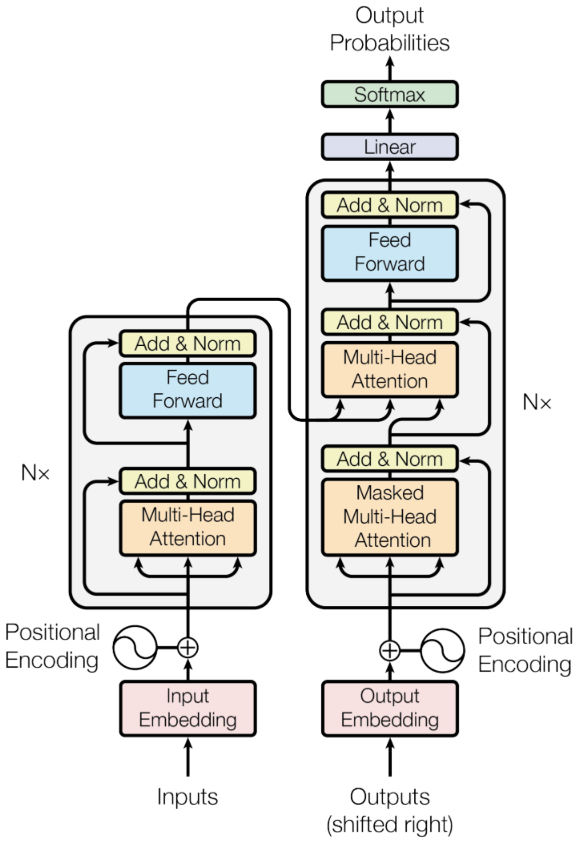


Figure 3.2: Architecture of Transformer (Vaswani et al., 2017).

Figure 3.2 shows the structure of the famous Transformer model. The model consists of an encoder and a decoder structure. The encoder is composed of a set of 6 identical layers, where each layer uses a multi-head self-attention mechanism, followed by a simple fully-connected feed-forward neural network. The decoder is also composed of 6 identical layers, but in addition to the multi-head self-attention and the feed-forward neural network layers, it has a third sub-layer that performs multi-head attention over the output of the encoder layers, referred to as the masked multi-head self-attention layer shown in the figure.

The Transformer models have achieved good performance on various NLP tasks, such as language translation, paragraph generation, and question-and-answering. There are many publicly available pre-trained Transformer models that can be directly applied to downstreaming tasks. As we shall see the next, BERT, the state-of-the-art in NLP modeling, is also a Transformer-based model.

3.2.3 BERT

BERT (Bidirectional Encoding Representations of the Transformer) is a pre-trained machine learning model based on the original Transformer class, which achieved the state-of-the-art performance in many NLP tasks. The original English-language BERT class contains two models – the base model, which has 12 encoders with 12 bi-directional self-attention heads, and the large model, with 24 encoders with 16 bi-directional self-attention heads. During the training process, BERT was pre-trained on two tasks: language modeling (where 15% of the input tokens were masked and the model was trained to predict those tokens from context), and next sentence prediction (in which the model was trained to predict whether a chosen next sentence was probable given the previous sentence). Since the BERT architecture is a pre-trained model, without repeating the computationally expensive training process, the model can be directly applied to downstream tasks.

BERT can be applied to downstream tasks in two major ways, the feature-based approach and the fine-tuning approach. The feature-based approach uses

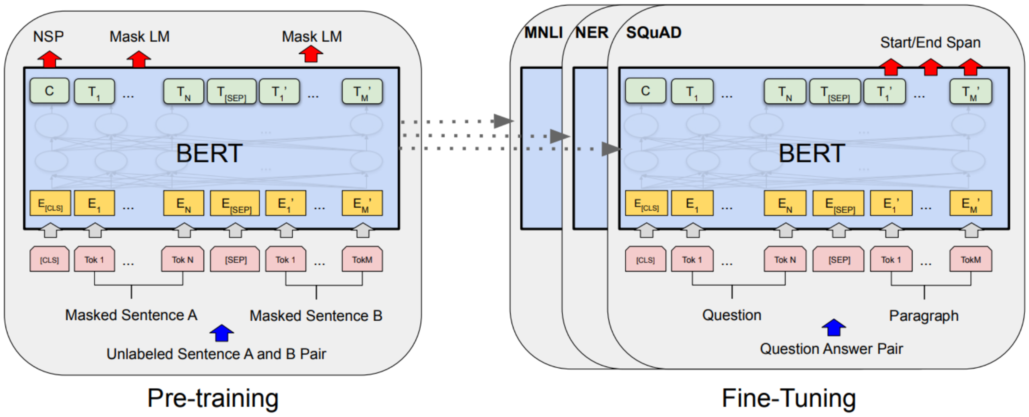


Figure 3.3: Overall pre-training and fine-tuning procedures for BERT (Devlin et al., 2018).

task-specific architectures, where the pre-trained word representations are used as the additional features. The fine-tuning approach contains a small set of task-specific parameters, where the model is trained on the downstream tasks by fine-tuning the pre-trained parameters in BERT. Figure 3.3 illustrates the pre-training and fine-tuning process used in BERT.

3.2.4 DistilBERT

One property of the BERT model is that it contains a very large number of pre-trained parameters (110 million in the base model, and 340 million in the large model). This rich set of parameters could potentially help improve the model performance in many tasks, however, it would be not efficient when there is a constrained amount of computation power and time. Hence, a distill version of BERT was introduced using knowledge distillation, known as the DistilBERT model.

Knowledge distillation is a compression technique in which a compact model (i.e., the student model) is trained to reproduce the behavior of a larger model (i.e., the teacher model) or an ensemble of models (Hinton et al., 2015). Before the development of DistilBERT, most prior work investigated the use of knowledge distillation for building task-specific language models. DistilBERT, on the other hand, leveraged knowledge distillation on the general-purpose BERT model, a process that successfully reduced the model size by 40%, while retaining 97% of the model performance and becoming 60% faster. This faster and lighter model is less expensive to pre-train, and it has achieved performance comparable to BERT on many NLP tasks, despite that it is much more compact (Sanh et al., 2019).

3.3 Evaluating the NLP Models

To fairly compare the model performance, it's important to choose the right evaluation metrics. Because there are only ″right″ and ″wrong″ predictions in classification problems, it's not meaningful to compute the prediction scores such as the variation explained (R^2), or the mean squared error (MSE). Instead, we can evaluate the classification models based on their prediction accuracy, recall, precision, and F_1-score.

Accuracy is the metrics that tells the proportion of the samples correctly classified by the model. It's defined as the ratio of the number of correctly classified samples to the total number of samples in the pool of interest. The sentiment analysis models shown in Table 3.1 were ranked based on their accuracy on the IMDb testing set. Formally, the accuracy is defined as follows:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} \tag{3.1}$$

where

		Predicted Condition	
		Positive (PP)	**Negative (PN)**
Actual Condition	**Positive (P)**	True Positive (TP)	False Negative (FN)
	Negative (N)	False Positive (FP)	True Negative (TN)

Table 3.4: Terms used in binary classification problems.

Besides the model accuracy, people also use other metrics to evaluate the model's ability to predict a certain class. In classification problems, the recall of a class is the fraction of the true positives to the sum of the true positives and false negatives, which is also referred to as the true positive rate (TPR), or sensitivity. The precision of a class is the fraction of the true positives to the sum of the true positives and false positives, which is also known as the positive predictive value (PPV). Another useful metric is the specificity, which is the ratio of the true negatives to the sum of the true negatives and the false positives, which is typically used together with sensitivity. Equation 3.2 - 3.4 summarize the definition of the three metrics.

$$\text{recall} = \frac{\text{TP}}{\text{P}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.2}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.3}$$

$$\text{specificity} = \frac{\text{TN}}{\text{N}} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3.4}$$

In addition to the precision and recall, another frequently used metric is the F_1-score, computed as the harmonic mean of the recall and precision (see Equation 3.5 for the formula). It is a summary of the recall and precision, with the highest possible value being 1, indicating a perfect score in both the recall and the precision, and the lowest possible value being 0, if either the recall or the precision is zero. The F_1-score is more commonly used than the precision or recall alone, especially in the cases where the classes are considered as equally important.

$$\text{F}_1\text{-score} = 2 \cdot \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})} \tag{3.5}$$

Consider the example of a test for diagnosing a disease. The sensitivity (recall) is the test's ability to correctly identify patients who have the disease, and the specificity is its ability to correctly identify the healthy patients (see Equation 3.6 and 3.7 for the probabilistic representation). From this example, we see there is always a trade-off between the sensitivity and specificity, and depending on the scenario, one metric might be favored more than the other.

$$\text{sensitivity} = \mathbb{P}(\text{positive test result}|\text{patient has the disease}) \tag{3.6}$$

$$\text{specificity} = \mathbb{P}(\text{negative test result}|\text{patient doesn't have the disease}) \tag{3.7}$$

CHAPTER 4

Methodology

Having introduced the prior work done in sentiment analysis, in this chapter, we will look at how the sentiment analysis pipeline was designed and implemented in the study. In particular, the pipeline consisted of two components, the automated labeling step to annotate the raw Bruinwalk reviews, and the sentiment analysis modeling step to pre-train a model using the annotated Bruinwalk reviews.

4.1 Bruinwalk Review Automated Annotation

We looked at the structure of the Bruinwalk corpus back in Chapter 2, but one issue of this dataset is that there was no sentiment label for each review. Having an unlabeled corpus is a commonly encountered problem in NLP, in which case we need to first generate the annotation before supervised learning tasks can be done. One way to annotate the text is through automated labeling, which is to train a model based on some existing human-labeled or manually validated corpus, and then use the trained model to annotate the unlabeled corpus (Madaan et al., 2020). Leveraging the prior work done on the IMDb review corpus, we can train a sentiment analysis model and use it to label the Bruinwalk corpus.

4.1.1 Model Architecture Selection

As we saw in the previous chapter, the CNN+LSTM, Transformer, BERT and DistilBERT models have achieved state of the art performance in sentiment analysis on the IMDb dataset. Therefore, these model architectures were re-built and used in the automated labeling process. The word embedding used in the CNN+LSTM architecture was the 300-dimensional word2vec, as it's currently one of the most ubiquitous and well-established word embedding models.

4.1.2 Cross-domain Validation

To make sure the re-established models achieved comparable performance to the benchmark listed in Table 3.1, the models were validated on a second corpus, the US airlines Twitter review corpus (shortened as airline review corpus) using cross-domain validation. The airline review corpus contains around 10K Twitter posts that are customer reviews for various US airlines. Each review is equipped with a sentiment label, negative (-1), neutral (0) or positive (1). To be consistent with the IMDb corpus, only the negative and positive reviews were selected in validating the models. Cross-domain validation is a machine learning technique to test a model's generalizability to data in different domains (Heredia et al., 2016), which is to train a model using one dataset, and test it on a dataset from a different domain. We want the model to generalize well so that it can both interpolate and extrapolate well on various types of data, including the Bruinwalk corpus.

To enable faster and larger amount of computation, the models were trained using Google Colab Pro, a service that provides faster and more GPUs and more computation memory. However, because there was still limited amount of computation resources, the DistilBERT architecture was selected over BERT, considering DistilBERT is more compact while maintaining comparable performance to BERT. Hence, the models that were reproduced in the automated labeling step were the CNN+LSTM, Transformer and the DistilBERT models.

The implementation of CNN+LSTM and Transformer was through Tensorflow Keras, an open-source Python package that provides a large variety of tools for neural network modeling in machine learning and deep learning. The implementation of DistilBERT was through the HuggingFace transformers module, a repository for Transformer-based machine learning models and datasets. To search for the best setup of each architecture, hyperparameter optimization was applied using grid search based on the performance on the IMDb testing set (Shekar and Dagnew, 2019). The primary evaluation metrics used was the accuracy, and to be less biased, the recall, precision and F_1-score were also evaluated. See Table C.1 in the appendix for the hyperparameters selected for each model. After getting the optimal setup, each model was then validated on the entire airline review corpus. Table 4.1 shows the cross-domain validation performance using each model.

Model	**IMDb Testing Set**				**Airline Reviews**			
	Accuracy	**Precision**	**Recall**	**F_1-score**	**Accuracy**	**Precision**	**Recall**	**F_1-score**
LSTM-CNN	87.7%	88.0%	88.0%	87.5%	69.5%	78.4%	69.4%	72.4%
Transformer	87.3%	87.5%	87.0%	87.5%	73.2%	80.0%	73.0%	74.9%
DistilBERT	91.2%	91.0%	91.5%	91.0%	81.4%	86.4%	81.4%	82.5%

Table 4.1: Model performance on the IMDb testing set and the airline reviews corpus.

From this table, we see the results are quite comparable to the benchmarks shown in the leaderboard, where the DistilBERT model achieved the best performance on both the IMDb testing set and the airline review corpus, followed by Transformer and the CNN+LSTM model. The observation that the model performance on the airline review corpus was slightly worse than those on the IMDb testing set, but still reasonably good, indicates the models could generalize quite well to corpora in different domains. After the validation step, each model was used to predict the sentiment (i.e., automated labeling) of the Bruinwalk reviews.

4.1.3 Testing Set Performance Evaluation

To evaluate the predictions, a small subset of the Bruinwalk reviews was manually annotated. The selection of the reviews was based on the upvotes and downvotes variables, where 50 reviews with the highest number of upvotes and 50 with the most downvotes were selected, respectively. This set of 100 reviews was then labeled based on their sentiment. After creating and annotating the small testing set, the predictions were evaluated based on the same metrics as in the cross-domain validation procedure.

Since there are three sets of the sentiment labels, model ensembling was used to create a summary of the predictions, a method that could potentially generate better annotation by internally canceling some prediction errors (Minaee et al., 2019). Majority voting is a commonly used ensemble method in machine learning, which is to combine the predictions from multiple models using a majority vote. Figure 4.1 shows the distribution of the votes received by the reviews using majority voting.

From the figure, we see around 60% of the reviews received the same annotation from all three models (i.e., three votes). This indicates that most of the predictions from the models were quite consistent. Table 4.2 shows the performance of the individual models, as well as the majority voting ensemble method, on the manually annotated Bruinwalk testing set.

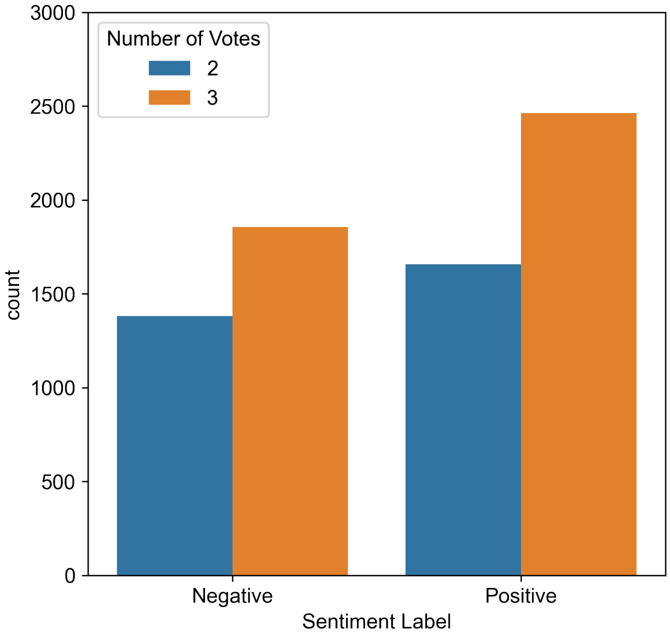


Figure 4.1: Majority voting results from ensembling the CNN+LSTM, Transformer, and the DistilBERT models.

Model	**Metrics**			
	Accuracy	**Precision**	**Recall**	**F_1-score**
CNN+LSTM	72.0%	71.6%	71.7%	71.2%
Transformer	78.0%	79.2%	77.8%	78.0%
DistilBERT	87.0%	87.5%	87.0%	86.6%
Majority Voting	83.0%	83.2%	82.7%	83.0%

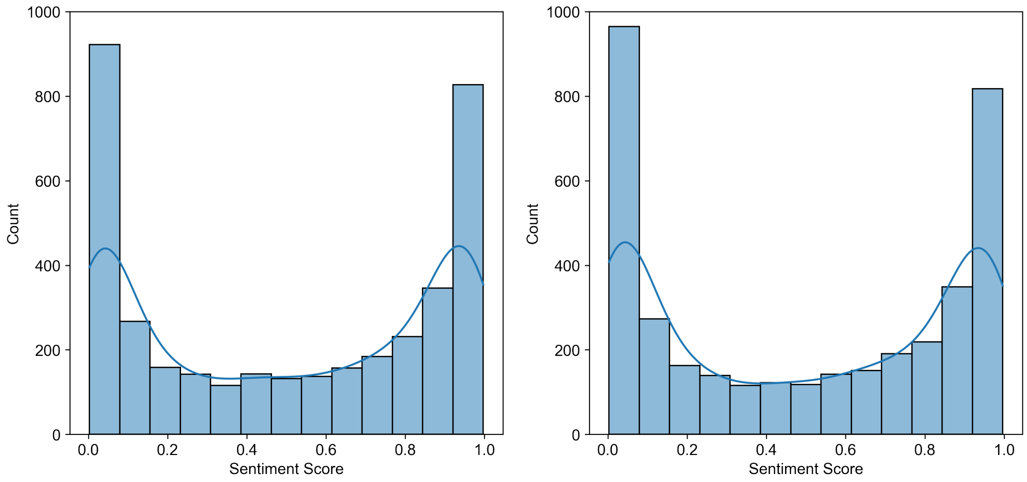
Table 4.2: Model performance on the manually annotated Bruinwalk testing set.

From Table 4.2, we observe the DistilBERT model achieved better performance than the other two architectures. Comparing the majority voting method with the individual models, we see majority voting did better than CNN+LSTM and Transformer alone, but it did slightly worse than the DistilBERT model. Based on the results, the automated labeled annotation from the DistilBERT model was chosen as the final annotation for the Bruinwalk corpus.

4.2 Bruinwalk Review Sentiment Analysis Modeling

The final step in the sentiment analysis pipeline, is to train a model using the automated labeled Bruinwalk reviews. There are two main benefits for pre-training this model: firstly, we can further examine the quality of the labels by cross-domain validation, and secondly, we can later utilize this pre-trained model to label new reviews on Bruinwalk, and to get more insights from the reviews in future investigation.

Stratified sampling was used create the training and testing set of the Bruinwalk review corpus. The reviews were stratified into 10 strata based on their sentiment score, where the sentiment score is a number between 0 and 1 – a value closer to 0 means the model has a strong confidence that the review is negative, versus a value closer to 1 indicates a strong confidence in the review being positive. In each stratum, 50% of the reviews were randomly selected into the training set, and the remaining half went into the testing set. Figure 4.2 shows the distribution of the sentiment score in the training and testing set. From this figure, we see from using the stratified sampling, the obtained training and testing set have similar distribution in the sentiment score, which is close to the train-test split setup used in the IMDb corpus.



(a) Training set.

(b) Testing set.

Figure 4.2: Distribution of the sentiment score in the Bruinwalk training and testing set based on a 50% split from stratified sampling.

Based on the observation that the DistilBERT model achieved the best performance, DistilBERT was selected as the pre-training architecture. The model was trained with the Bruinwalk training set, validated on the Bruinwalk testing set, IMDb testing set and the airline review corpus, and was tested on the manually annotated Bruinwalk testing set. We will later refer to this model as the Bruinwalk sentiment analysis model. Table 4.3 shows the model performance on the validation sets and the testing sets.

From the table, we see the Bruinwalk sentiment analysis model achieved comparable results to the models trained with the IMDb reviews. This further verified that the labels obtained from the automated labeling process are quite reliable, and the automated labeled corpus can help train a model that generalizes well to cross-

Model	**Metrics**			
	Accuracy	**Precision**	**Recall**	**F_1-score**
Bruinwalk Testing Set	85.3%	85.5%	85.5%	85.5%
IMDb Testing Set	84.0%	84.5%	84.0%	84.0%
Airline Reviews	86.0%	88.0%	85.6%	86.7%
Annotated Bruinwalk Test Set	85.4%	85.0%	85.5%	84.9%

Table 4.3: DistilBERT-based Bruinwalk sentiment analysis model performance on the cross-domain validation sets and the testing sets.

domain data. After validating the model, we could utilize this pre-trained model to label new Bruinwalk reviews in the future studies, and to gain more insights in the long run. The pre-trained model is published on the HuggingFace spaces (similar to GitHub repositories), and it can be easily retrieved via the model alias `kaixinwang/NLP` [1]. The HuggingFace spaces also have an interactive demo of the analysis, which allows the user to enter a text input, after which the program utilizes the pre-trained model to make the prediction, and the predicted sentiment label and the associated prediction score of the input will be returned [2].

[1] Model repo at `https://huggingface.co/kaixinwang/NLP`.

[2] Interactive demo available at `https://huggingface.co/spaces/kaixinwang/NLP`.

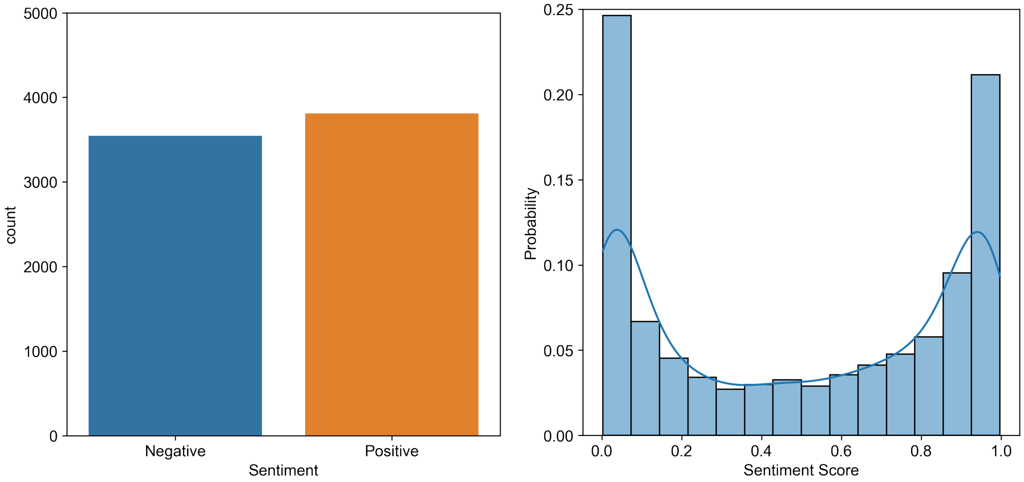
CHAPTER 5

Results and Evaluation

In the previous chapter, we looked at the implementation of the sentiment analysis pipeline and its validation process. Similar to the EDA step prior to the modeling, in this chapter, we will analyze the results using similar data visualization methods, such as visualizing the distribution of the sentiment label and score, time-series plots, word frequency plots, etc.

5.1 Distribution of the Sentiment Annotation

As we saw in the previous chapter, the predictions from the sentiment analysis models included both a sentiment score that is continuous between 0 and 1, and a discrete sentiment label that is either -1 (negative) or 1 (positive). The sentiment score can be thought as a measure of the prediction confidence, since reviews with a score above or equal to 0.5 are considered positive, and the rest are classified as negative. Figure 5.1 shows the distribution of the sentiment score and the sentiment label. From Figure 5.1a, we see there are slightly more positive reviews than the negative reviews in the Bruinwalk corpus, and from Figure 5.1b, we see most of the reviews have a prediction score above 90% or below 10%, which indicates the DistilBERT model had quite strong confidence in its predictions.



(a) Sentiment label.

(b) Sentiment score.

Figure 5.1: Distribution for two types of sentiment annotation.

Figure 5.2 shows the distribution of the sentiment score within each major/specialization. We observe that some distributions are more skewed towards the regions where the score is above 90% or below 10%, such as in Chemistry, Mathematics and Statistics, while the other distributions have a larger spread. This reflects that the DistilBERT model had stronger confidence in predicting the reviews from certain STEM fields, and predicting the reviews from the other fields was more challenging.

Figure 5.3 shows the distribution of the reviews based on the grade and major/specialization, grouped by the sentiment label. Figure 5.3a verified that there is approximately a positive association between the letter grade and the sentiment – students who received a better letter grade tend to write a review that is positive. It's clear from Figure 5.3b that some majors/specializations have a quite balanced distribution between the positive and the negative reviews, such as Electrical En-

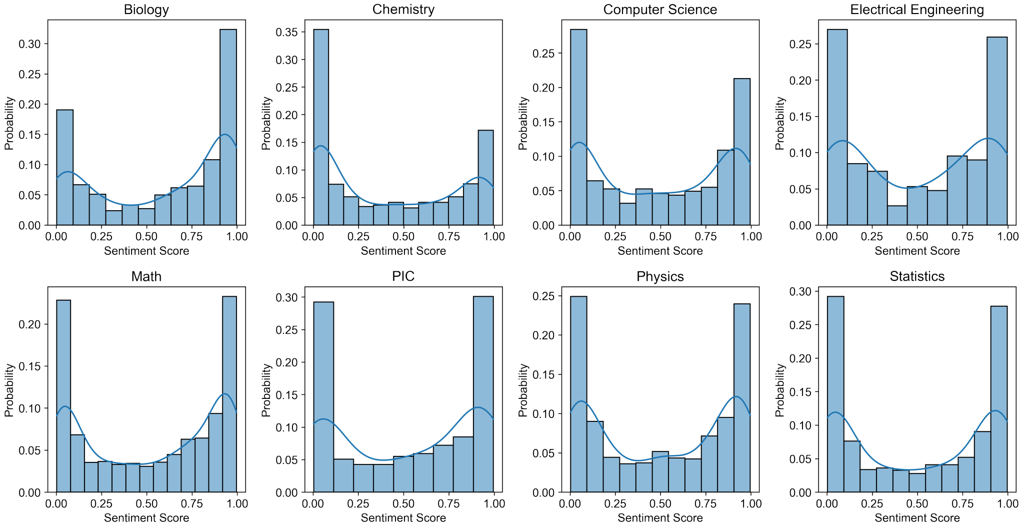
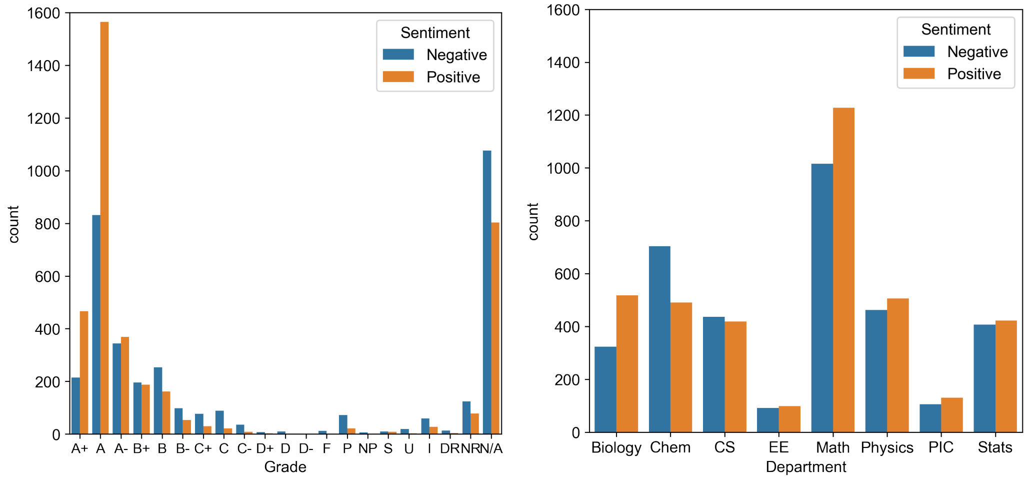


Figure 5.2: Sentiment score distribution based on the major/specialization.

gineering, Mathematics, PIC, Physics and Statistics, while the other distributions are slightly skewed. For example, there are more positive reviews than the negative ones in Biology, and there are more negative reviews than the positive ones in Chemistry and Computer Science.

To further confirm the positive association between the grade and the sentiment of the reviews, Figure 5.4 shows the relation between the positive sentiment rate and the grades. From these two plots, we observe a clear positive association between the letter grade (A+ to F) and the sentiment, although with some light violations – for example, the positive sentiment rate for the grade D- is 100%, as there was only one review belonging to this group and the review was classified as positive. For the non-letter grades listed in the plots, it's more difficult to conclude a general trend or a association relation since it's less meaningful to rank the categories in a numeric order. However, we can still observe an overall positive



(a) Grade.

(b) Major/Specialization.

Figure 5.3: Number of reviews based on the grade (Figure 5.3a) and major/specialization (Figure 5.3b), colored by the sentiment label.

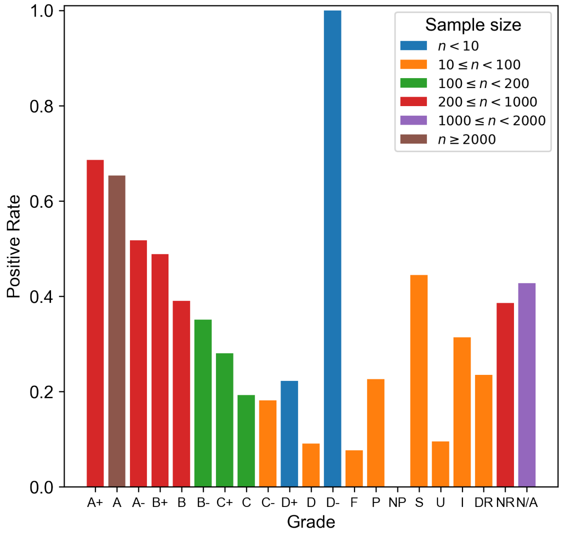
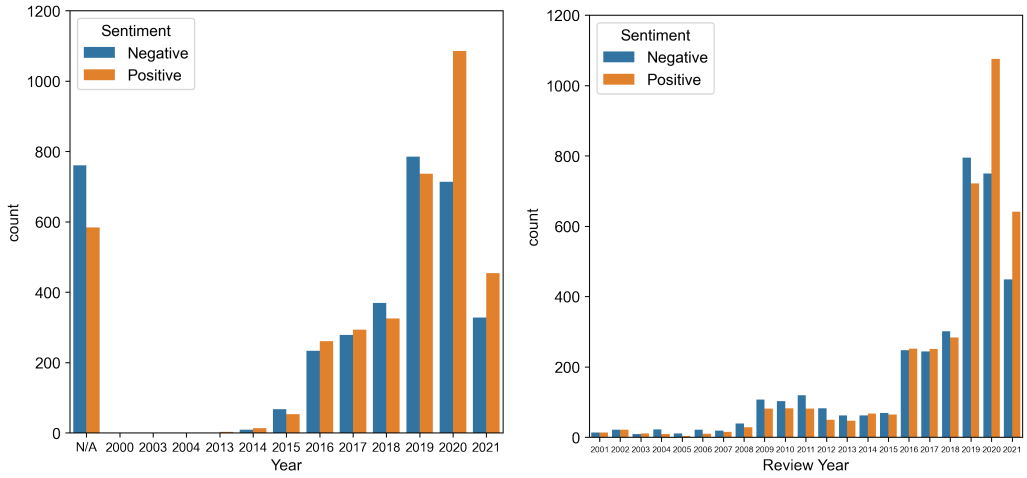


Figure 5.4: Positive sentiment rate vs. grade for the Bruinwalk reviews.

association, e.g., the positive rate is higher for the grade P (passed) than NP (not passed), which is also the case for the pair S (satisfied) and U (unsatisfied).

We can also look at the time-series plots based on the annotated Bruinwalk corpus. Figure 5.5 shows the distribution based on the year of the course and the year the review was written, colored by the sentiment label. We see from both plots that the positive reviews are dominating in more recent years, such as 2020 and 2021, although for most of the years, the distribution is quite balanced in sentiment.

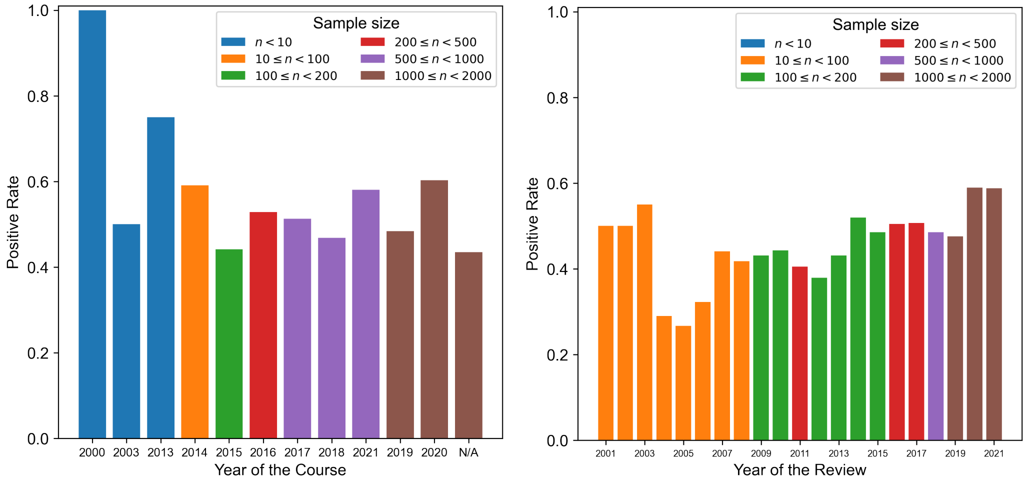
Figure 5.6 shows the positive sentiment rate with respect to the course year and the review year. From Figure 5.6b, we observe reviews written in the most recent years (2019 to 2021) have a relatively high positive sentiment rate. In addition, we see reviews written from 2019 to 2021 have large sample sizes (between 1000 and 2000 reviews), which take up around 40% of the entire Bruinwalk corpus. Similarly in Figure 5.6a, we see most of the reviews were written for courses in 2019 and 2020. However, we observe a higher positive sentiment rate for courses in earlier years (e.g., 2000, 2003 and 2013), which have smaller sample sizes, and we observe slightly lower positive sentiment rate when the sample size gets larger, such as for years between 2017 and 2020. The observation that the positive sentiment rate is also associated with the sample size indicates the bias in selecting the samples, which is not a rigorous random sampling procedure. Same behavior could also be observed in the grade distribution (Figure 5.4).



(a) Academic year.

(b) Review year.

Figure 5.5: Review distribution based on the academic year the course was taken (Figure 5.5a) and the year the review was written (Figure 5.5b), colored by the sentiment label.



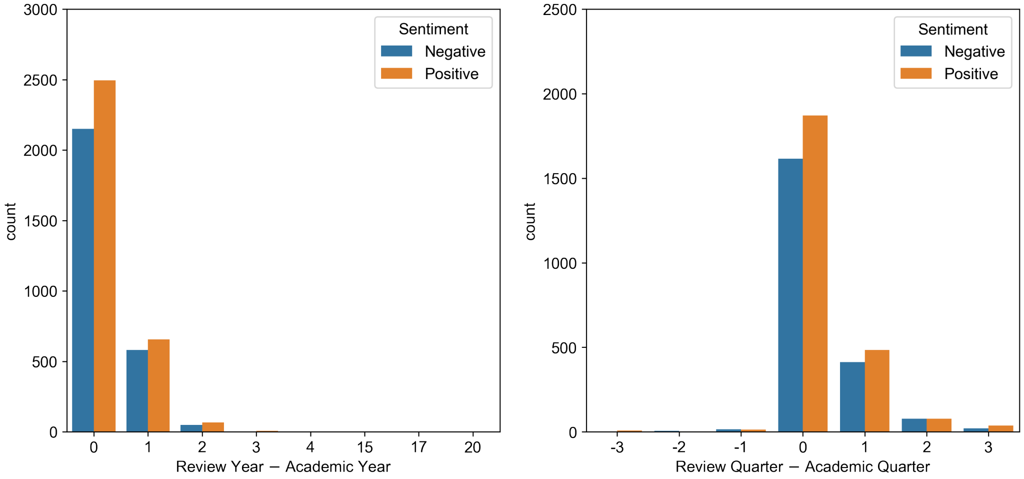
(a) Year of the course.

(b) Year of the review.

Figure 5.6: Positive sentiment rate vs. course year (Figure 5.6a) and positive sentiment rate vs. review year (Figure 5.6b).

We saw the distribution based on the academic year/quarter and the review year/quarter are slightly different. To better understand where the difference came from, it's helpful to look at the distribution based on the elapse between the review time and the time the course was taken. Figure 5.7 shows the distribution based on the time elapse between the year variables and the elapse between the quarter variables. From the year elapse plot, we see most of the reviews were written in the year that the student took the course (elapse of 0), around 1300 reviews were written within one to two years after taking the course (elapse of 1 or 2), and very few had an elapse greater than two years (elapse of 2 or larger). Likewise, in the quarter elapse plot, most of the reviews were written while still taking the course or immediately after taking the course, although some reviews were written one to three quarters afterwards. A new observation from Figure 5.7b, is that there are negative values in the quarter elapse. Upon manually checking the corpus, the reviews that had a negative quarter elapse typically came with the wrong information in the academic year or quarter column. For example, there were reviews labeled for Fall 2021 courses but written in June 2021. This indicates that some improvement can be made to the Bruinwalk review interface so that users can more easily and accurately fill out the information.

The last step in the post-annotation EDA, is to analyze the distribution based on individual courses. Figure 5.8 shows the distribution of the reviews based on the courses, colored by the sentiment label. It's clear that some courses are heavily dominated by the positive reviews, such as Life Sciences 30A, Computer Science 32 and Electrical Engineering 102, although there are also a few courses dominated by the negative reviews, such as Chemistry 20A and Chemistry 20B. If we rank the courses base on the positive sentiment rate (shown in Figure 5.9a), we see



(a) Year elapse.

(b) Quarter elapse.

Figure 5.7: Distribution of the reviews based on the elapse between the review year and the academic year (Figure 5.7a) and the elapse between review quarter and the academic quarter (Figure 5.7b).

some courses had a positive rate of 100%, resulted from the fact that there were very few reviews for some courses, all of which were classified as positive in the modeling step. To fix the issue of small sample size in some courses, a constraint was added to filter down to courses with a large enough sample size. Figure 5.9b shows the top 20 courses ranked based on the positive sentiment rate, where each course had at least 10 samples. From this plot, we see a much smoother distribution of the positive sentiment rate than before, aligning with the observation from Figure 5.8.

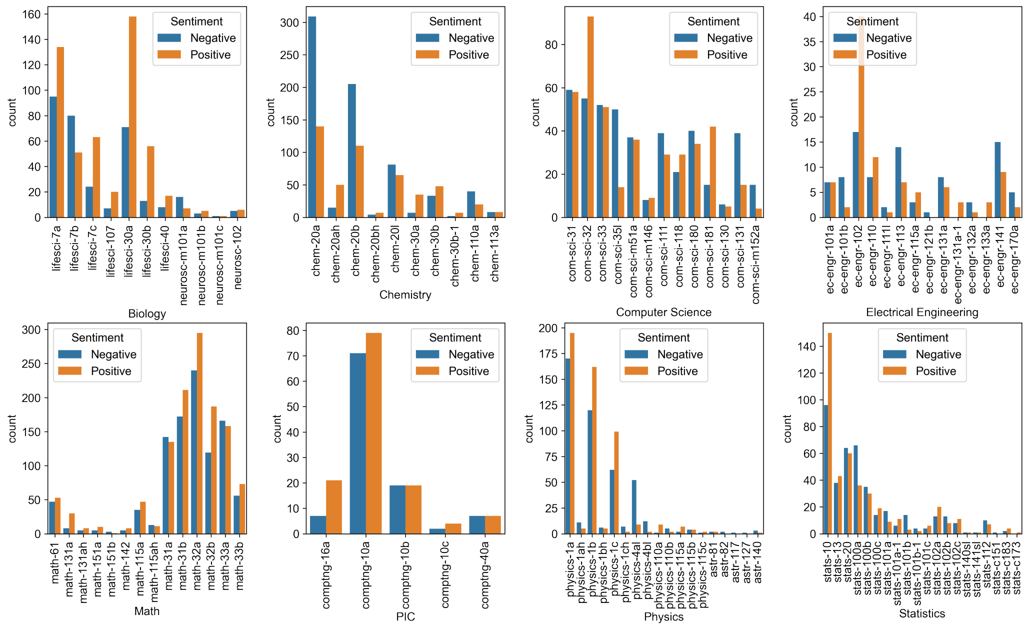
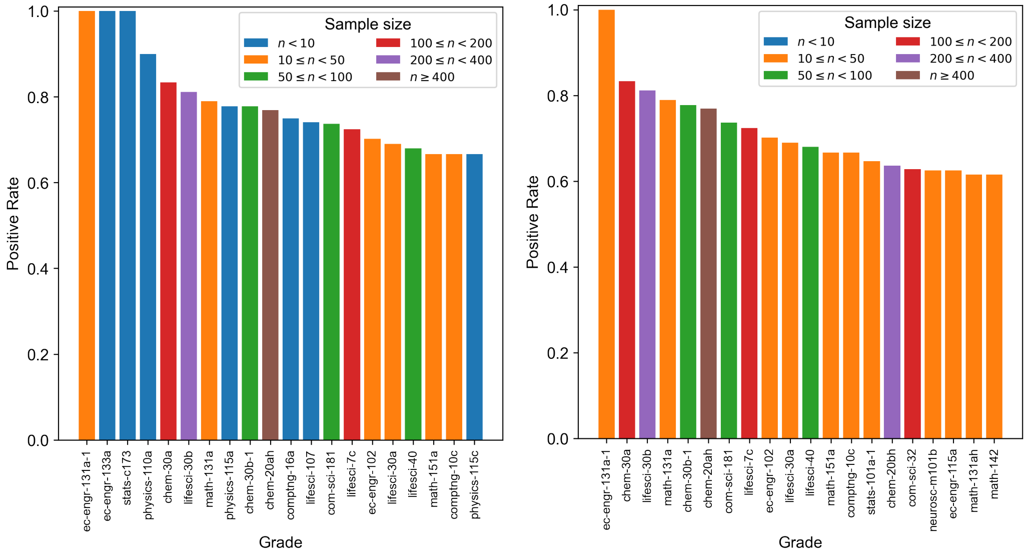


Figure 5.8: Sentiment label distribution of the selected course reviews.



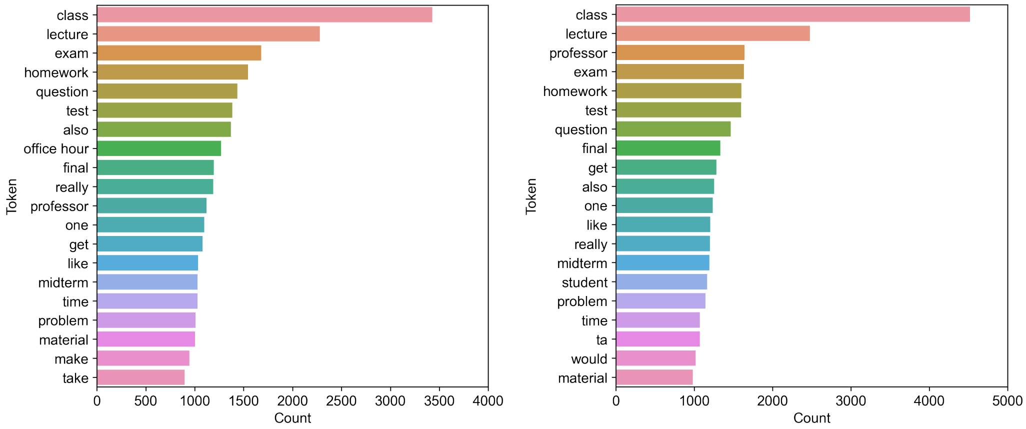
(a) Top 20 courses.

(b) Top 20 courses (with 10 or more reviews).

Figure 5.9: Top 20 courses ranked based on the positive sentiment rate.

5.2 Word Frequency Plots

Now that the reviews are now annotated, we can re-construct the word frequency plot the word cloud map to compare the positive and negative sentiment groups. Figure 5.10b and 5.10a shows the frequency distribution of the 20 most common words in the negative and positive sentiment group, respectively. Like in the overall word frequency plot, it's clear that the word ″class″ is dominating in both classes. By treating the word ″class″ as a stopword, Figure 5.11 shows the word cloud in the negative and positive sentiment groups.

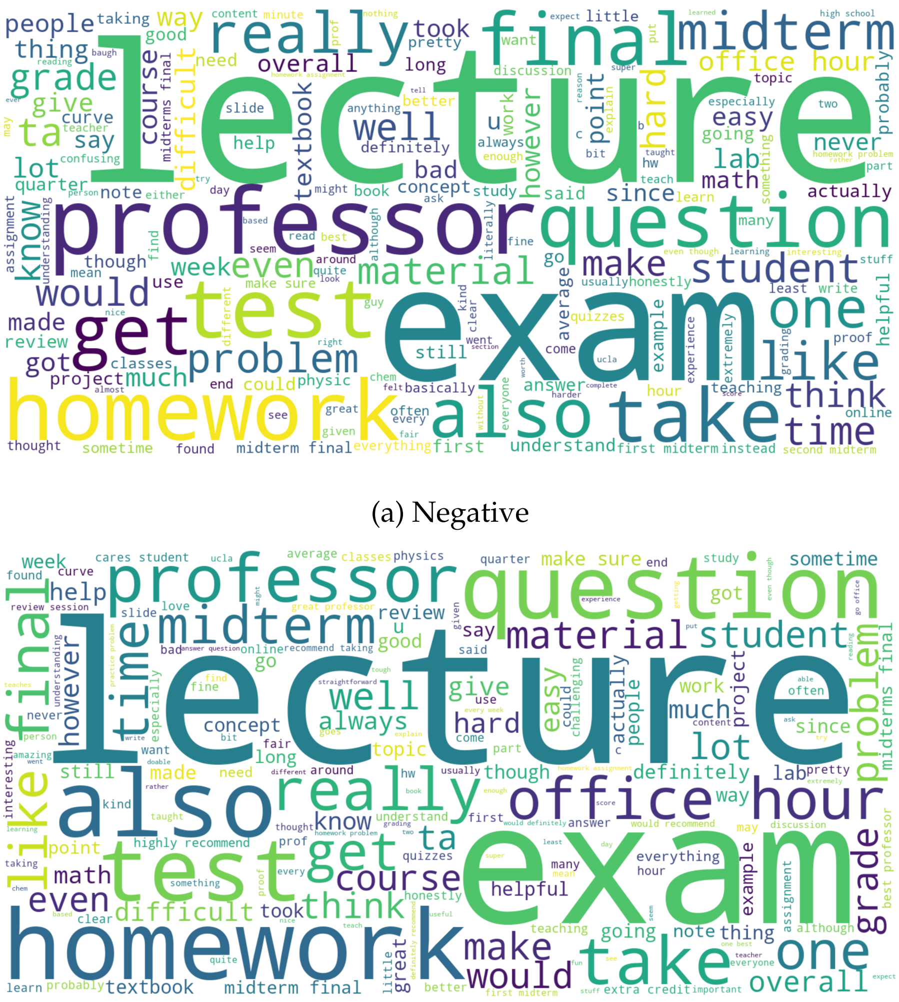


(a) Negative

(b) Positive

Figure 5.10: Top 20 most common words in the Bruinwalk corpus.

From the word frequency plots and the word clouds, we see some words and phrases appeared frequently in both groups of reviews, such as ″lecture″, ″professor″, ″final″, ″exam″, etc. Some differences are that words such as ″office hour″ and ″take″ appeared more frequently in the positive reviews, while the words ″ta (TA)″ and ″student″ appeared more often in the negative reviews. We can see



(a) Negative

(b) Positive

Figure 5.11: Word cloud of reviews grouped by the sentiment.

that words that contain positive sentiment tend to appear more frequently in the positive reviews, while words that have a negative meaning, such as criticism, appeared more often in the negative group.

5.3 SHAP Analysis

One of the biggest challenges in machine learning is to balance the trade-off between the model accuracy and model interpretability. This is especially the case for NLP modeling. As the name ”black-box” model suggests, it’s typically hard to obtain a good understanding of how the NLP model learned the relation between the inputs (the text) and the output (the sentiment label) by optimizing the connection weights of the latent nodes. One novel approach to quantify the impact of the inputs on predicting the output, is the SHapley Additive exPlanations (SHAP) analysis. In brief, SHAP gives each feature an importance value based on its contribution to a particular prediction (Lundberg and Lee, 2017). SHAP has been applied to different types of machine learning models, including the tree ensembling models (e.g., Extreme Gradient Boosting, or XGBoost model), NLP models (e.g., Transformer model), deep neural network models, as well as many other general-purpose models (e.g., Support Vector Machine models, Gaussian Process Regression models, etc.).

To understand how the DistilBERT model established in this study made the sentiment predictions, SHAP was applied to visualize the feature importance. Figure 5.12 shows the SHAP summary plot on predicting the small annotated Bruinwalk test set (i.e., the set that contains 100 polar Bruinwalk reviews that were manually annotated). The x-axis shows the average marginal contribution (i.e., the mean SHAP value) of each feature in predicting the sentiment label, where a positive value means the feature had a positive effect on the prediction score, and a larger magnitude indicates a higher feature importance. The results from Figure 5.12 align with our expectation, where words that contain strong positive sentiment (e.g., ”thank”, ”appreciate” and ”like”) were found to have a positive contribution to the prediction, whereas words that have negative sentiment, such as ”run” and ”awful”, had an negative contribution. An interesting finding is that the word ”goat” (shortcut for ”greatest of all time”) was treated as a negative word in predicting the sentiment, although we would expect the opposite trend. One possible reason is that ”goat” wasn’t a common word in the training data lexicon, meaning the model needs to extrapolate the effect of this unseen feature when making the predictions.

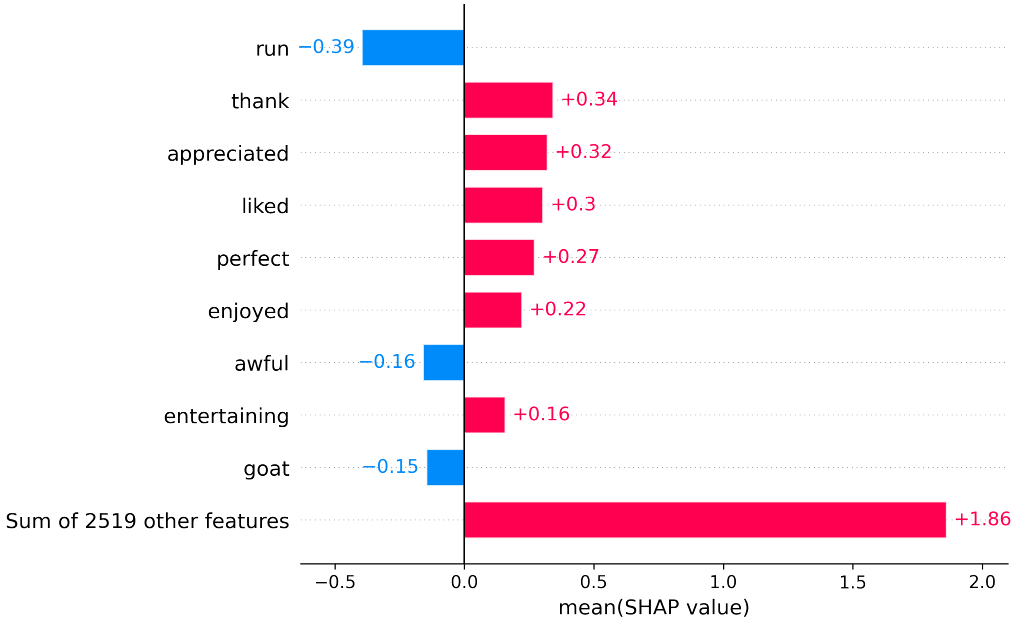


Figure 5.12: SHAP summary plot.

Another interesting way to interpret the NLP model is to look at the combined effect of certain tokens, which could also highlight the effect of word contextualization. Figure 5.13 shows how the model predicted the sentiment of one example review by ranking different groups of words base on their mean SHAP values.

From this plot, we see the words ”easy”, ”lecture”, ”were” and ”recorded” together gave a relatively high positive contribution to the sentiment score. The group that gave a relatively strong negative contribution is the one that contained the word ”didn(’t)”, which is also within our expectation. Because the selected review used in this example is strongly positive in sentiment (with a score greater than 98%), it makes sense that SHAP found most of the words, as well as the combined effect of the words, had a positive impact on the prediction score. This observation also indicates the sentiment analysis model built in the study was able to learn the right patterns between the utterances and the sentiment label.

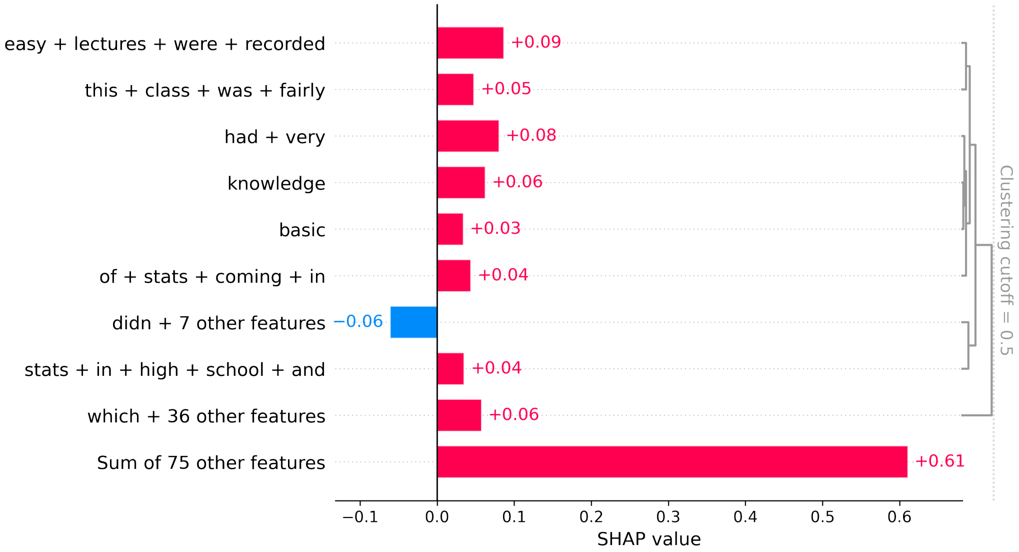


Figure 5.13: Combined-effect SHAP summary plot.

CHAPTER 6

Discussion and Conclusion

6.1 Summary

In this paper, we looked at the results from applying the sentiment analysis pipeline on the Bruinwalk reviews. The process started with collecting and cleaning the course reviews from the Bruinwalk website, exploratory data analysis on the unlabeled corpus, followed by the automated labeling and the cross-domain validation steps, and it ended with pre-training a new sentiment analysis model using the automated annotated Bruinwalk reviews. This is the first time that the sentiment analysis pipeline is designed and applied to the Bruinwalk reviews, and as we saw in the previous chapters, the final pre-trained model (published on HuggingFace) achieved quite comparable results to the benchmarks. Saving the pre-trained model will allow future explorations on this topic, such as annotating more Bruinwalk reviews in the STEM fields, or to extend the current analysis to the entire UCLA community. There are some interesting results from the analysis: the Bruinwalk corpus contained slightly more positive reviews (60%) than the negative ones (40%); most of the reviews were written in the most recent years (2019 to 2021), and reviews from year 2020 and 2021 are more dominated by the positive sentiment. In addition, we found that the review sentiment has a positive association with the grade the reviewer received, and the positive sentiment rate varies among different major/specialization and courses, meaning the reviews for some courses could be highly polarized. Overall speaking, from the modeling results, the feedback from the undergraduates at UCLA suggested that most of the students had a positive experience taking courses in the STEM community.

6.2 Limitations and Future Steps

One major limitation of the current analysis, is that comparing to the official evaluation data from UCLA, the Bruinwalk corpus could be a more biased sample. This means the Bruinwalk sample might not represent the entire undergraduate STEM community at UCLA. One possible future direction is to apply the sentiment analysis pipeline to the official evaluation data, and compare the results to what we observed in this study. This would help us ”visualize” the differences between the two types of review data, and what's more important, since the feedback from the official evaluation data are assumed to be more formal and rigorous, the sample could better represent the true population, and hence the results generated from the sentiment analysis would be representative.

Another future direction is to generalize the current analysis to the entire STEM community at UCLA, or even to the entire UCLA community. Generalizing to the STEM community would help us understand the difference between the undergraduate and graduate community through student feedback, and generalizing to the entire UCLA community would enable us to have a wider variety of analyses, such as comparing the STEM community with the HASS community. However, extending the scope of the study would again require having a sample that is more representative of the true population, meaning it’s imperative to gain access to the formal evaluation data.

Although the official evaluation reviews are assumed to be more representative of the true STEM community at UCLA, the dataset will still be biased, since the reviews are always subjective and they can be highly polarized. In the future study, it will be helpful to collect some official statistics, such as the grade distribution from the Registrar, or the enrollment rate in different courses, and then use them to re-weigh the distributions obtained from the sentiment analysis via post-stratification (Holt and Smith, 1979).

Lastly, as we saw in the previous chapters, the Bruinwalk reviews didn’t come with the sentiment label, which implies a few possible improvements to the current Bruinwalk review system. One helpful feature to add on would be an overall score to the course (e.g., a rating between 1 to 5), or a sentiment tag (e.g., would or would not recommend taking the course), which would serve as a good summary that supplements the review contents. It would also be beneficial if a word auto-detection feature could be implemented so that reviews that are not relevant to the course contents (such as selling the textbooks), or reviews that contain explicit words, will be filtered out. Ensuring the quality of the reviews and the review forum would help maintain a positive and welcoming community, and thus encouraging more students to share their feedback to the community.

APPENDIX A

Dataset

Major/Specialization	Courses
Biology	LIFE SCI 7A, 7B, 7C, 30A, 30B, 40, 107; NEUROSC M101A, M101B, M101C, 102
Chemistry	CHEM 20A, 20AH, 20B, 20BH, 20L, 30A, 30B, 110A, 113A
Computer Science	COM SCI 31, 32, 33, 35L, 111, 118, 130, 131, 180, 181, M51A, M146, M152A
Electrical Engineering	EC ENGR 101A, 101B, 102, 110, 111, 113, 115A, 121B, 131A, 132A, 133A, 141, 170A
Mathematics	MATH 31A, 31B, 32A, 32B, 33A, 33B, 61, 115A, 115AH, 131A, 131AH, 142, 151A, 151B
Physics	PHYSICS 1A, 1AH, 1B, 1BH, 1C, 1CH, 4AL, 4BL, 110A, 110B, 115A, 115B, 115C; ASTR 81, 82, 117, 127, 140
Program in Computing	COMPTNG 10A, 10B, 10C, 16A, 40A
Statistics	STATS 10, 13, 20, 100A, 100B, 100C, 101A, 101B, 101C, 102A, 102B, 102C, 112, 140SL, 141SL, C151, C173, C183

Table A.1: Undergraduate STEM courses at UCLA selected in the analysis.

Variable	Description
Course	Course name in the format of ”Major/Specialization + course number”.
Review	Review of the course.
Professor	Instructor of the course.
Grade	Grade the reviewer received.
Quarter	Quarter the reviewer took the course. Four categories are available: Fall, Winter, Spring, and Summer.
Year	Year the reviewer took the course.
Date	Date (day, month, and year) the review was submitted on Bruinwalk.com.
Upvotes	Number of upvotes the review received.
Downvotes	Number of downvotes the review received.

Table A.2: Features collected for each Bruinwalk review.

APPENDIX B

Prior Work

Rank	Architecture	Accuracy
1	NB-weighted-BON + dv-cosine	97.4
2	XLNet	96.21
3	EFL	96.1
4	GraphStart	96.0
5	BERT large finetune UDA	95.8
6	BERT_large+ITPT	95.79
7	L MIXED	95.68
8	BERT_base+ITPT	95.63
9	BERT large	95.49
10	ULMFiT	95.4
11	Block-sparse LSTM	94.99
12	CEN-tpc	94.52
13	oh-LSTM	94.1
14	Virtual adversarial training	94.1
15	Nyströmformer	93.2
16	Modified LMU	93.2
17	DistilBERT	92.82
18	seq2-bown-CNN	92.33
19	BP-Transformer + GloVe	92.12
20	BCN+Char+CoVe	91.8
21	ToWE-SG	90.8
22	LSTM with dynamic skip	90.1
23	CNN+LSTM	88.9

Table B.1: Top 23 models from sentiment analysis on the IMDb dataset.

APPENDIX C

Hyperparameters

Architecture	batch_size	epochs	learning_rate	hidden_dim	n_heads
CNN+LSTM (IMDb)	32	3	1e-3	20	N/A
Transformer (IMDb)	16	5	1e-5	32	4
DistilBERT (IMDb)	32	2	5e-5	default	6
DistilBERT (Bruinwalk)	16	3	5e-5	default	6

Table C.1: Optimal hyperparameters selected for each model in the sentiment analysis pipeline.

As mentioned in Chapter 4, hyperparameter selection was applied to select the optimal setup of each model architecture. The selection was done through the grid search, which is to fit the models with all possible combinations of the given set of hyperparameters, and then compare the their performance based on certain metrics. Table C.1 shows the hyperparameters selected for each model in the sentiment analysis pipeline, using the accuracy as the primary evaluation metric, and the recall, precision, and F_1-score as the secondary metrics.

Bibliography

F. Amato, F. Guignard, S. Robert, and M. Kanevski. A novel framework for spatio-temporal prediction of environmental data using deep learning. *Scientific Reports*, 10(1), Dec 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-79148-7. URL http://dx.doi.org/10.1038/s41598-020-79148-7.

J. Camacho-Collados and M. T. Pilehvar. On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. *CoRR*, abs/1707.01780, 2017. URL http://arxiv.org/abs/1707.01780.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

H. B. Gonzalez and J. J. Kuenzi. Science, technology, engineering, and mathematics (stem) education: A primer. *Fas.org*, 2017. URL https://sgp.fas.org/crs/misc/R42642.pdf.

B. Heredia, T. M. Khoshgoftaar, J. Prusa, and M. Crawford. Cross-domain sentiment analysis: An empirical investigation. IEEE Press, 2016. doi: 10.1109/IRI.2016.28. URL https://doi.org/10.1109/IRI.2016.28.

G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015. URL http://arxiv.org/abs/1503.02531.

D. Holt and T. M. F. Smith. Post stratification. *Journal of the Royal Statistical*

Society. Series A (General), 142(1):33–46, 1979. ISSN 00359238. URL http://www.jstor.org/stable/2344652.

Y. Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014. URL http://arxiv.org/abs/1408.5882.

L. Kirtibas Singh and R. Renuga Devi. Student feedback sentiment analysis: A review. *Materials Today: Proceedings*, 2021. ISSN 2214-7853. doi: https://doi.org/10.1016/j.matpr.2020.10.782. URL https://www.sciencedirect.com/science/article/pii/S2214785320384054.

S. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions, 2017.

A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

A. Madaan, A. Setlur, T. Parekh, B. Póczos, G. Neubig, Y. Yang, R. Salakhutdinov, A. W. Black, and S. Prabhumoye. Politeness transfer: A tag and generate approach. *CoRR*, abs/2004.14257, 2020. URL https://arxiv.org/abs/2004.14257.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

S. Minaee, E. Azimi, and A. Abdolrashidi. Deep-sentiment: Sentiment analysis using ensemble of CNN and bi-lstm models. *CoRR*, abs/1904.04206, 2019. URL http://arxiv.org/abs/1904.04206.

T. Niu and M. Bansal. Polite dialogue generation without parallel data. *CoRR*, abs/1805.03162, 2018. URL http://arxiv.org/abs/1805.03162.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108.

B. H. Shekar and G. Dagnew. Grid search-based hyperparameter tuning and classification of microarray cancer data. In *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, pages 1–8, 2019. doi: 10.1109/ICACCP.2019.8882943.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.