EVALUATING SENTIMENT ANALYSIS MECHANISM FOR LABELLED AMAZON REVIEWS

by

Parth Trada, B.Tech

THESIS

Presented to the Faculty of

The University of Houston-Clear Lake

In Partial Fulfillment

Of the Requirements

For the Degree

MASTER OF SCIENCE

in Computer Science

THE UNIVERSITY OF HOUSTON-CLEAR LAKE

MAY, 2023

EVALUATING SENTIMENT ANALYSIS MECHANISMS FOR LABELLED AMAZON REVIEWS

by

Parth Trada, B.Tech

APPROVED BY

_____

Kewei Sha, Ph.D., Chair

_____

Yalong Wu, D.Sc., Committee Member

_____

Wei Wei, Ph.D., Committee Member

APPROVED/RECEIVED BY THE COLLEGE OF SCIENCE AND ENGINEERING

_____

David Garrison, Ph.D., Associate Dean

_____

Miguel A. Gozalez, Ph.D., Dean

Dedication

I dedicate this thesis to my family in Gujarat – my parents Suresh Trada and Varsha Trada, my little sister Krisha Trada, my grandfather Samjibhai Kyada, my uncle Dinesh Chothani, and my entire Trada family. Without your tremendous love and support, I would not be where I am today. Thank you to everyone who has supported me in my endeavors and lifted my spirits during my years spent studying abroad.

Acknowledgements

I would like to take this opportunity to express my gratitude to all those who has supported me throughout the course of this research work. Firstly, I would like to thank my thesis advisor Dr. Kewei Sha, for the essential advice, and feedback. That helped me in analyzing and finishing my thesis. I am also very thankful to my committee members Dr. Yalong Wu and Dr. Wei Wei for their support, time, and valuable suggestions.

Special mention to Ritu Kadve for constantly encouraging me in every step during the thesis. I would like to thank my father Mr. Suresh Trada, for his financial and moral support in completing my Master's. He motivates me to achieve all the success so far.

.

ABSTRACT

EVALUATING SENTIMENT ANALYSIS MECHANISM FOR LABELLED AMAZON REVIEWS

Parth Trada
University of Houston-Clear Lake, 2023

Thesis Chair: Dr. Kewei Sha

Sentiment analysis has become increasingly important in understanding customer opinions, feedback, and preferences towards products and services, particularly on marketplaces like Amazon. Researchers have proposed various techniques and algorithms for sentiment analysis. However, there still lacks a good guidance that can systematically direct data scientists to select appropriate algorithms and models, although a few efforts have been made. This thesis aims to fill the gap by presenting a comprehensive evaluation on different sentiment analysis mechanisms for labeled Amazon reviews. To achieve the above goal, we first prepare an accurately labelled Amazon review dataset through manually labeling. This builds a solid foundation for our evaluation. Then, we evaluate the effectiveness of popular mechanisms used in sentiment analysis, including both data preprocessing techniques such as Bag of Words (BOW), Term Frequency-Inverse Document Frequency (TF-IDF) weighting, spell correction, stemming, and lemmatization, and various sentiment analysis models such as K-Nearest Neighbors (KNN), Logistic Regression (LR), Support Vector Machine (SVM), Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT). These mechanisms were selected based on their prominence in the field of sentiment analysis, their potential to yield high-accuracy

results, and their representation of different designs. We conducted five experiments using a combination of above data preprocessing techniques and analysis models. Through these experiments, we aim to identify a set of optimal combinations of preprocessing techniques and classification models that demonstrate superior performance in sentiment analysis of labeled Amazon reviews.

The experiment results show that the use of BERT with BOW, TF-IDF, Spell Correction, and Lemmatization achieved the highest accuracy of 98.99%, outperforming other combinations. The addition of TF-IDF weighting, spell correction, stemming, and lemmatization improves the accuracy of four analysis models by about 6%, i.e., from 87.34% to 93.4% for KNN, from 86.6% to 94.22% for SVM, from 90.68% to 96.87% for ANN, and from 92.87% to 97.95% for LSTM. However, LR shows a comparatively lower accuracy ranging from 74.32% to 81.09% regardless different preprocessing techniques due to its limitations as a linear model, which may struggle to capture complex patterns and non-linear relationships in the sentiment data. This work provides insights into the effectiveness of different data processing and analysis mechanisms for sentiment analysis of labeled Amazon reviews. The findings can be applied to improve the effectiveness of customer review analysis to help achieve higher level of customer satisfaction, which can be essential in areas such as product and business strategy development.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I:

INTRODUCTION

1.1 Background and Significance

Sentiment analysis, commonly referred to as "opinion mining", focuses on locating and extracting subjective data from text. The main goal of sentiment analysis is to analyze the emotions and attitudes expressed in a given piece of text. Sentiment analysis has become increasingly important in recent years, particularly with the growth of social media and online review platforms. These platforms offer users the opportunity to express their opinions and share their experiences regarding various products and services. Within this context, sentiment analysis is a valuable tool for extracting insightful information from textual data. This information can be utilized by organizations to make intelligent decisions, improve the quality of their products or services, and enhance the customer experience. According to Kristine D'Arbelles et al. [2], if a verified purchaser has written a large quantity of positive online reviews, the company experiences a surge in its sales revenue. This comprehension enables businesses to identify emerging trends and shifts in customer sentiment. Therefore, the efficacy of sentiment analysis can significantly influence the success of the business.

Sentiment analysis can be performed using various preprocessing methods and machine learning models. Choosing the best approach is essential for organizations. The aim of this study is to systematically analyze all possible approaches and evaluate their effectiveness in order to assist businesses in performing efficient sentiment analysis. Inadequate sentiment analysis may result in the loss of valuable insights into customer opinions and preferences, leading to incomplete or inaccurate information that can hinder decision-making processes. Consequently, businesses may fail to meet customer needs, implement ineffective marketing strategies, and provide poor customer service,

ultimately resulting in decreased customer satisfaction, reduced sales, and a decline in business success. Therefore, effective sentiment analysis is crucial for businesses to remain competitive and maintain a positive relationship with their customers. This thesis, which evaluates sentiment analysis mechanisms for labeled Amazon reviews, will contribute to the selection of appropriate approaches for performing effective sentiment analysis.

1.1.1 Natural Language Processing

Sentiment analysis can be regarded as a subfield or branch of natural language processing (NLP). NLP enables computers to interpret, manipulate, and comprehend human language in a meaningful way. Organizations have access to large volumes of text data in various formats, such as text messages, emails, and social media newsfeeds, through different communication networks such as Amazon and Facebook Marketplace. To accurately analyze the true sentiments or emotions of this available data, organizations rely on NLP and respond in real-time to human communication. NLP plays a critical role in efficiently analyzing text and speech data by accurately handling the nuances in grammar, slang, and other irregularities present in day-to-day communications. Additionally, NLP is utilized in face-to-face customer communication applications, such as chatbots, which can automatically understand and sort customer queries, respond to frequently asked questions, and redirect to customer support if necessary.

NLP uses various techniques, including statistical and machine learning methods, to process human language. Researchers employ syntactic and semantic analysis methods to design frameworks that enable machines to comprehend conversational human language. By utilizing sample data, machine learning can be used to train a computer to enhance its efficiency. The intricacies of human language, such as sarcasm, metaphors,

diverse sentence structures, and deviations from grammar and usage norms, require years of human learning. By utilizing machine learning techniques, researchers can accurately recognize and comprehend the properties of human language from the outset using NLP. A subset of machine learning called deep learning teaches computers to learn and think like people. It makes use of a neural network, which is made up of data-processing nodes organized to resemble human brains. Deep learning allows computers to identify, and correlate intricate patterns in the incoming data.

1.1.2 Sentiment Analysis

Sentiment analysis is a technique used in NLP to determine the emotional tones of a text. This is a common method used by organizations to identify and group ideas regarding a certain good, service, or concept. Machine learning (ML), computational linguistics, and data mining are all used in sentiment analysis to mine text for sentiment and subjective information, such as whether it is expressing positive, negative, or neutral feelings.

There are different algorithms to implement the sentiment analysis models, depending on how much data is needed to analyze and how accurate the model needs to be. There are three main approaches used by sentiment analysis: rule-based, machine learning-based, and hybrid, which is a combination of rule-based and machine learning-based approaches.

- **Rule-Lexicon Based Approach**

The rule-lexicon based methodology uses preestablished lexicons to identify and categorize keywords. Lexicons are collections of words used to convey the intention, feeling, and tone of the text. A rule-lexicon based approach rates the emotional impact of various terms by assigning sentiment scores to positive, negative, and neutral lexicons. The algorithm searches for words from the lexicon to determine whether a sentence is

positive, negative, or neutral before calculating the sentiment score. To establish the overall sentiment, the final score is compared with the sentiment range. It is easy to set up a rule-based sentiment analysis system, but it is difficult to scale. For instance, we will need to continually add new words to the lexicon as you find new ways to express purpose in text input.

- **Machine Learning based Approach**

This method teaches computer software to recognize emotional sentiment from text using machine learning (ML) techniques and sentiment classification algorithms, including neural networks and deep learning. To create a sentiment analysis model that can accurately predict the sentiment in new data, it must first be built and frequently trained on known data. Data scientists employ sentiment analysis datasets with a lot of instances during the training process. The datasets are used as input by the ML software, which then trains itself to draw the planned result. The software distinguishes and calculates how alternative word arrangements affect the final sentiment score by training with a huge number of distinct cases. Because it accurately analyzes a large variety of text information, ML sentiment analysis is helpful. ML sentiment analysis can precisely anticipate the emotional tone of the communications as long as the software is trained with enough instances. A trained ML model, however, is unique to a single industry. As a result, sentiment analysis software that has been educated using marketing data cannot be utilized to monitor social media without being retrained.

- **Hybrid Approach**

Using ML and rule-based systems together allows hybrid sentiment analysis to function. To maximize efficiency and accuracy when determining contextual intent in text, it combines elements from both approaches. The rule-based system provides a set of pre-defined rules and logic, while the machine learning-based system learns from data to

improve its performance. The rule-based system provides a set of guidelines or constraints that the machine learning-based system must follow, which helps to reduce the risk of the machine learning-based system making incorrect decisions or predictions. Unfortunately, integrating the two distinct systems may not always be useful due to complexities in implementation, a lack of synergy between different techniques, and potential inconsistencies in sentiment predictions, leading to reduced accuracy and reliability of the overall results.

1.2 Motivation and Research Challenges

Evaluation is crucial to assessing the performance and reliability of sentiment analysis mechanisms. It helps understand the strengths and weaknesses of different approaches and identifying their limitations. This work aims to provide insights into the accuracy, robustness, and suitability of sentiment analysis methods for analyzing Amazon reviews. Generally speaking, there are two steps in sentiment analysis, processing and analysis. Preprocessing is an essential step in sentiment analysis as it involves cleaning, transforming, and organizing the data before analysis. This step is necessary to remove noise, standardize the data, and prepare a high-quality dataset for accurate sentiment analysis. The analysis phase helps us spot patterns and trends in customer sentiment and offer insightful data to organizations. Our evaluation targets to provide robust and credible results, enable meaningful comparisons among different sentiment analysis techniques, and provide valuable insights into the performance, reliability, and suitability of sentiment analysis mechanisms for analyzing Amazon reviews. The findings of this research will guide the selection and improvement of sentiment analysis techniques for analyzing Amazon reviews, which can ultimately aid businesses in making intelligent decisions to enhance their products or services based on customer feedback.

There are mainly two challenges to this project. First, obtaining a large-scale Amazon review dataset with high-quality labels is a big challenge. Large-scale review labeling can take a long time and requires human skill, which might cause mistakes and discrepancies in the labeled data. Second, for sentiment analysis, a variety of algorithms are available, including rule-based and machine learning-based techniques. To get accurate results, the right algorithm must be chosen for the targeted dataset. A significant research challenge is assessing how well various algorithms work for Amazon reviews. We focus on addressing the second challenge in this thesis. Choosing the right algorithm for sentiment analysis is essential to obtain accurate results because different algorithms perform differently with certain types of data and tasks. Therefore, selecting the right algorithm for sentiment analysis depends on the type of data and the task at hand. It is important to consider the strengths and limitations of each algorithm and choose the one that is best suited for the targeted dataset and the desired level of accuracy.

1.3 Research Design and Results

1.3.1 Research Design

To address the above challenges, a procedure of four steps are followed. Firstly, we will download Amazon review dataset from stanford amazon review dataset (SNAP). We then manually sample 20,000 reviews for 4 differennt categories, i.e. books, electronics, heath & beauty, and food. Each sample contains a single and subjective sentence from a review paragraph. In this thesis, we use sentence-based reviews instead of paragraph-based reviews. Sentences provide a more granular level of analysis, allowing for a more precise identification of sentiment, tone, and emotion. Furthermore, Sentences can capture subtle nuances in sentiment that may be missed when analyzing an entire paragraph. Secondly, the collected data will be pre-processed using standard NLP techniques such as tokenization, stop word removal, stemming, etc. Thirdly, the

sentiment analysis algorithms such as K-Nearest Neighbor (KNN), Logistic Regression (LR), Support Vector Machine (SVM), Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT) will be implemented on the pre-processed data using Python libraries such as NLTK and Scikit-learn. The models will be trained and validated using a cross-validation approach. Lastly, the limitations of the study will be discussed, and suggestions for future research will be provided. The methodology will ensure the data used is correctly labeled and authentic.

1.3.2 Research Results

The results of the experiments on the labelled Amazon reviews are summarized as follows. The results report accuracy with the models used for sentiment analysis, including KNN, SVM, LR, ANN, LSTM, and BERT. The results show that accuracy generally increases with the addition of TF-IDF, spell correction, stemming, and lemmatization techniques. The highest accuracy is achieved with the BERT model, i.e., 98.99%, using all preprocessing methods mentioned earlier. Overall, the results indicate that the use of advanced models and pre-processing techniques such as BERT and lemmatization significantly improves the accuracy of sentiment analysis on Amazon product reviews, but the selection of appropriate data preprocessing and machine learning algorithms certainly impacts the accuracy.

1.4 Organization of Thesis

Chapter I introduces the significance of sentiment analysis for Amazon review and presents our motivation in this study. The remainder of the thesis is organized as follows. Chapter II delivers an extensive literature review that covers the current trends in sentiment analysis and system design principles. Chapter III presents an overview of the dataset. Chapter IV illustrates the selected data pre-processing techniques, which include

different data cleaning and data transformation methods. Chapter V explains the selection of sentiment analysis models. Chapter VI showcases the design and implementation of an evaluation system, where we describe each of our experiments in detail. In chapter VII, we discuss and evaluate the experimental results of our research. Finally, we conclude the thesis and identify future work in Chapter VIII.

CHAPTER II:

RELATED WORK

2.1 Sentiment Analysis

Sentiment analysis is a multidimensional research area that includes diverse aspects such as techniques, methodologies, applications, and evaluation metrics. Peter Turney in his research [4] employed an unsupervised technique to categorize reviews by calculating their semantic orientation. Author's approach to sentiment analysis focuses on phrases that include adjectives or adverbs that carry a positive or negative sentiment based on the context. By utilizing this method, author was able to attain a 75% accuracy rate in categorizing reviews based on their semantic orientation. Another way to conduct sentiment classification is to analyze each sentence independently to determine its sentiment. It's essential to distinguish between subjective and objective data when conducting sentiment analysis. The research by Vandana Jagtap et al. [5] assumes that each sentence has a single sentimet. Jiani Zhang et al. introduced an innovative method for sentiment analysis, where they utilized a hierarchical neural network for categorizing sentiment at the aspect level. The neural network was designed to identify two key elements, namely the aspect category and the polarity of the sentiment. The text data was processed using recurrent neural networks as part of this approach. Bo Pang et al. [6] used multiclass approach for text categorization. This technique is used to find the sentiment of the entire text document and not an individual sentence or phrase. This research takes into account the star ratings (i.e 1 to 5) that are available to rate products on different e-commerce websites [7]. The reviews are classified as per the ratings initially and the multi-class method classify each review into multiple classes which is later used to categorize the entire dataset.

2.2 Evaluation of Sentiment Analysis

Many studies have evaluated different approaches for sentiment analysis, including machine learning algorithms, lexicon-based methods, deep learning techniques, and hybrid approaches, with the aim of improving sentiment analysis accuracy and performance. Authors in [8] evaluated sentiment analysis for amazon reviews using Multinomial Naive Bayesian (MNB) and support vector machine (SVM) classifiers, with TF-IDF, and Part of Speech (POS) tagging as their main preprocessing techniques. In contrast to our approach, they employed N-grams vectors instead of BOW. Their findings demonstrate that the SVM model achieved an accuracy rate of 82.27% using the TF-IDF technique on the Amazon product dataset. In a paper [9], the author examined the effectiveness of five different deep learning models, namely RNN, LSTM, Gated-LSTM, GRU, and Update Gate-RNN, in predicting the sentiment of mobile phone reviews on Amazon. The authors employed preprocessing methods like spell correction, stop word and punctuation removal, and lemmatization before comparing three different vectorization techniques, one of which was Word2vec (also known as Continuous Bag of Words). Their study found that the LSTM model achieved a high accuracy rate of 93.63%. The author in the paper [10] compared the results of decision trees and naive bayes algorithms for sentiment analysis using Amazon review datasets. The classifiers were trained on the Kindle dataset. Our approaches include more techniques and models as compared to [10]. In another study [11], researchers explored the impact of feature selection techniques and their combinations on sentiment analysis of dialectal Arabic. They examined the effects of various term weighting techniques, stemming, stop word removal, and feature models on the model's performance, which is similar to our proposed approach. Their findings indicate that the SVM classifier performed best in

terms of accuracy. Similarly, in the paper [12], the authors aimed to predict rating-based sentiment analysis on review texts and identified words with positive and negative effects on the "Health & Personal Care" category. The authors used Root Mean Squared Error (RMSE) as their evaluation metric, while we used accuracy for our evaluations. The authors of [13] explored various feature extraction and selection methods, including phrase level, single word, and multiword techniques, for sentiment analysis on the Amazon dataset. They utilized POS tagging to extract features, while we used lemmatization without POS tagging in our study. This is because POS tagging may not be as useful for identifying sentiment in short social media posts or tweets.

CHAPTER III:

DATASET

The development, examination, and validation of a system typically depend on the quality and structure of data used for building, operating, and maintaining the model. The overall performance of a model depends on the data used from the boundless and voluminous source of available data to a great extent. Many public data sources are available which are used by some researchers to design a sentiment analysis model. Publicly available dataset namely Blitzer's multi-domain sentiment data (Blitzer et al) [16] is used by Dang et al. [17]. Public product reviews by Epinions (epinions.com) [53] are also used by many researchers including Kharde and Sonawaner [18], Fahrni and Klenner [19].

Apart from above datasets, The Amazon review dataset from Stanford SNAP is considered one of the best datasets for sentiment analysis and other natural language processing tasks for several reasons. Firstly, the dataset is incredibly large, with over 9 million products, making it a comprehensive and diverse source of data. This allows for a wide range of analyses and evaluations to be performed, providing a more nuanced understanding of the sentiment expressed in reviews across multiple product categories. Secondly, the dataset is publicly available, meaning that it can be accessed and utilized by researchers and practitioners worldwide, allowing for open and collaborative research. Thirdly, the dataset includes a vast amount of metadata, such as reviewer ID, product ID, and helpfulness rating, which can be used for a more granular analysis and evaluation of the reviews. Lastly, the dataset has been widely used in academic research, with several papers [8, 9, 10, 12, 13] published based on its analysis, evaluation, and application. This has led to a standardization of methods and techniques, making it easier to compare and contrast results from different studies. Overall, the Amazon review dataset from Stanford

SNAP is a valuable resource for sentiment analysis and other natural language processing tasks, providing a comprehensive and diverse source of labeled data that can be utilized by researchers and practitioners worldwide.

The SNAP provides a public dataset of Amazon product reviews, which can be downloaded from their website. The dataset contains reviews of products in various categories such as books, electronics, and movies. To acquire the dataset, one can visit the SNAP website [54], and navigate to the Amazon reviews dataset page. The dataset is available in two formats: JSON and text. The JSON format includes additional metadata such as reviewer ID, product ID, and helpfulness rating as showed in Fig. 3.1, while the text format only includes the review text.


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the piano.
He is having a wonderful time playing these old hymns.  The music  is
at times hard to read because we think the book was published for
singing from more than playing from.  Great purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


Figure 3.1 SNAP review dataset in JSON format.

To convert the JSON data format into the CSV dataset, we use the built-in to_csv() method in Python to convert the data to a CSV file. After that, we created a separate dataframe using the "reviewText" and "overall" columns and discarded all the unnecessary columns as described in Fig. 3.2. Eventually, the "reviewText" column

refers to the "Reviews," and the "overall" column indicates the “Ratings”, indicating positive (1) and negative (0). Sample data is shown in Fig. 3.2.

Categories	Rating	Reviews
Electronics	1	amazon kindle is always the best ebook, upgrade every new model
Beauty	0	Battery blew up in the charger
Electronics	0	It does its job but I would buy one which the screen is brighter
Books	1	Even though you already know what the outcome will be, this is still an enjoyable and erotic read
Books	0	Gave up about half way through
Electronics	0	Looking at the picture and seeing it was 8th generation I assumed it would be a great device
Books	0	I read the other reviews and decided to give it a try despite the review labeling it brutal
Electronics	1	This kindle is light and easy to use especially at the beach!!!
Beauty	1	Work Great, Great Value

Figure 3.2 Sample entries in our Amazon review dataset

After acquiring the dataset, it is essential to ensure that the data is labelled correctly and authentic. For this thesis, we manually analyze the context of the 20,000 reviews, where reviews fall into one of four categories (i.e., books, electronics, health & beauty, and food), and classify them as 1 for positive and 0 for negative. Based on the data presented in Table 3.1, it appears that the categories of beauty and electronics have a relatively lower number of negative reviews compared to the other categories. This is because it is really hard to find a negative review on the SNAP website for these two categories. In order to balance the ratio of positive and negative reviews in the dataset, two categories with higher numbers of negative reviews were selected, i.e., books and food. Table 3.1 includes the number of positive and negative reviews for each category.

Table 3.1 Overview of dataset categories

Category	Positive Reviews	Negative Reviews	Total Reviews
Beauty	4,306	694	5,000
Books	2527	2473	5,000
Food	2163	2837	5,000
Electronics	4686	314	5,000

Further, we validated our classification with other people to make our dataset more authentic. The decision to manually label the samples was made for several reasons. It is challenging to locate reliable and high-quality datasets that are readily accessible. Manual labeling allows for greater control over the quality of the data. While automated sentiment analysis tools can be useful for quickly processing large volumes of data, they may not always accurately capture the sentiment expressed in a review. By manually labeling the samples, it was possible to ensure that the sentiment was accurately captured. Manual labeling allows for greater flexibility in the labeling process. By defining the categories and criteria for labeling, it was possible to ensure that the data was labeled in a way that was specific to the research questions being asked, rather than relying on a pre-defined labeling scheme. Overall, manual labeling was necessary for this thesis to gain a more nuanced understanding of the sentiment expressed in the reviews, train and evaluate machine learning models, ensure the quality of the data, and have flexibility in the labeling process.

In this thesis, we use sentence-based reviews instead of paragraph-based reviews for two main reasons. First, sentences provide a more granular level of analysis, allowing for a more precise identification of sentiment, tone, and emotion. Sentences can capture subtle nuances in sentiment that may be missed when analyzing an entire paragraph. Second, sentence-based sentiment analysis can be more practical and efficient for real-

time applications due to less processing, whereas processing paragraphs takes comparatively longer.

CHAPTER IV:

SELECTION OF DATA PREPROCESSING TECHNIQUES

4.1 Data Cleaning

Data cleaning [20] is a critical step in the data mining [21] as it prepares high quality data for further analysis and modelling. The goal of data cleaning is to clean, transform, and normalize the data into a suitable format for processing by data analytical algorithms and models. Text data contains noise in various forms like emotions, punctuation, text in a different case. The challenge in dealing with human language is that there are multiple ways to express the same idea, which poses a difficulty for machines as they require numerical inputs rather than words, making it crucial to efficiently convert text into numbers.

4.1.1 Lower Casing

Lowercasing is a common step in text preprocessing for NLP as it helps to standardize the text data. The motivation for performing lowercasing is to ensure consistency in the text data and reduce the dimensionality of the data. In NLP, text data can come from various sources, including web pages, social media, and other sources, and may contain a mixture of upper and lowercase characters. This inconsistency can cause problems when analyzing the text data as the same words may be represented in different ways. For example, "Apple" and "apple" would be considered different words by most NLP algorithms. Lowercasing helps to resolve this issue by converting all the text data to lowercase [22], which ensures that all the words in the text data are represented in the same way. This makes the analysis more effective and consistent, as the same words will be treated as the same, regardless of their case. In addition, lowercasing also helps to reduce the dimensionality of the text data by eliminating the difference between upper and lowercase characters. This can lead to more efficient and

accurate results in NLP tasks, as the algorithms will have to consider a smaller number of unique words. Overall, lowercasing is an important step in text preprocessing for NLP as it helps to standardize the text data and improve the accuracy and efficiency of NLP algorithms.

4.1.2 Removal of punctuations

Removing punctuation is a common step in text preprocessing for NLP as it helps to eliminate noise in the text data and improve the accuracy of NLP algorithms. The motivation for removing punctuation is to ensure that the text data is focused on the content of the words and not the surrounding characters. Punctuation marks, such as commas, periods, and exclamation points, can add information to the text data, but they can also interfere with the analysis of the text data. For example, punctuation marks may break up words and cause the same word to be treated as different words by NLP algorithms. In addition, punctuation marks can also affect the frequency of words, which is an important factor in many NLP tasks, such as text classification and sentiment analysis. Removing punctuation helps to eliminate these issues by focusing the text data on the content of the words. This allows NLP algorithms to analyze the text data based on the meaning of the words and not the surrounding characters [23]. This can lead to more accurate and effective results in NLP tasks, as the algorithms will have a clearer understanding of the content of the text data. In addition, removing punctuation can also help to reduce the dimensionality of the text data, as the number of unique words will be reduced. This can make the analysis more efficient and improve the performance of NLP algorithms. Overall, the elimination of punctuation is a crucial step in text preprocessing as it minimizes textual noise and increases NLP algorithm precision, with the exception of meaningful punctuations. It is not recommended to eliminate all punctuation marks as certain ones can convey meaning and sentiment. For instance, the sentence "I am

happy!!!" would lose its intensity if we removed the exclamation marks, reducing it to "I am happy." As a result, it is important to exercise caution in determining which punctuation marks to remove and which ones to preserve during sentiment analysis.

4.1.3 Removal of stopwords

Stop words are commonly occurring words in a language that do not add significant meaning to the text and are often removed in NLP preprocessing. They can safely be ignored without sacrificing the meaning of the sentence. For some search engines, these are some of the most common, short function words, such as the, is, at, which, and so on. In this case, stop words can cause problems when searching for phrases that include them, particularly in names such as “The Who” or “Take That”. The removal of stop words helps to reduce the dimensionality of the text data and makes it easier to process and analyze. Additionally, removing stop words can also improve the performance of NLP models by reducing noise and highlighting the important words that carry more meaning. This is particularly useful in tasks such as text classification, sentiment analysis, and topic modeling, where focusing on the key words can lead to more accurate results. Once we have split text into tokens, it often becomes clear that not all words carry the same amount of information, if any information at all, for a predictive modeling task. It is common advice and practice to remove stop words for various NLP tasks. Stopwords are the words in any language which does not add much meaning to a sentence.

4.1.4 Removal of frequent and rare words

The removal of high and low frequent words, in text preprocessing in NLP serves a similar purpose as the removal of stopwords. Frequent words are those that appear very frequently or very rare in the text and do not add much meaning to the overall content. Removing frequent words can help to reduce the dimensionality of the text data, making

it easier to process and analyze. By reducing the impact of words that appear frequently but do not convey much information, NLP models can focus on more meaningful and unique words, leading to improved results in tasks such as text classification, sentiment analysis, and topic modelling. Additionally, removing frequent words can also help to mitigate the impact of text data that is skewed towards certain common words and can help to prevent overfitting of NLP models.

The Counter library from the collection package gives a list of most common and least common words from the review corpus. A corpus refers to a large and structured collection of written or spoken texts that are used as a source of linguistic information and analysis. The structure of the collection allows for efficient access and retrieval of the texts, enabling linguistic analysis and the identification of patterns or trends in language use. Counter is a subclass of dict that is specially designed for counting hashable objects in Python. In Python, **dict** (short for dictionary) is a built-in data structure that represents an associative array. It is an unordered collection of key-value pairs, where each key must be unique within the dictionary. The keys of a dictionary must be hashable objects, such as integers, strings, or tuples, while the values can be of any data type, including other dictionaries. To count with Counter, we typically provide a sequence as an argument. This results in a dictionary-like object that shows the frequency of each unique element in the sequence.

4.1.5 Removal of HTML and URLs

The web generates tons of text data and this text might have URLs and HTML tags in it. These unneccessary tags and URLs do not add any value to text data and only enable proper browser rendering. For example, we are using Amazon Review dataset, then there is a good chance that the review will have some URLs in it. These URLs do not add any value to the meaning of sentence so we should remove that from the corpus.

The removal of URLs in text preprocessing is done to improve the performance of NLP models and to make the text data more relevant to the task at hand. Additionally, removing irrelevant tags and information can also help to reduce the dimensionality of the text data and make it easier to process and analyze. This step can also help to ensure that the text data is in a standard format and is free of any noisy or distracting information, making it easier to work with and interpret.

4.1.6 Spell Correction

The main reason for spell correction in text preprocessing is to improve the accuracy and reliability of sentiment analysis models by correcting spelling mistakes and typos in the text data. Spelling mistakes and typos can have a significant impact on the performance of SA models, especially in tasks such as text classification, sentiment analysis, and topic modeling. By correcting spelling mistakes and typos, SA models can focus on the content of the text and can produce more accurate results. Additionally, spell correction can also help to ensure that the text data is in a standard format and is free of any distracting or confusing errors, making it easier to work with and interpret. Furthermore, spell correction can also help to mitigate the impact of data sparsity, as SA models are trained on a more representative set of text data that is free of errors. This can lead to improved generalization and accuracy of SA models.

4.1.7 Stemming and Lemmatization

Stemming is the process of reducing words to their root form, which can help to identify words that are semantically similar or related to each other. By reducing words to their root form, NLP models can better capture the meaning of the text and can produce more accurate results in tasks such as text classification, sentiment analysis, and topic modeling. Additionally, stemming can also help to reduce the impact of data sparsity, as SA models are trained on a more representative set of text data that captures the core

meaning of the words. This can lead to improved generalization and accuracy of models. Furthermore, stemming can also help to make the text data more manageable and easier to process, as it reduces the number of unique words in the text data, which can be especially important when dealing with large amount of text data.

Lemmatization is the process of reducing words to their base form, which is also known as their lemma. Unlike stemming, which reduces words to their root form, lemmatization reduces words to their core meaning, taking into account the context and the grammatical structure of the words. This can help to identify words that are semantically similar or related to each other, and to capture the core meaning of the text. By reducing words to their base form, NLP models can better capture the meaning of the text and can produce more accurate results in tasks such as text classification, sentiment analysis, and topic modeling. Similar to stemming, lemmatization can also help to make the text data more manageable and easier to process, as it reduces the number of unique words in the text data, which can be especially important when dealing with large amounts of text data.

4.2 Data Transformation

Vector space models are a common way to represent text data for sentiment analysis. They aim to transform the text into a numerical vector, which can then be used as input to machine learning or deep learning models for sentiment classification. For example, documents and queries are represented as vectors in equation 1, where $w_{i,d}$ and $w_{i,q}$ represents weights of the terms in the document (d) and the query (q) respectively.

$$d = \left(w_{1,d}, w_{2,d}, \dots, w_{n,d} \right), \quad q = \left(w_{1,q}, w_{2,q}, \dots, w_{n,q} \right) \tag{1}$$

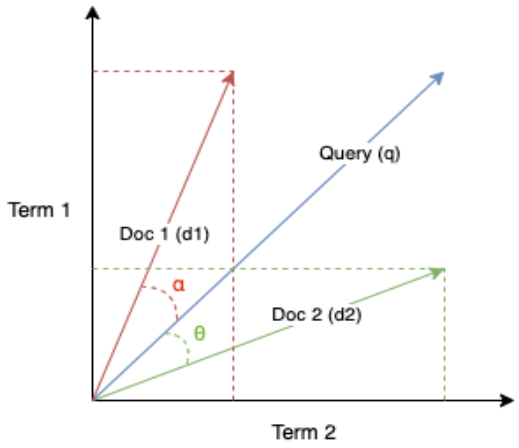

Figure 4.1 vector representation of Vector space model

Each dimension corresponds to a separate term in Fig 4.1. If a term occurs in the document, its value in the vector is non-zero. The definition of terms depends on the application. Typically terms are single words, keywords, or longer phrases. If words are chosen to be the terms, the dimensionality of the vector is the number of words in the vocabulary (the number of distinct words occurring in the corpus). The vector space model is an algebraic model that represents objects (like text) as vectors. This makes it easy to determine the similarity between words or the relevance between a search query and document. Several different ways of computing these values, also known as (term) weights, have been developed. Two best known schemes are Bag of words (BOW) and Term Frequency-Inverse Document Frequency (TF-IDF) weighting.

The similarity between the document vector and query vector is measured in the Fig. 4.1, and the documents are ranked based on the measure. One of the most popular and common ways to measure the similarity is known as cosine rule. The logic behind the cosine rule of ranking is as follows. Given a query vector, the highest ranked document should be the document that is the closest to the query in angular sense. When two vectors are identical then the angle, θ between them would be zero, i.e., $\cos(\theta) = 1$ since

$\theta = 0$. Similar documents with the query vector will have higher scores. The cos rule for ranking the documents is given below in the equation 2.

$$\cos\theta = \frac{d_i \cdot q}{\|d_i\| \cdot \|q\|} \tag{2}$$

Where the numerator of the equation represents the dot product of the document (d_i where i=1,2…n) and the query (q) vectors. The denominator of the equation represents the product of the norm of vector d_i, and the norm of vector q. Using the cos, the similarity between document d_j and query q can be calculated as.

$$cos\ (q, d_j) = \frac{1}{w\ (q) * w(d)} \sum_{t=1}^{n} w\ (q,t) * w(d,t) \tag{3}$$

In the above equation 3, $w_{q,t}$, $w_{d,t}$ denote the weights of the terms in the query and the document respectively, where t represents term in the equation.

According to Abilhoa and De Castro (2014) [13], the frequencies of terms can be binary, absolute, relative, or weighted. Algorithms such as binary, Term Frequency (TF), TF–IDF, etc. are used in traditional term weighting schemes. BOW and TF-IDF are widely used vector space models in sentiment analysis due to their simplicity, effectiveness, and ability to handle large volumes of data.

4.2.1 BOW

The BOW model is a simple and effective way to represent text data. It involves creating a matrix where each row represents a document, and each column represents a word in the corpus. The cells of the matrix represent the frequency of the word in the corresponding document. For example, suppose we have the following two sentences:

1. "The movie was great, and the acting was fantastic."
2. "I did not like the movie at all."

To apply the BOW model, we first create a vocabulary of unique words in the corpus. Then, we create a matrix where each row represents a document and each column represents a word in the vocabulary. The cells of the matrix represent the frequency of the word in the corresponding document. The resulting matrix for the above sentences showed in beloew Table 4.1:

Table 4.1 BOW vector representation

	and	acting	at	did	fantastic	great	I	like	not	the	was	all
Sentence 1	1	1	0	0	1	1	0	0	0	2	2	0
Sentence 2	0	0	1	1	0	0	1	1	1	1	0	1

4.2.2 TF-IDF

While BOW is a useful technique, it has some limitations. One of the main limitations is that it treats all words as equally important, which may not always be true. Some words may be more important than others in determining the sentiment of a review.

To address this issue, we can use the TF-IDF technique. which, we not only consider the frequency of a word in a review, but also the frequency of the word in the entire dataset. TF-IDF is calculated as follows:

$$TF = \frac{Number\ of\ times\ a\ word\ appears\ in\ review}{Total\ number\ of\ words\ in\ the\ review} \quad (4)$$

$$IDF = log\left(\frac{Total\ number\ of\ reviews}{Number\ of\ reviews\ containing\ the\ word}\right) \quad (5)$$

$$TF\text{-}IDF = TF\,x\,IDF \quad (6)$$

The TF-IDF value for a word in a review is higher if it occurs frequently in the review, but infrequently in the entire dataset. This means that the TF-IDF technique gives more weight to words that are rare in the dataset but important in a particular review. For example, suppose we have a corpus of three documents:

1. "The movie was great, and the acting was fantastic."
2. "I did not like the movie at all."
3. "The movie was long and boring."

The IDF for the word "movie" would be log(3/3) = 0, because it appears in all three documents. The IDF for the word "acting" would be log(3/1) = 1.58, because it only appears in one document.

To apply the TF-IDF model, we first create a vocabulary of unique words in the corpus. Then, we create a matrix where each row represents a document and each column represents a word in the vocabulary. The cells of the matrix represent the TF-IDF score of the word in the corresponding document. The resulting matrix for the above corpus showed in below Table 4.2:

Table 4.2 TF-IDF vector representation

	the	movie	was	great	and	acting	fantastic	long	boring
Sentence 1	0	0.6	0.6	1.58	1.58	0	0	0	0
Sentence 2	0	0	0	0	0	0.79	1.58	1.58	1.58
Sentence 3	0	0	0	0	0	0	0	1.58	1.58

CHAPTER V:

SELECTION OF SENTIMENT ANALYSIS MODELS

This thesis focus on machine learning based sentiment analysis, which uses machine learning algorithms along with linguistic features to identify the sentiment in the text [24]. Given a set of data, machine learning algorithms focus on learning models from the data [25].

The supervised machine learning approach [27] is a type of machine learning approach where the algorithm learns to make predictions or decisions based on labeled training data. There are many different types of supervised machine learning algorithms such as linear regression, decision trees, Support Vector Machine (SVM) , K-Nearest Neighbor (KNN), and neural networks. Among the different supervised learning techniques SVM, KNN, Logistic Regression (LR), Artificial Neural Network (ANN), Long Short-Term Memory networks (LSTM), and Bidirectional Encoder Representations from Transformers (BERT) are some of the most popular techniques which are employed in the sentiment analysis process [17]. These algorithms are implmeneted over other algorithms because of their unique capabilities. LR and SVM are popular and efficient algorithms for binary classification tasks, while KNN can work well with small to medium-sized datasets. ANN is a versatile algorithm that can be used for a wide range of tasks, including sentiment analysis. LSTM, and BERT are a type of neural networks that are particularly effective for analyzing sequences of data, such as text, due to its ability to capture long-term dependencies. Among these learning techniques, SVM is Kernel based [28], and LR is regression-based technique. KNN is non-parametric, ANN and LSTM are neural network-based, while BERT is a type of deep learning technique. We use libraries like sklearn, pandas, and numpy to implemet these algorithms in Python. A brief introduction to each of these techniques is presented below.

1. SVM

SVM is a kernel-based classifier that has gained popularity in different regression and classification problems. Many researchers indicated that Gaussian kernel, and Radial Basis kernel function (RBF) performs better for sentiment analysis [29, 30]. The main difference between the two is that a Gaussian-based kernel is a specific type of RBF kernel. A Gaussian-based kernel uses the Gaussian function as its kernel function, which is a bell-shaped curve that assigns a weight to each data point based on its distance from the center of the kernel. Ultimately, the choice of kernel function depends on the specific problem and the data at hand. In the case of a very large dataset, the linear kernel function proves to be the best for text classification among all other different kernel functions used in the SVM classifier [31]. The linear kernel function is represented as given in equation 7:

$$K\left(x_i, x_j\right) = x_i^T x_j \tag{7}$$

where x_i and x_j are the input space vector and x_i^T is the transpose of x_i

2. LR

Logistic Regression [32] is used most commonly for binary dataset. Independent variables are examined in order to make the forecast. The independent variables in the instance of a positive or negative review can be glad, disagree, like, etc., with the results falling into one of two groups. In sentiment analysis, the characteristic or input that aids in sentiment prediction is the independent variable, and the sentiment we are attempting to predict is the dependent variable. We may develop precise sentiment analysis models to assist us better comprehend people's attitudes and opinions toward various topics and products by examining the relationship between these factors. The dependent variables

are always categorical, but the independent variables may be numeric or categorical, and they are written as given in equation 8:

$$P\left(Y = \frac{1}{x}\right) \; or \; P\left(Y = \frac{0}{x}\right) \tag{8}$$

Given the independent variable X, it determines the approximation of the dependent variable Y. In our model, this method can be used to determine if the review has a positive sentiment or a negative sentiment.

3. KNN

KNN [33] is a data classification algorithm that attempts to determine what group a data point is in by looking at the data points around it. KNN algorithm works by calculating the distance between a new data point and all other data points, and then selecting the k-nearest data points based on this distance to predict the output of the new data point. KNN is an example of a lazy learner algorithm [34] because it does not explicitly learn a model from the training data. Instead, it stores the entire training dataset in memory and waits until a new data point needs to be classified. When a new data point arrives, the KNN algorithm retrieves the k-nearest neighbors (i.e., the k training instances that are most similar to the new instance) from the stored training data, and assigns the class label that is most frequent among these neighbors to the new instance. This makes KNN very easy to implement for data mining.

4. ANN

ANN [35] is a machine learning classifier that is designed based on the biological brain. In ANN, a set of fundamental processing units, known as neurons, are connected and organized according to specific tasks. The network topology, weights between the neurons, activation function, bias, momentum, etc. form the basis of learning in ANN.

Compared to traditional ANN, deep ANN or deep learning has emerged as a powerful technique in the context of sentiment analysis.

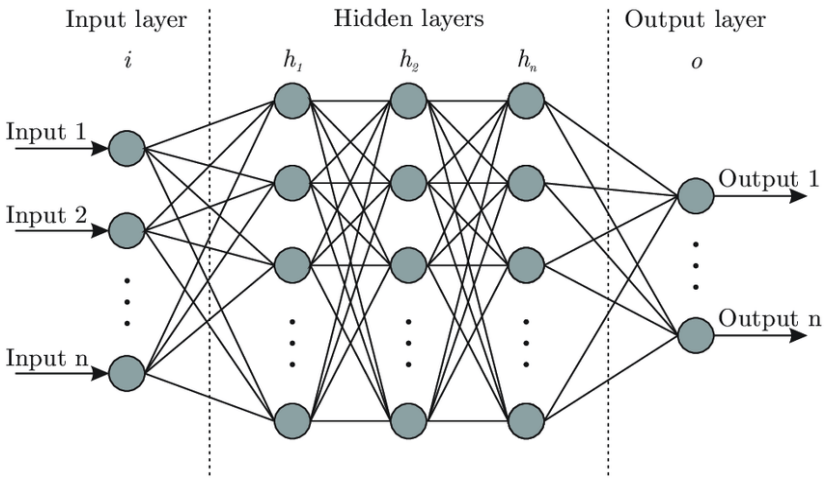

Figure 5.1 Artificial Neural Network architecture

The architecture of an ANN [36] is composed of several layers of neurons as shown in Fig 5.1 [55]. Each layer performs a specific task in the processing of the input data. The input layer is responsible for receiving the input data, which is then passed through one or more hidden layers. The hidden layers perform complex calculations on the input data, and output the results to the next layer in the network. The output layer is responsible for producing the final output of the network, which can be a prediction, classification, or other type of output depending on the application.

5. LSTM

Long Short Term Memory networks – usually just called LSTM [37] - are a special kind of RNN, capable of learning long-term dependencies. Hochreiter and Schmidhuber first proposed recurrent neural networks, which were subsequently developed and widely adopted by others. One notable feature of these networks is their

ability to store information over extended periods of time, which is a fundamental aspect of their operation. In general, recurrent neural networks are composed of a sequence of identical neural network modules that are connected in a chain-like fashion. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer [40].

LSTM contains a similar architecture except two additional layer called bidirectional layer and embedding layer. Bidirectional layer [38] processes the input sequence in both directions (forward and backward) using two parallel LSTM layers. This allows the model to capture both the past and future context information of each word in the sequence. Embedding layer takes the input sequence of integers and converts each integer into a vector of fixed size by looking up the corresponding vector representation from a pre-trained embedding matrix.

6. BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers [39], is a type of Transformers, a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection. In NLP, this process is called attention [38]. BERT is designed to help computers understand the meaning of complex language in text by using surrounding text to establish context. The BERT framework was pre-trained using text from Wikipedia and can be fine-tuned with other datasets.

LSTM has ability to retain the selective information over time, and thus is commonly used for sequential data processing. On the other hand, BERT is a pre-trained transformer-based neural network architecture that uses a multi-layer bidirectional approach to learn contextual relationships between words in a text. Due to this, BERT has

some edge over LSTM in terms of higher coplexity, and capcity. In summary, LSTM is a type of RNN that can selectively retain information over time, while BERT is a transformer-based pre-trained neural network that uses a bidirectional approach [40] to understand the context of words and sentences better.

CHAPTER VI:

DESIGN & IMPLEMENTATION OF EVALUATION SYSTEM

Sentiment analysis algorithms can be divided into three categories as described in Fig. 6.1 as Rule Lexicon-Based algorithms [43], machine learning algorithms, and Hybrid algorithms. Rule-lexicon algorithms have a predetermined dictionary of words along with their valence scores. These valence scores are then used to calculate the sentiment score. This is a fast approach for sentiment analysis but not accurate. Machine learning algorithms especially deep lerning algorithms [44] however are generally more accurate than the Rule Lexicon-Based algorithms. Additionally, machine learning algorithms can automatically identify relevant features or patterns in the data, whereas rule lexicon-based algorithms require manual feature engineering. In case of Hybrid algoritms, combining the two separate systems may not always be beneficial due to the difficulties in implementation, a lack of compatibility between various methods, and the possibility of disparities in sentiment predictions, which could result in lower precision and dependability of the ultimate outcomes. The choice of algorithm will depend on the specific requirements of the analysis task and the availability of labeled training data.

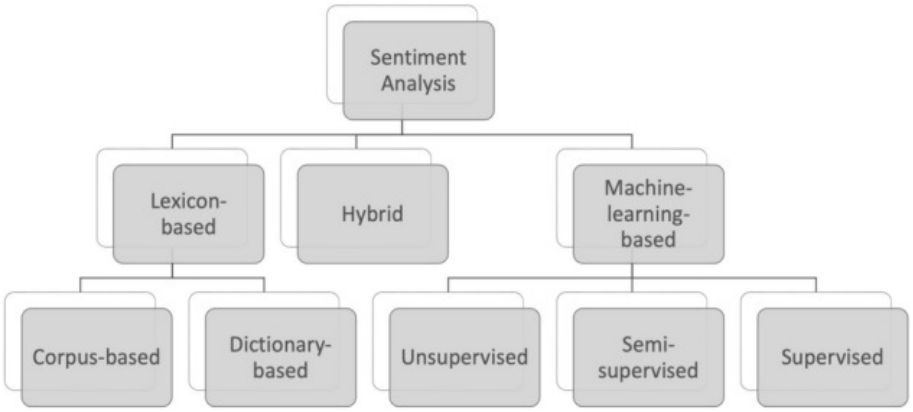

Figure 6.1 Classes of Sentiment Analysis algorithms

An approach, on the other hand, is a more general term that refers to a broader strategy or methodology used to solve a problem or address a research question. Approaches can involve multiple algorithms and techniques. Sentiment analysis approaches can be broadly classified into four types. Polarity-Based [41], Beyond Polarity-Based, Aspect-Based [27], and Document or Sentence level. The Polarity-Based approach considers the sentiment score of product reviews and classifies the review into positive, negative, or neutral. This technique can be extended in Beyond Polarity-Based to detect complex emotions such as enjoyment, disgust, anger, and sadness. However, detecting such emotions would require expert knowledge of English grammar. Aspect-Based sentiment analysis is another approach that is used to find the various aspects or features related to a product and identify the sentiment associated with it. Document-level sentiment analysis is mostly used by healthcare and finance companies to fulfill their audit requirements. In this thesis, we have performed sentence level sentiment analysis for amazon review dataset because paragraph-level sentiment analysis may overlook subtle differences in sentiment expressed in each sentence, which can lead to inaccurate results [42].

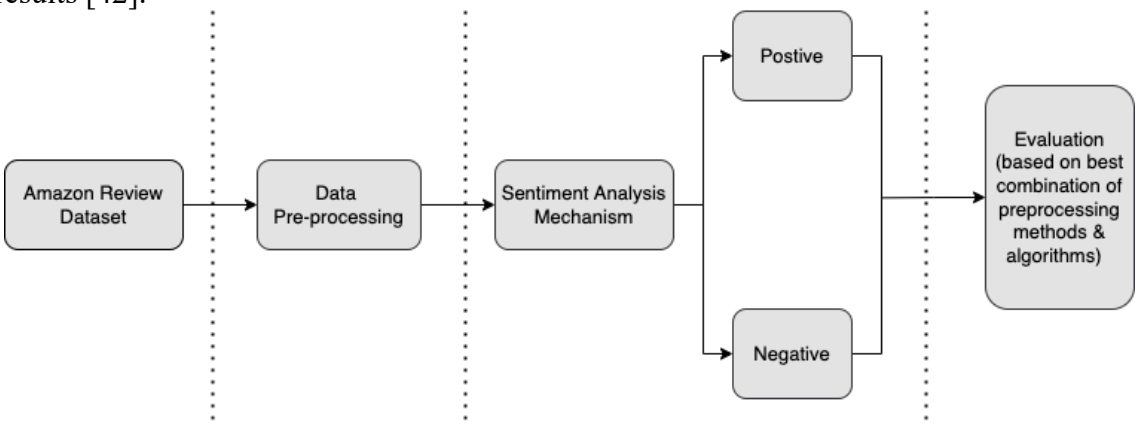

Figure 6.2 Main workflow of sentiment analysis

After going through more than 500 sentiment analysis frameworks proposed till now, a general framework of sentiment analysis is summarized in Fig. 6.2. The framework comprises mainly four steps. These steps perform collection and standardization of data, pre-processing of the dataset [13, 18, 20], prediction or classification of the sentiments associated with the keywords or the whole sentence or document, and summarization of the overall sentiment associated with the dataset

6.1 Design of evaluation experiments

The major goal of this thesis is to evaluate the performance of sentiment analysis mechanisms through a serious of well-designed experiments. The main aim of these experiments is to determine the most effective combination of pre-processing techniques and machine learning models for sentiment analysis. The experiments are designed to evaluate the performance of different combinations of pre-processing techniques and machine learning models for sentiment analysis. By comparing the results of each experiment, we can determine which combination is most effective in accurately predicting the sentiment of the reviews. This information can be used to develop more accurate sentiment analysis models in the future. Additionally, by gradually adding more pre-processing techniques to the models, we can understand the impact of each technique on the performance of the model [45].

In this thesis, there are five experiments conducted, each using a specific combination of pre-processing techniques and machine learning models. The first experiment uses only BOW, while the second experiment adds the TF-IDF model in addition to BOW. The third experiment includes spell correction in addition to BOW and TF-IDF, while forth experiment adds stemming to this combination. Finally, the fifth experiment includes lemmatization along with BOW, TF-IDF, and spell correction. Table 6.1 showcases the summary of experiments with purpose.

Table 6.1 Summary of experiments

Experiments	Purpose
BOW	To evaluate the performance of different machine learning algorithms on the dataset using BOW
BOW + TF-IDF	To evaluate the effectiveness of combining two pre-processing techniques, BOW and TF-IDF
BOW + TF-IDF + Spell Correction	To examine the impact of misspelled words in the text data
BOW + TF-IDF + Spell Correction + Stemming	To evaluate the impact of stemming on the performance of the classification models
BOW + TF-IDF + Spell Correction + Lemmatization	To evaluate the influence of lemmatization

For each experiment, the dataset is pre-processed with the data cleaning techniques as discussed in chapter III, and fed into six selected machine learning models. These models are then trained on the pre-processed dataset and tested on a validation set to evaluate their performance. The results of each experiment are then compared to determine the best combination of pre-processing techniques and machine learning models for sentiment analysis on this particular dataset. The performance of each algorithm is evaluated using metric as accuracy. Accuracy [46] is a commonly used evaluation metric in sentiment analysis because it measures the proportion of correctly classified sentiments out of the total number of sentiments in the dataset. Since the sentiment analysis task is a classification problem, accuracy provides a simple and intuitive measure of how well the model is performing. The motivation for these experiments is to compare the performance of different machine learning algorithms on the sentiment analysis task using a simple vector space model like BOW. The results of

this experiment provide insights into the effectiveness of each algorithm and help to identify the best-performing algorithm for the sentiment analysis task.

The choice of pre-processing techniques depends on various factors such as the type of data, the goal of analysis, and the available resources. Other pre-processing techniques such as part-of-speech tagging, named entity recognition, and sentiment lexicon-based methods can also be used depending on the specific task and data. However, in the given context of sentiment analysis, BOW, TF-IDF, Spell Correction, Stemming, and Lemmatization are commonly used and have been found to be effective in various studies, hence the selection of these techniques for the experiments [48]. BOW and TF-IDF are widely used techniques in text analysis due to their simplicity and effectiveness in representing the text data. Spell correction helps in fixing common spelling errors in the text, which can improve the accuracy of the analysis. Stemming and lemmatization [47] are also popular techniques for text normalization, which can help in reducing the dimensionality of the data and improving the performance of the machine learning models.

6.1.1 BOW

This experiment is designed to evaluate the performance of different machine learning algorithms on the dataset using Bag-of-Words (BOW) [49] representation for text. BOW is a simple and popular vector space model for text representation that considers the frequency of words in a document and ignores their order. In this experiment, the raw text data is first preprocessed by tokenizing the text, removing stop words, and converting the text to lowercase. Then, the text is converted to numerical vectors using BOW. The resulting vectors represent each document as a high-dimensional vector in the feature space. The next step is to apply various machine learning algorithms to the BOW vectors to classify the sentiment of the reviews. The

results of this experiment provide a baseline for the performance of different machine learning algorithms on the sentiment analysis task using BOW representation.

6.1.2 BOW + TF-IDF

The purpose of this experiment is to evaluate the effectiveness of combining two preprocessing techniques, BOW and TF-IDF, in improving the performance of the machine learning models. The results of this experiment can be used to determine whether the combination of BOW and TF-IDF provides better results [50] than either technique used alone, and to identify the best performing machine learning model for the given dataset.

Second experiment builds upon the previous experiment and evaluates the performance of the models when using a combination of two preprocessing techniques: BOW and TF-IDF. TF-IDF is a weight measure that takes into account both the frequency of a word in a document and the inverse frequency of the word across the entire corpus. This measure assigns a higher weight to words that are frequent in a document but rare in the corpus, as such words are likely to be more important in distinguishing between documents. After applying BOW, the next step is to apply TF-IDF to the resulting matrix. This step involves multiplying the BOW matrix by the IDF matrix, which is calculated by dividing the total number of documents by the number of documents in which a particular word appears and then taking the logarithm of that quotient. The resulting TF-IDF matrix is then used as input to train and evaluate the performance of various machine learning models.

6.1.3 BOW + TF-IDF + Spell Correction

The motivation behind this experiment is examine if reducing the impact of misspelled words [50] in the text data improves the accuracy of the classification model. Misspelled words can negatively affect the performance of the model by introducing

noise in the data and reducing the accuracy of the model's predictions. By correcting the misspelled words, the model can better understand the context of the text data and improve its accuracy.

Third experiment contains the preprocessing techniques used in second experiment (BOW + TF-IDF) plus spell correction. In this experiment, the misspelled words in the dataset are corrected using a spell correction algorithm before applying the BOW and TF-IDF vectorization [51]. To implement spell correction, a Python library called "pyspellchecker" is used. This library provides an interface to perform spell correction using a pre-trained language model. The library checks each word in the text data against a pre-built dictionary and replaces any misspelled words with the most probable correction. After spell correction, the text data is vectorized using BOW and TF-IDF as in experiment 2. The resulting vectorized data is then used to train and evaluate the classification models. The performance of these models is compared with the models trained on the data preprocessed using other techniques, to assess the impact of the spell correction technique on the accuracy of the models.

6.1.4 BOW + TF-IDF + Spell Correction + Stemming

The motivation behind this experiment is to evaluate the impact of stemming, which is a technique used to reduce words to their root form, on the performance of the classification models.

The forth experiment includes the following preprocessing techniques: BOW, TF-IDF, spell correction, and stemming. First, spell correction is performed on the preprocessed data to correct any spelling mistakes present in the data. Then, stemming is applied to the preprocessed data to convert words to their root form. Finally, the text data is preprocessed by applying BOW and TF-IDF techniques to convert the textual data into numerical vectors. The results of this experiment are compared to those of the previous

experiments to determine the impact of stemming on the performance of the classification models.

6.1.5 BOW + TF-IDF + Spell Correction + Lemmatization

The motivation behind this experiment is to evaluate the influence of Lemmatization. This experiment uses lemmatization, which is a more advanced technique [52] than stemming to reduce words to their base form because it produces more accurate results. Stemming can result in words that are not real words or have a different meaning, while lemmatisation produces real words that have a meaningful base form. The aim is to evaluate if lemmatization improves the accuracy of the model by reducing the number of features and addressing the issue of sparsity in the dataset.

This experiment is the last experiment of this thesis, where a combination of pre-processing techniques including BOW, TF-IDF, spell correction, and lemmatization is used. The only change in this experiment is taking lemmatization in place of stemming.

6.2 Implementation

Python (version 3.7.8) and its libraries are used for data collection, dataset construction, as well as model implementation and evaluation. The most popular and commonly used Python libraries for data manipulation and machine learning model implementation were used in this project. The following is a list of the used libraries, their versions, and what they are used for:

- **Scikit-learn (sklearn):** This library provides a wide range of machine learning algorithms and tools for data analysis, including KNN, LR, and SVM.
- **NumPy:** NumPy's most used functionality is the creation of N-dimensional arrays. Moreover, it offers tools for performing mathematical operations on vectors and matrices, as well as functions for generating random numbers.

Furthermore, a range of other Python libraries, such as PyTorch, utilize it to implement the tensor data structure.

- **Pandas:** Pandas is a software library written for the Python programming language for data manipulation and analysis. In general, it offers data structures and operations for manipulating numerical dataframe and time series.
- **TensorFlow:** This library were used for implementing artificial neural networks (ANN), and long short-term memory (LSTM) model and BERT, which are popular deep learning architectures for text classification tasks.
- **Transformers:** This library was used for implementing BERT, which is a state-of-the-art pre-trained language model known for its exceptional performance in various natural language processing (NLP) tasks.
- **NLTK (Natural Language Toolkit) and SpaCy:** These libraries were used for text processing tasks such as spell correction, stemming, and lemmatization, which are common pre-processing techniques in NLP.

6.2.1 Data pre-processing implementation

To implement data pre-processing in Python, we use libraries such as pandas, numpy, and sklearn as listed in Table 6.2.

Table 6.2 Pre-processing Techniques for Text Data in Python

Data Preprocessing techniques	Description
Lower Casing	Converts text to lowercase using the built-in Python method "lower()"
Removal of Punctuations	Removes punctuations using the "translate()" method with an empty string and a list of punctuation characters as arguments
Removal of Stop Words	Uses the Natural Language Toolkit (NLTK) library to remove stop words
Removal of Frequent and Rare Words	Uses the Counter library from the collection package to obtain a list of most common and least common words
Removal of HTML and URLs	Uses the "re.sub()" function with a regular expression pattern to find and replace HTML and URLs with an empty string
Spell Correction	Uses the TextBlob library to correct spelling errors in the text
Stemming and Lemmatization	Uses the Porter Stemmer and Porter Lemma libraries in Python to perform stemming and lemmatization on the text

The Table 6.2 summarizes several text pre-processing techniques used in natural language processing. These techniques include lower casing, removal of punctuations, stop words, frequent and rare words, HTML and URLs, spell correction, and stemming/lemmatization. Python provides built-in functions to perform most of these pre-processing steps, such as the "lower()" method for lower casing, "translate()" method for removing punctuations, and the Natural Language Toolkit (NLTK) library for removing stop words. The Counter library is used for identifying frequent and rare words, while regular expressions are used for removing HTML and URLs. Table 6.3 showcases commonly used metacharacters to extract specific string patterns and remove all

unnecessary characters from the string. Spell correction is performed using the TextBlob library, and stemming and lemmatization are implemented using Porter Stemmer and Porter Lemma, respectively.

Table 6.3 Metacharacters Supported by the re module

Character (s)	Meaning
.	Matches any single character except newline
$	Anchors a match at the end of a string
*	Matches zero or more repetitions
+	Matches zero or more repetitions
?	Matches zero or one repetition and specifies the non-greedy versions of *, +, and ?
^	Anchors a match at the start of a string and complements a characte
{}	Matches an explicitly specified number of repetitions

6.2.2 Model implementation

To implement K-Nearest Neighbours (KNN), Support Vector Machines (SVM), and logistic regression (LR) in Python, we used the scikit-learn (sklearn) library, which provides efficient and easy-to-use implementations of these machine learning algorithms.

- **KNN, LR, and SVM:** For KNN, we created an instance of the **KNeighborsClassifier** class, specify the number of neighbours to consider, and then fit the model to your training data using the **fit()** method. For SVM, we used the **SVC** class for classification or **SVR** class for regression, and again, fit the model to training data using the **fit()** method. For LR, we used the **LogisticRegression** class, and fit the model to training data using the **fit()**

method. Once the models are trained, the **predict()** method is used to make predictions on training data. We evaluated the performance of these models using various metrics such as accuracy, which are available in the **metrics** module of the sklearn library.

- **ANN:** We implemented ANN model with 95 neurons in first input layer followed by 75, 55, 35, and 15 neurons in each hidden layers. Determining the appropriate number of neurons in each layer of an ANN can be a trial-and-error process, and there is no universally accepted method for selecting the exact number of neurons. However, there are some general guidelines that can be used to justify the number of neurons in each layer. One common approach is to start with a bigger number of neurons and gradually decrease until a satisfactory level of performance is achieved. Our dataset has binary output so ouput layer has only one neuron. Finally, the model is compiled using the Adam optimizer, binary cross-entropy loss function, and accuracy metric. The optimizer is used to update the model's parameters during training, the loss function is used to calculate the error between the predicted and actual values, and the accuracy metric is used to evaluate the model's performance.
- **LSTM:** Our LSTM network contains vectors of fixed size 100, bidirectional layer with 256 neurons, dence layer containing 24 neurons, followed by 1 neuron in output layer. The LSTM network was designed with a vector size of 100 to handle input data. A bidirectional layer with 256 neurons was used to enable the network to process the input sequence in both forward and backward directions. The use of the power of two for the number of neurons in bidirectional layers is a common practice in machine learning. It is because many hardware devices, such as GPUs, perform better when the number of computations is in powers of two. As

bidirectional layers require more computations, using a power of two for the number of neurons can make the computations more efficient and faster. The dense layer contained 24 neurons, which is a common approach is to start with a bigger number of input neurons and gradually decrease until a satisfactory level of performance is achieved.. The model has a total of 387,601 trainable parameters, which are updated during training using the Adam optimizer and binary cross-entropy loss function. The model is trained to minimize the binary cross-entropy loss between the predicted and actual labels of the input sequence, and to maximize the classification accuracy on the training data.

- **BERT:** Our BERT model has two input layers for two separate text inputs, each with a shape of (None, 128) indicating the maximum sequence length of 128 tokens. The choice of a maximum sequence length of 128 tokens is a common practice in BERT-based models. This is because BERT is a very large model and has a significant computational cost, especially for longer sequences. The input is passed through a pre-trained BERT model with 109,482,240 parameters, which generates a sequence of hidden states as output. The output is then passed through a dense layer with 1 output units and a softmax activation function, which produces a probability distribution over the two classes. The total number of trainable parameters in this model is 109,484,547, which includes the parameters in the BERT model and the dense layer.

CHAPTER VII:

EXPERIMENTAL RESULTS

To evaluate the performance of our sentiment analysis models, we split our dataset of 20,000 reviews into a training set and a testing set. We used a 75:25 ratio, with 15,000 reviews for training and 5,000 reviews for testing. The training set was used to train our models, while the testing set was used to evaluate their performance on unseen data. In addition to splitting our dataset into training and testing sets, we also employed cross-validation to further validate the performance of our models. We used k-fold cross-validation with k=5, which involved splitting the training set into five equal-sized subsets, training our models on four of the subsets and using the remaining subset as the validation set.

These experiments aim to investigate the impact of different pre-processing techniques on the performance of various machine learning models. Table 7.1 summarises the accuracy of all five experiments for six machine leaning models.

Table 7.1 Experimental results

Experiments	KNN	LR	SVM	ANN	LSTM	BERT
BOW	87.34	81.09	86.6	90.68	92.87	93.61
BOW + TF-IDF	88.92	74.32	87.34	92.43	94.76	95.72
BOW + TF-IDF + Spell Correction	89.56	80.52	89.43	93.97	95.33	96.44
BOW + TF-IDF + Spell Correction + Stemming	91.2	77.32	93.67	95.89	97.90	98.78
BOW + TF-IDF + Spell Correction + Lemmatisation	93.4	78.34	94.22	96.87	97.95	98.99

In the first experiment, the baseline model was built using the BOW representation. We can observe that all models performed reasonably well with an accuracy ranging from 87.34% to 93.61%. LSTM and BERT achieving the highest accuracies, indicating that they are better suited for text classification tasks than other models.

In the second experiment, the BOW approach was combined with TF-IDF. The results show that almost all the models performed better than in Experiment 1, with an accuracy ranging from 88.92% to 95.72%. However, the accuracy of LR model decreased from 81.09% to 74.32%. This is because TF-IDF assigns weights to each word in the document based on its importance, and this can affect the performance of linear models like LR.

In the third experiment, spell correction was applied to the pre-processed text data. The results show a significant improvement in accuracy for all six models, with an accuracy ranging from 89.56% to 96.44%. The improvement in accuracy is due to the correction of spelling errors that could have affected the models' ability to correctly classify the sentiment of the text.

In the fourth experiment, stemming was applied to the pre-processed data from the third experiment. Stemming involves reducing words to their root form, which can help in reducing the number of unique words in the text and improving the models' ability to generalize to unseen data. The results show that all six models achieved higher accuracy than in third experiment, ranging from 91.2% to 98.79%. This shows that stemming has a positive impact on the models' performance for sentiment analysis.

In the last experiment, lemmatization was applied to the pre-processed data from third experiment. Lemmatization is similar to stemming, but it reduces words to their base form (lemma) rather than just their root form. The results show a further

improvement in accuracy for all six models, ranging from 93.4% to 98.99%. This suggests that lemmatization is an effective technique for improving sentiment analysis models' performance. Overall, the result demonstrates that the choice of pre-processing techniques can have a significant impact on the performance of sentiment analysis models. Spell correction, stemming, and lemmatization were found to be effective techniques for improving the models' accuracy. Deep learning models like LSTM and BERT performed best overall, followed by SVM, ANN, KNN, and LR.

Next, we compare this work with other similar efforts targeting to evaluate the efficiency of sentiment analysis mechanisms in terms of dataset used and the review format adopted. Table 7.2 shows a comparison of datasets and review format used in seven prior studies [8, 9, 10, 11, 12, 13], and the proposed approach. The studies used datasets from Amazon reviews and commercial businesses and covered a variety of categories, including electronics, mobile phones, online books, health & personal care, and others. Most of the studies used paragraph-based reviews, while the proposed approach utilized sentence-based reviews. Therefore, we can compare the proposed approach with prior studies in terms of their accuracy and effectiveness in analyzing sentiment in the selected categories using a different review format.

Table 7.2 Comparison of datasets and review format

Papers	*Dataset*	*Categories*	*Review format*
[8]	Amazon Review	Electronics	Paragraph based reviews
[9]	Amazon Review	Mobile Phones	Paragraph based reviews
[10]	Amazon Review	Online Books	Paragraph based reviews
[11]	Commercial Business Dataset	Restaurants, Shopping, Fashion, Education, Entertainment, Hotels, and tourism	Paragraph based reviews
[12]	Amazon Reviews	Health & Personal Care	Paragraph based reviews
[13]	Amazon Review	Books, Camera, Magazines, Electronics	Paragraph based reviews
Proposed Approach	Amazon Review	Books, electronics, health & beauty, and food	**Sentence based reviews**

Compared with other similar efforts, our evaluation is more comprehensive as in terms of both pre-processing techniques and analytical algorithms. . The other main key difference between all these previous efforts and our proposed method is that we use sentence based reviews because In a paragraph-based dataset, the sentiment of the entire paragraph may be influenced by only a few words or sentences, making it difficult for sentiment analysis models to accurately classify the sentiment of the entire paragraph. By using a sentence-based dataset, each individual sentence can be classified on its own, allowing for a more balanced and accurate sentiment analysis. Moreover, paragraph-based sentiment can be based on our literature search [8, 9, 10, 11, 12, 13], we compared our selected pre-processing techniques with other six previous works. Table 7.3. summarizes the comparison of above recent studies on different pre-processing techniques with this study.

Table 7.3 Comparison of different pre-processing methods *on Amazon Review dataset*

	Data Pre-processing techniques				
Papers	*BOW*	*TF-IDF*	*Spell Correction*	*Stemming*	*Lemmatization*
[8]		√			√
[9]	√		√		√
[10]	√		√		
[11]		√	√	√	
[12]	√			√	
[13]	√				
Proposed Approach	√	√	√	√	√

We compared our proposed evaluation techniques and sentiment classification models with six previous efforts by Sinnasamy, & Sjaif (2022) [8], Alharbi, Alghamdi, N. S. (2021) [9], Rain, Callen [10], Omar Al-Haribi [11], Chen, Weikang [12], and Shaikh, Tahura & Deepa [13] in the below Table 7.4. In [8], authors implemented Multinomial Naive Bayes and Support Vector Machine algorithms to achieve an accuracy of 82.27%. In [9], authors used LSTM model for classifications and achieved an accuracy of 93.63%. In [10], authors implemented a simple bias and decision tree algorithm, and achieved an accuracy of 87.33%. In [11], author used commercial business dataset which is written in Arabic language. They assessed the effectiveness of the SVM classifier for dialectal Arabic sentiment analysis using five feature selection techniques and achieved 93.25% accuracy. In [12], authors evaluated the performance using Root Mean Square Error (RMSE) score. Because RMSE and Accuracy are used to evaluate the performance of models in different contexts, it is not appropriate to directly

compare them. A study [13] used a simple biased algorithm and achieved an accuracy of 80.00%.

Table 7.4 Comparison of different classification algorithms

Papers	*Models*	*Highest Accuracy*
[8]	MNB, SVM	82.27%
[9]	LSTM	93.63%
[10]	Naïve Bias, Decision Tree	87.33%
[11]	SVM	93.25%
[13]	Naïve Bias	80.00%
Proposed Approach	KNN, LR, SVM, ANN, LSTM, BERT	**98.99%**

Based on our analysis, we have found that a combination of appropriate pre-processing techniques and machine learning algorithms can result in excellent sentiment analysis accuracy. Our approach includes techniques such as stop word removal, stemming, lemmatization, and spell correction, along with machine learning algorithms like SVM, KNN, LR, ANN, LSTM and BERT. Our results have shown that this combination outperforms many previous studies in the field, achieving high levels of accuracy on various datasets, including those containing paragraph-based and sentence-based reviews. We believe that this combination can be used as a reliable and effective approach for sentiment analysis tasks in various domains.

CHAPTER VII:

CONCLUSION & FUTURE WORK

8.1 Conclusion

In this study, we proposed an efficient approach that evaluates the effectiveness of different mechanisms for sentiment analysis of Amazon reviews. We conducted five experiments using a combination of different data preprocessing techniques and analysis models. Our proposed method covers evaluation of the preprocessing techniques and various sentiment analysis models. We first prepare a high-quality Amazon review data by manually labeling. Then, we perform different types of data cleaning techniques and train various sentiment analysis models. The results of carefully-designed experiments show that the deep learning based algorithms such as ANN, LSTM, and BERT outperformed other types of machine learning algorithms. By adding TF-IDF, spell correction, stemming, and lemmatization, we observed a significant increment in accuracy from 74.32% to 98.99%. However, LR performed relatively lower in terms of accuracy due to its limitations as a linear model. This suggests that effective data pre-processing and appropriate model selection are very important in sentiment analysis.

8.2 Future Work

Although the findings of this thesis provide valuable insights into the effectiveness of different mechanisms for sentiment analysis of Amazon reviews, we plan to extend this research in the following directions.

First , we would like to investigate the performance of the models on a larger dataset, including reviews from other e-commerce platforms or different domains. This would enable the generalization of the findings and provide a more comprehensive understanding of the effectiveness of different mechanisms in sentiment analysis.

Second, we will explore the impact of other preprocessing techniques such as part-of-speech tagging or named entity recognition on the performance of sentiment analysis models. We may lose some meaningful information while cleaning the data. We may need to consider techniques like sentiment-specific stop words or sentiment-aware lemmatization to avoid removing essential sentiment-related information in the future. Moreover, investigating the effect of different hyperparameters on the performance of models could also be beneficial in identifying optimal configurations.

Furthermore, Due to the time limitation, this research only focused on the accuracy of the models in predicting sentiment labels. However, future work could explore other evaluation metrics such as precision, recall, and F1-score to provide a more comprehensive evaluation of the models' performance.

Finally, the findings of this study can be applied in practical applications such as customer service or product development. Future research can be conducted to investigate how these mechanisms can be integrated into real-world applications to improve customer satisfaction and business strategies.

REFERENCES

[1] Junichi Tsujii; Natural Language Processing and Computational Linguistics. *Computational Linguistics* 2021; 47 (4): 707–727-doi: https://doi.org/10.1162/, coli_a_00420

[2] D'Arbelles, Kristine et al. "Electronic word-of-mouth marketing on Amazon: Exploring how and to what extent Amazon reviews affect sales." (2020).

[3] Jiang, Baojun and Bin Wang. "Impact of Consumer Reviews and Ratings on Sales, Prices, and Profits: Theory and Evidence." *International Conference on Interaction Sciences* (2008).

[4] Peter Turney." Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews", Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, 417-424.

[5] Vandana Jagtap & Karishma Pawar, "Analysis of different approaches to Sentence-Level Sentiment Classification", International Journal of Scientific Engineering and Technology, 2013, 164-170.

[6] Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, and Hui Wang, "Aspect-level Sentiment Classification with HEAT (HiErarchical ATtention) Network", In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17). Association for Computing Machinery, New York, NY, USA, 97–106.

[7] Bo Pang and Lillian Lee. "Seeing stars Exploiting class relationships for sentiment categorization with respect to rating scales ". Proceedings of the ACL, 2005.

[8] Sinnasamy, & Sjaif, N. N. A. (2022). Sentiment Analysis using Term based Method for Customers' Reviews in Amazon Product. *International Journal of Advanced*

Computer Science & Applications, *13*(7). https://doi.org/10.14569/IJACSA.2022.0130780

[9] Alharbi, Alghamdi, N. S., Alkhammash, E. H., & Al Amri, J. F. (2021). Evaluation of Sentiment Analysis via Word Embedding and RNN Variants for Amazon Online Reviews. *Mathematical Problems in Engineering*, *2021*, 1–10. https://doi.org/10.1155/2021/5536560

[10] Rain, Callen. "Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning."Swarthmore College (2013).

[11] Omar Al-Haribi. "A Comparative Study of Feature Selection Methods for Dialectal Arabic Sentiment Classification Using Support Vector Machine”**,** IJCSNS International Journal of Computer Science and Network Security, VOL.19 No.1, January 2019, 167-176.

[12] Chen, Weikang, Chihhung Lin, and Yi-Shu Tai."Text-Based Rating Predictions on Amazon Health & Personal Care Product Review." (2015)

[13] Shaikh, Tahura, and DeepaDeshpande. "Feature Selection Methods in Sentiment Analysis and Sentiment Classification of Amazon Product Reviews.",(2016)

[14] Nasr, Mona Mohamed, Essam Mohamed Shaaban, and Ahmed Mostafa Hafez. "Building Sentiment analysis Model using Graphlab." IJSER, 2017

[15] Text mining for yelp dataset challenge; Mingshan Wang; University of California San Diego, (2017)

[16] Blitzer J, Dredze M, Pereira F (2007) Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In: Proceedings of the 45th annual meeting of the Association of Computational Linguistics. ACL, pp 440–447

[17] Dang Y, Zhang Y, Chen H. A lexicon-enhanced method for sentiment classification: an experiment on online product reviews. *IEEE Intell Syst.* 2009;25(4):46–53. doi: 10.1109/MIS.2009.105.

[18] Kharde V, Sonawane P, et al. Sentiment analysis of twitter data: a survey of techniques. *Int J Comput Appl.* 2016; 975:0975–8887.

[19] Fahrni A, Klenner M (2008) Old wine or warm beer: target-specific sentiment analysis of adjectives. University of Zurich, pp 60–63

[20] S. K. Dwivedi and B. Rawat, "A review paper on data preprocessing: A critical phase in web usage mining process," 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), Greater Noida, India, 2015, pp. 506-510, doi: 10.1109/ ICGCIoT.2015.7380517.

[21] S. Agarwal, "Data Mining: Data Mining Concepts and Techniques," 2013 Interna- tional Conference on Machine Intelligence and Research Advancement, Katra, India, 2013, pp. 203-207, doi: 10.1109/ICMIRA.2013.45.

[22] HaCohen-Kerner Y, Miller D, Yigal Y. The influence of preprocessing on text classification using a bag-of-words representation. PLoS One. 2020 May 1;15(5):e0232525. doi: 10.1371/journal.pone.0232525. PMID: 32357164; PMCID: PMC7194364.

[23] Abilhoa WD, De Castro LN. A keyword extraction method from twitter messages represented as graphs. *Appl Math Comput.* 2014;240:308–325.

[24] Cambria E, Hussain A. *Sentic computing. Cogn Comput.* 2015;7(2):183–185. doi: 10.1007/s12559-015-9325-0.

[25] Patil P, April Yalagi P. Sentiment analysis levels and techniques: a survey. *Int J Innov Eng Technol.* 2016;6:523.

[26] Serrano-Guerrero J, Olivas JA, Romero FP, Herrera-Viedma E. Sentiment analysis: a review and comparative analysis of web services. *Inf Sci.* 2015;311:18–38. doi: 10.1016/j.ins.2015.03.040.

[27] N. U. Pannala, C. P. Nawarathna, J. T. K. Jayakody, L. Rupasinghe and K. Krishnadeva, "Supervised Learning Based Approach to Aspect Based Sentiment Analysis," *2016 IEEE International Conference on Computer and Information Technology (CIT)*, Nadi, Fiji, 2016, pp. 662-666, doi: 10.1109/CIT.2016.107.

[28] T.M.Saravanan and Angamuthu Tamilarasi. “A Hybrid Kernel Based Extreme Learning Machine for Effective Sentiment Analysis.” .

[29] Kim H, Howland P, Park H, Christianini N. Dimension reduction in text classification with support vector machines. *J Mach Learn Res.* 2005;6(1):37–53

[30] Li X, Li J, Wu Y. A global optimization approach to multi-polarity sentiment analysis. *PLoS ONE.* 2015;10(4):0124672. doi: 10.1371/journal.pone.0124672

[31] Mullen T, Collier N (2004) Sentiment analysis using support vector machines with diverse information sources. In: Proceedings of the 2004 conference on empirical methods in natural language processing. ACL, pp 412–418

[32] Reddy, P & Sri, D & Reddy, C & Shaik, Subhani. (2021). Sentimental Analysis using Logistic Regression. International Journal of Engineering Research and Applications. 11. 36-40. 10.9790/9622-1107023640.

[33] Hota, Soudamini & Pathak, Sudhir. (2018). KNN classifier based approach for multi-class sentiment analysis of twitter data. International Journal of Engineering and Technology(UAE). 7. 1372-1375. 10.14419/ijet.v7i3.12656.

[34] Ashraf, Mohammed & Bhatt, Naw. (2021). A LAZY APPROACH FOR TWITTER SENTIMENT ANALYSIS. 10.13140/RG.2.2.36413.18406.

[35] Srinivas, Akana & Satyanarayana, Ch & Divakar, Ch & Sirisha, Katikireddy. (2021). Sentiment Analysis using Neural Network and LSTM. IOP Conference Series: Materials Science and Engineering. 1074. 012007. 10.1088/1757-899X/1074/1/012007.

[36] Manogaran Madhiarasan, Mohamed Louzazni, "Analysis of Artificial Neural Network: Architecture, Types, and Forecasting Applications", *Journal of Electrical and Computer Engineering*, vol. 2022, Article ID 5416722, 23 pages, 2022. https://doi.org/10.1155/2022/5416722

[37] Murthy, Dr & Allu, Shanmukha & Andhavarapu, Bhargavi & Bagadi, Mounika. (2020). Text based Sentiment Analysis using LSTM. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS050290.

[38] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, Nov. 1997, doi: 10.1109/78.650093.

[39] Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171-4186.

[40] Graves, A., Fernández, S., Schmidhuber, J. (2005). Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds) Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005. ICANN 2005. Lecture Notes in Computer Science, vol 3697. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11550907_126

[41] Schmidt, S.; Zorenböhmer, C.; Arifi, D.; Resch, B. Polarity-Based Sentiment Analysis of Georeferenced Tweets Related to the 2022 Twitter Acquisition. *Information* **2023**, *14*, 71. https://doi.org/10.3390/info14020071

[42] Wankhade, M., Rao, A.C.S. & Kulkarni, C. A survey on sentiment analysis methods, applications, and challenges. *Artif Intell Rev* **55**, 5731–5780 (2022). https://doi.org/10.1007/s10462-022-10144-1

[43] Hota HS, Sharma DK, Verma N. Lexicon-based sentiment analysis using Twitter data: a case of COVID-19 outbreak in India and abroad. Data Science for COVID-19. 2021:275–95. doi: 10.1016/B978-0-12-824536-1.00015-0. Epub 2021 May 21. PMCID: PMC8989068.

[44] Ghorbani, M., Bahaghighat, M., Xin, Q. *et al.* ConvLSTMConv network: a deep learning approach for sentiment analysis in cloud computing. *J Cloud Comp* **9**, 16 (2020). https://doi.org/10.1186/s13677-020-00162-1

[45] A. Krouska, C. Troussas and M. Virvou, "The effect of preprocessing techniques on Twitter sentiment analysis," *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*, Chalkidiki, Greece, 2016, pp. 1-5, doi: 10.1109/IISA.2016.7785373.

[46] Jochen Hartmann, Mark Heitmann, Christian Siebert, Christina Schamp, More than a Feeling: Accuracy and Application of Sentiment Analysis, International Journal of Research in Marketing, Volume 40, Issue 1,2023, Pages 75-87, ISSN 0167-8116

[47] Rianto, Mutiara, A.B., Wibowo, E.P. *et al.* Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation. *J Big Data* **8**, 26 (2021). https://doi.org/10.1186/s40537-021-00413-1

[48] Åsmund Rinnan, Frans van den Berg, Søren Balling Engelsen, Review of the most common pre-processing techniques for near-infrared spectra, TrAC Trends in Analytical Chemistry, Volume 28, Issue 10, 2009, Pages 1201-1222, ISSN 0165-9936

[49] Mohey El-Din, Doaa. (2016). Enhancement Bag-of-Words Model for Solving the Challenges of Sentiment Analysis. International Journal of Advanced Computer Science and Applications. 7. 10.14569/IJACSA.2016.070134.

[50] Kundi, Fazal & Khan, Aurangzeb & Asghar, Dr. Muhammad & Ahamd, Shakeel. (2014). Context-Aware Spelling Corrector for Sentiment Analysis. MAGNT Research Report. 2. 1-10.

[51] S. Singh, K. Kumar and B. Kumar, "Sentiment Analysis of Twitter Data Using TF-IDF and Machine Learning Techniques," *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, Faridabad, India, 2022, pp. 252-255, doi: 10.1109/COM-IT-CON54601.2022.9850477.

[52] Balakrishnan, Vimala & Ethel, Lloyd-Yemoh. (2014). Stemming and Lemmatization: A Comparison of Retrieval Performances. Lecture Notes on Software Engineering. 2. 262-267. 10.7763/LNSE.2014.V2.134.

[53] Hu, Xiao & Downie, J. & West, Kris & Ehmann, Andreas. (2005). Mining Music Reviews: Promising Preliminary Results.. 536-539.

[54] Jure Leskovec and Andrej Krevl, SNAP Datasets: Stanford, Large Network Dataset Collection, http://snap.stanford.edu/data, jun, 2014.

[55] Bre, Facundo & Gimenez, Juan & Fachinotti, Víctor. (2017). Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks. Energy and Buildings. 158. 10.1016/j.enbuild.2017.11.045.

APPENDIX A:

ACRONYMS

ML – Machine Learning

DL – Deep Learning

NLP – Natural Language Processing

SA – Sentiment Analysis

BOW – Bag of Words

TF-IDF – Term Frequency-Inverse Document Frequency

LSTM – Long Short Term Mermory

BERT – Bidirectional Encoder Representations from Transformers

LR – Logistic Regression

SVM – Support Vector Machine

KNN – K-Nearest Neighbours

ANN – Artificial Neural Network