Enhanced Sentiment Analysis in Microblogs through the use of XGboost Classifier and Genetic Algorithm

Roza Hikmat Hama Aziz

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Applied Mathematics and Computer Science

Eastern Mediterranean University
September 2021
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy in Applied Mathematics and Computer Science.

Prof. Dr. Nazim Mahmudov
Chair, Department of Mathematics

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Doctor of Philosophy in Applied Mathematics and Computer Science.

Assoc. Prof. Dr. Nazife Dimililer
Supervisor

Examining Committee

1. Prof. Dr. Hamza Erol

2. Prof. Dr. Mehmet Reşit Tolun

3. Assoc. Prof. Dr. Nazife Dimililer

4. Asst. Prof. Dr. Ersin Kuset Bodur

5. Asst. Prof. Dr. Müge Saadetoğlu

ABSTRACT

Studies on sentiment analysis and opinion mining initially focused on polarity classification through the use of positive, negative, or neutral categories. Nevertheless, despite their importance in a wide range of applications, the classification of extreme opinions, such as highly negative and very positive ones were not targeted until recently. In this work, we focus on a 5-point scale to include extreme sentiments as well. The majority of studies in this domain have focused on approaches tailored towards special datasets. This doctoral thesis proposes two novel ensemble classifier approaches to improve the performance of the sentiment analysis task. The first proposed ensemble classifier framework called "SentiXGboost" is designed to improve binary sentiment analysis tasks using the XGBoost algorithm as a meta-classifier for stacked ensembling. The second proposed approach provides a framework based on the concept of the Genetic Algorithms for producing an optimized classifier ensemble for binary, ternary, and fine-grained, denoted "SentiGA", sentiment analysis task. Both of the proposed approaches are evaluated on the major sentiment datasets, including SemEval-2017 (Sentiment Analysis in Twitter) task (4A, 4B, and 4C), Stanford Sentiment Treebank (SST-2 and SST-5), Sentimet140, Sentiment Labelled Sentences (Amazon), Stanford Sentiment Gold Standard, Yelp Challenge Dataset and Movie Review (Sentiment Polarity Dataset V2.0). The performance of both proposed approaches is compared with other existing well-known methods in the field using the same datasets. The results show that our proposed approaches have successfully enhanced the performance of sentiment analysis classification compared to other existing methods.

Keywords: sentiment analysis, feature extraction methods, machine learning approaches, ensemble learning approaches, simple majority voting, weighted majority voting, optimized ensemble classifier, XGBoost, and genetic algorithm.

ÖZ

Duygu analizi ve fikir madenciliği alanındaki ilk çalışmalar olumlu, olumsuz veya tarafsız kategorilerinden yararlanarak, özellikle görüşlerin polarite veya kutupluluklarına göre iki veya üç kategoriye göre sınıflandırılması gibi konulara yoğunlaştı. Bununla birlikte, duyguların ve görüşlerin derecelendirilmesi ve çok olumsuz ve çok olumlu görüşler gibi aşırı görüşlerin de tanımlanması birçok uygulamada büyük önem taşıdığı halde bu konularında çalışmalara fazla yer verilmemiştir. Bu çalışmada, kutupluluk sınıflandırması yanında, aşırı duyguları da içerecek şekilde 5 puanlık bir ölçeğe odaklanıyoruz. Bu alandaki çalışmaların çoğu, özel veri setlerine için uyarlanmış yaklaşımlara odaklanmıştır. Bu doktora tezi, duygu analizinde performansı iyileştirmek için iki yeni sınıflandırıcı topluluğu yaklaşımı önermektedir. "SentiXGboost" olarak adlandırılan ilk önerilen sınıflandırıcı topluluğu, yığın kümeleme için bir meta sınıflandırıcı olarak XGBoost algoritmasını kullanarak iki sınıflı duygu analizi sistemi geliştirmek için tasarlanmıştır. Önerilen ikinci sınıflandırıcı toplululuğu, SentiGA, ikili, üçlü ve ince taneli sınıflandırmalar için optimize edilmiş bir sınıflandırıcı topluluğu üretmek için Genetik Algoritma kavramına dayalı bir çerçeve sunar. Önerilen yaklaşımların her ikisi de SemEval-2017 (Sentiment Analysis in Twitter /Twitter'da Duygu Analizi) görevi (4A, 4B ve 4C), Stanford Sentiment Treebank (SST-2 ve SST-5), Sentimet140, Sentiment Labeled Sentences/Duygu Etiketli Cümleler (Amazon), Stanford Sentiment Gold Standard, Yelp Challenge ve Movie Review/Film İncelemesi (Sentiment Polarity Dataset V2.0) veri seti dahil olmak üzere önemli duygu veri setlerinde değerlendirilmiştir. Önerilen her iki yaklaşımın performansı, aynı veri setleri kullanılarak sahada mevcut diğer iyi bilinen yöntemlerle

karşılaştırılmıştır. Sonuçlar, önerilen her iki yaklaşımın da diğer mevcut yöntemlere kıyasla duygu/görüş analizinin performans sınıflandırmasını başarıyla geliştirdiğini göstermektedir.

Anahtar Kelimeler: duygu analizi, öznitelik çıkarımı yöntemleri, makine öğrenimi yaklaşımları, topluluk öğrenimi yaklaşımları, basit çoğunluk oylaması, ağırlıklı çoğunluk oylaması, optimize edilmiş topluluk sınıflandırıcı, XGBoost ve genetik algoritma

DEDICATION

To My Family

ACKNOWLEDGMENT

In the name of Allah, the Compassionate, the Merciful

I want to first thank the Almighty Allah for His mercies and kindness which sustained me right from the beginning of this program.

My sincere gratitude goes to my supervisor, Assoc. Prof. Dr. Nazife Dimililer, who has been of tremendous support and assistance in this thesis. On many occasions, she provided constructive criticisms and corrections to enhance the quality of the thesis. I am equally grateful to the Chair and staff of the Department of Applied Mathematics and Computer Science for the knowledge imparted to me during my Ph.D. program.

I must acknowledge my parents and the entire family members for their support, encouragement, and show of love throughout my program. Particularly, I thank my husband Dr. Ako Muhammad for standing by me all the time.

I am equally grateful to the management of the University of Sulaimani for the privilege given to me to embark on this Ph.D. program in North Cyprus. To all my friends and coursemates who supported me in different ways, I say thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

D_t	Distribution at Iteration t
H(x)	Combination of Outputs $\{h_1(x), \ldots, h_t(x)\}$
h_t	Weak Classifier from Trained on D_t
$\mathfrak{I}$	Base Classifier
Ω	Regularization Term
ϵ_t	Error of h_t
Acc	Accuracy
BoW	Bag of Words
CRF	Conditional Random Filed
DT	Decision Tree
FLR	Fuzzy Lattice Reasoning
KNN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
LR	Logistic Regression
NB	Naïve Bayes
NLG	Natural Language Generation
NLP	Natural Language Processing
ME	Maximum Entropy
PoS	Part-of-Speech
Pre	Precision
RBF	Radial Basis Function
Rec	Recall
RF	Random Forest

RFT	Random Forest Tree
SGD	Stochastic Gradient Descent
SMO	Sequential Minimal Optimization
SVM	Support Vector Machine
SVR	Support Vector Regression
T'F-IDF	Term Frequency-Inverse Document Frequency

Chapter 1

INTRODUCTION

Recently, the production of textual documents has increased exponentially in social media and online web stores. For example, as of 2020, Twitter had 186 million followers sending about 500 million tweets daily[1]. Generally, social media can be used for different purposes, such as tour consulting services, election forecasts, and financial trends. Other uses include advertisements, spam spreading and reporting, receiving customer feedback on products, platforms, and websites of e-commerce such as Amazon [1]. Social media provides interactive connections between users, where people express opinions, share views and establish relations by sharing messages and comments. Social media allow users to quickly and conveniently express their thoughts, feelings, and viewpoints with others, resulting in an enormous amount of messages, reviews, and feedback on various topics [2]. This massive amount of data can be valuable for consumers as well as advertisers in analyzing public opinion on various topics. However, reading and analyzing every review can be an overwhelming task due to the large volumes of text produced. Therefore, there is a need for systems to summarize them. One such method is sentiment analysis.

Sentiment analysis, also known as opinion mining, identifies and predicts people's emotions, feelings, thoughts, and attitudes from accumulated opinions towards a subject [4]. For instance, the users of Rotten Tomatoes can quickly determine

[1] https://www.businessofapps.com/data/twitter-statistics/, (accessed on 02 April 2021)

whether they watch a movie depending upon the sentiment of other users' comments about that movie. According to the sentiments of citizens available in online comments, the government can adapt its policies [3]. Notably, the emergence and growth of sentiment analysis research coincide with the prominence and exponential progress of social media. On the one hand, a massive amount of unstructured data is available on the websites, and on the other hand, the need for extracting and analyzing information automatically from them renders sentiment analysis an important research area in text mining. Sentiment analysis is a challenging issue in the field of natural language processing which is not a recent development.

At the beginning of 2000, sentiment analysis appeared as a research area [5]. Primarily, sentiment analysis was used for categorizing documents into topics such as politics, sport, and business [6]. Researchers had shown great interest in sentiment analysis; however, it was soon understood that it was very different from the classification of standard documents and required external information in the form of sentiment polarity lexicon. The seminal work on sentiment analysis was carried out by Pang et al. [6] and Turney [7]. They introduced the methods utilized in the implementations of sentiment analysis to determine the sentiment orientation of phrases or words as positive or negative. Following that, studies were conducted on the linguistic aspects of expressing opinions and views or sentiments in addition to deeper linguistic processing such as negation handling, finer-grained sentiment distinctions, positional information, and the role of context in determining sentiment orientation. [28-31]. Furthermore, Stoyanov and Cardie note that for fine-grained opinion or sentiment analysis, it is essential not only to determine the polarity of sentiment but also the topic category of the sentiment [8]. Later, with the rapid

growth of social media, sentiment analysis in Twitter became an important research topic. Unfortunately, the lack of suitable lexicons and databases for training, development, and testing systems hindered research in that direction. Over time, some Twitter resources were developed, but they were limited and proprietary. For instance, I-sieve corpus [9] was produced just for Spanish and TASS corpus [10] or depended on noisy labels automatically obtained based on hashtags and emotions such as Hashtag Emotion Corpus [11, 6]. In recent years, this situation changed with the shared task on sentiment analysis on Twitter. The Semantic Evaluation (SemEval) is one of the most important sources of contribution, historically known as the SensEval, which provides public datasets and holds competition on sentiment analysis tasks. Since 2013, this task has been run yearly [14]. The main competition on this task began with SemEval-2013 task 2 [12] and SemEval-2014 task 9 [13] with 2-point scales. Then, in the SemEval-2015 task, sentiment towards a topic was introduced [15], while the SemEval-2016 task added 5-point scales classification and quantification [16, 17].

Most researchers have attempted to build intelligent automated approaches for improving the performance efficiency and accuracy of analyzing sentiment in tweets utilizing different techniques and architecture. A large number of studies have been conducted to assess the sentiment of tweets and classify them using machine learning techniques[18,23,26,27]. In a broader sense, the approaches used in sentiment analysis are generally categorized into two distinct groups, namely the supervised and the unsupervised machine learning approaches [18]. In the supervised learning approach, classification models learn from a labeled set of product reviews to construct a model, which then makes predictions on new datasets. Unsupervised

learning approaches can either work based on lexicon or machine learning. In these approaches, the classifiers do not always require the labeled data to discriminate the given input text. In unsupervised approaches, only the input text is provided to the classifier; hence, the classifiers do not require labeling. The majority of the wide range of methods proposed for sentiment analysis have been supervised machine learning approaches [21-23]. Furthermore, many attempts have been made to enhance the predictive performance of supervised machine learning classifiers in analyzing the sentiment of tweets in different ways. One such way is using ensemble learning techniques which are a significant subfield of machine learning. Ensemble learning techniques aim to develop classification models with better performance by combining the prediction of different base classifiers into a strong classifier. In producing effective ensemble classifiers, it is crucial to identify base learning classifiers that can perform the classification task and ideally involve classifiers with a variety of structures and outputs. Besides, an appropriate combination schema for base learning classifiers is also critical for the performance of ensemble learning approaches [19].

Additionally, combining the well-performing classifiers can be modeled as an optimization problem; hence, the well-established means of meta-heuristic algorithms can provide optimal solutions. Meta-heuristic approaches are widely categorized into two groups: one based on a single solution and the other based on population [20]. Meta heuristics based on a single solution include guided local search, variable neighborhood search, tabu search, and simulated annealing. Meta-heuristics based on population encompass particle swarm optimization, genetic algorithm, differential evolution, and ant colony optimization algorithms. Among the

many approaches used for sentiment analysis, machine learning-based approaches and meta-heuristic algorithms have been successfully implemented in optimizing ensemble classifier approaches [24,25]. In recent years, different ensembling approaches have been proposed and applied for sentiment analysis and critically evaluated. Detailed information regarding these approaches to sentiment analysis is discussed in chapter 2.

1.1 Problem Statement

As noted, sentiment analysis has been employed in many different domains such as retail business, elections, politics, movies, and microblogs to comprehend, track and control human sentiments or reactions toward products, events, or ideas. However, sentiment analysis is associated with some challenges, such as the use of negation and sarcasm, the invention of new words, the existence of spelling mistakes, and different writing styles. The challenges are obstacles in the correct classification of sentiment and reduce the classification performance. The first milestone of sentiment analysis is determining the polarity of sentiment in tweets and categorizing them as positive and negative. To end this, there is a need for efficient, systematic studies on how to extract comments and sentiments from the enormous amount of unstructured text data. Earlier studies have considered that it is possible to categorize text into positive, neutral, or negative sentiments. For example, on a 5-point scale, rating values of one and two are categorized as negative opinions, rating values of four and five are categorized as positive opinions, and rating value of three is categorized as neutral opinions. Fine-grained sentiment analysis tasks play a crucial role in the polarity classification process. Literature offered many classification methods that have been constructed in different ways for specific datasets. However, none of these methods has been designed for datasets that denote extreme sentiments on data

across all disciplines. In contrast, our thesis relies on binary, ternary, and fine-grained sentiment classification tasks focused on identifying opinions. Notably, choosing the best classification method to correctly classify the intensity of sentiment polarity in the text is crucial. It is necessary to design a novel, robust classifier scheme for handling these challenges and issues. Therefore, improving the performance of existing classification methods is still one of the most significant research directions considered by data scientists. The majority of previous sentiment analysis studies have focused on applying supervised machine learning methods and feature extraction techniques to design a classifier. The first objective in this thesis is to build an ensemble classifier method that can conduct classification on binary, ternary, and Fine-grain datasets. Secondly, we also design an ensemble classifier, a meta-classifier, for stacked ensembling and then design an optimized ensemble classifier that selects the well-performing single classifiers from the rest of the classifiers in the pool. The major aspect of our study is based on the construction of both ensemble classifier methods underlying the concept of both ensemble classifier approaches and optimization algorithm techniques.

1.2 Motivation and Research Objectives

Sentiment analysis has been considered an interesting research topic in the past few years. It provides organizations and businesses with solutions for monitoring and analyzing the public's opinion towards their products, brands, and services. Consequently, several studies have focused on sentiment detection of conventional text such as review data, online blogs, and discussion forums. It turns out that the coming on board of social networks and microblogging services has dramatically shifted the attention of research interests towards the analysis of sentiment of microblogging data, specifically tweets data. Twitter is one of the most popular

microblogging services which produces content that reflect opinions about topics, products, and life events [76, 110,47]. This chapter will discuss the main components of sentiment analysis - or Opinion Analysis - as a discipline.

The invention and popularity of social media services have significantly increased the amount of user content in such social environments. Social networks and microblogging services allow more and more people to write and share their opinions, sentiments and seek support on various topics. Currently, one of the most interesting topics is analyzing the sentiments of people. This has become an attractive topic of research where new ideas keep emerging. Nowadays, sentiment analysis is an important task of review mining because it helps producers to improve the quality of products, and consumers can make more accurate and faster-purchasing decisions. Thus, sentiment analysis has seen a considerable effort from business as well as academia. The main aim and motivation of this thesis are to explore key ways of developing an effective ensemble classification scheme that can improve state of the art in sentiment analysis. The focus is on improving the predictive performance of sentiment analysis, specifically on Twitter. To achieve this, two distinctive novel ensemble classification approaches are developed for sentiment analysis. The proposed sentiment classification schemes are carried out under two approaches as follows:

- In the first approach, we propose a novel ensemble classifier framework called “SentiXGboost.” This approach analyzes tweets based on binary sentiment polarity using the concept of the XGBoost ensemble approach. SentiXGboost uses a combination of multiple feature sets with ensemble classification by combining multiple base classifiers,

which are weak learners, into an ensemble classifier using the XGBoost algorithm as a meta-classifier for stacked ensembling. These feature sets include Bag of Words, Term Frequency–Inverse Document Frequency, Part of Speech, *n*-gram, Opinion Lexicon, and Term Frequency. For comparison, the following popular algorithms were implemented for Twitter sentiment analysis: Majority voting, AdaBoost, Gradient Tree Boosting, and Bagging.

- In the second approach, we propose a novel classifier ensemble optimization framework to improve binary, ternary, and fine-grained sentiment analysis tasks. The framework is based on the concept of Genetic Algorithms for optimizing the best-fitting classifier ensembles or sentiment analysis in Twitter. In this framework, we form a large pool of classifiers by training each classifier with different parameter settings and different combinations of feature sets. The predictions of the classifiers in an ensemble are combined using a weighted majority voting rule.

1.3 Thesis Contribution

This thesis proposes two novel approaches for sentiment analysis of tweets from Twitter using ensemble classifier techniques and optimization algorithms. To this end, two ensemble classification approaches are deployed. Both of the proposed ensemble classifier approaches are applied in the field of opinion mining and machine learning. The major contributions introduced by the proposed approaches in this thesis are summarized and listed as follows:

1.3.1 Proposed SentiXGboost Method Contributions

This proposed method contributes to the task of sentiment analysis as outlined below:

- Six different machine learning classifiers, Naive Bayes (NB), Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbors (KNN), Decision Tree (DT), and Stochastic Gradient Descent (SGD), are used as base classifiers.
- Using XGBoost as a meta-classifier for stacked ensembling and finally improving state of the art in terms of performance.
- The proposed approach is the only published work on a stacked ensemble system with XGBoost as a meta-classifier to the best of our knowledge.
- The proposed framework is tested on the sentiment datasets, including SemEval-2017 (Sentiment Analysis in Twitter), Sentiment Labelled Sentences (Amazon), Stanford Sentiment Gold Standard, Stanford Sentiment Treebank (SST-2), Yelp Challenge Dataset, and Movie Review (Sentiment Polarity Dataset V2.0).

1.3.2 Proposed Optimized Ensemble Classifier Method Contributions

The major contributions introduced by this method are summarized as follows:

- This work aims to explore an effective way to conduct fine-grained sentiment analysis by improving the performance and accuracy of sentiment classification and extracting aspects related to the sentiments.
- We propose a novel efficient ensemble classifier framework based on optimization techniques to enhance the predictive performance of sentiment analysis. We first experiment with different machine learning

algorithms, including LR, NB, RF, Support Vector Machines (SVM), DT, and SGD trained with different parameter settings and feature sets to generate a pool of classifiers. The classifier pool has 25 members.

- Selecting the best performing classifiers is crucial to developing an efficient ensemble classifier scheme. We experimented with using the weighted majority voting rule to combine the predictions of the base classifiers and forward search to identify the optimum subset of classifiers from a large classifier pool. The proposed method provides an optimal subset of well-known classifiers for sentiment analysis using the concept of Genetic Algorithms.
- The proposed architecture is tested on the major sentiment datasets, including SemEval 2017 Task (4A, 4B, and 4C) datasets and Stanford Sentiment Treebank (SST-2 and SST-5) datasets. The results of the framework are compared with existing works in the field using the same datasets.
- The results have exceeded the best-published results in all cases to the best of our knowledge.

1.4 Thesis Outline

The rest of this thesis is structured in separate chapters as follows:

Chapter 2 gives the background knowledge on the sentiment analysis task, subtasks, levels, difficulties in sentiment analysis, and popular approaches to implement sentiment analysis—following which an overview of the existing research on sentiment analysis on Twitter is provided. Next, we present the general fundamentals of ensembling approaches. We introduce the categories of ensemble methods used for sentiment analysis then describe the main types of ensemble methods for

classification. The aim is to provide the readers some theoretical and explanations to better understand the ensemble approaches. Lastly, we provide a basic discussion of the Genetic Algorithm optimization technique implemented in this thesis. Chapter 3 presents the experimental settings employed in this thesis to generate both proposed ensemble methods. We also present the detail of benchmark sentiment datasets and performance measures used to evaluate the performances of both proposed sentiment analysis methods presented in Chapter 4 and Chapter 5. Chapter 5 presents and describes the system architecture of the proposed sentiment analysis framework called "SentiXGboost." Additionally, this chapter discusses different parts of the proposed architecture and prototype classifiers + extracted features used. This chapter also explains the implementation of experimental setups based on the experiment settings designed in Chapter 3. The detailed evaluation of the results obtained from the implementation of the experiment is also discussed in this chapter to reflect the objectives of this project. Furthermore, the comparison results of the proposed method against the state-of-the-art methods on the same datasets are presented in this chapter. Chapter 6 describes the proposed framework for producing an optimized classifier ensemble for sentiment analysis using the Genetic Algorithm called "SentiGA." The design of the experiments and the detailed results of the proposed method are also presented in this chapter. Additionally, to further evaluate how effective the proposed framework is, comparisons are made between the results of the proposed method and those of some existing methods using similar sentiment datasets. Chapter 6 concludes the works proposed in this thesis, highlights the main contributions, the limitations of the thesis, and further research directions following this study.

Chapter 2

BACKGROUND ON SENTIMENT ANALYSIS

This chapter presents a comprehensive literature review that covers the theoretical background and technical foundations of sentiment analysis. Representative publications relevant to this area are critically analyzed and summarized. Section 2.1 provides a broad overview of sentiment analysis as well as categorization of its tasks, granularities, and difficulties. Section 2.2 provides an in-depth analysis of sentiment classification granularities involving different types of levels. Section 2.3 reviews the approaches toward sentiment analysis. In Section 2.4, we provide the Ensemble Learning approaches to sentiment analysis. In section 2.5, we describe the principle of the Genetic Algorithm. Finally, we review several research hypotheses that are associated with the research aims and objectives and the research gaps identified.

2.1 Sentiment Analysis

This is one of the areas of Natural Language Processing (NLP) where it is assumed that textual information consists either of facts or opinions [32]. The purpose of sentiment analysis is to analyze people's sentiments, opinions, attitudes, evaluations, and emotions in favor or against organizations, products, individuals, services, topics, events, and attributes [32].

The term sentiment analysis is believed to have first appeared in the late-20^{th} to the early-21^{st} century where predictive judgments regarding text for the analysis of financial markets were published in certain articles [33,34]. By the year 2002, the

possibility of using the semantic attribute of adjectives in classifying the entire opinion of a document was explored by Turney [7]. This was the turning point of research concerning sentiments or opinions inherent within textual information as an integral part of the field of NLP. The study by Nasukawa and Yi [35] is believed to be the first literature where the concept of sentiment analysis was first addressed. Furthermore, the study by Yi et al [36] in the same year was targeted at addressing the disciple of sentiment analysis. Sentiment analysis is categorized basically into two classes namely, subjectivity classification and emotion recognition [37,38,39,40]. These studies equally took cognizance of similar processes of knowledge discovery and problem-solving structures. Sentiment analysis, also known as opinion mining is frequently used in review articles when referring to high-level summarization of techniques and concepts.

Before addressing what sentiment analysis is, it is required to first understand how opinion and sentiment are related. According to Kim and Hovy [41], opinion can be represented by a four-tuple as (topic, holder, claim, and sentiment). Further, Liu et al. [42] defined an opinion as a kind of subjective expression forming a four-tuple of (entity, aspect, sentiment, holder, time), where sentiments are regarded as a kind of attribute of the expression. Specifically, the definition by Kin and Hovy [41] relates opinions with claims, holders, topics, and sentiments, whereas the definition by Liu et al. [42] integrates opinions with more specific targets such as entities, aspects, and times, side expressions, holders and sentiments. As a result of computational cost overheads, some studies consider sentiment analysis as a classification problem which purpose is to determine sentiment polarity. According to Liu et al. [42], sentiment polarity is the orientation of sentiment usually from a piece of text

(opinionated) that contains claims or expressions [37,40,43-45]. In this regard, sentiment analysis studies are categorized into three including document-level, sentence-level, or aspect-level, which depend on the granularity of the pieces of text. Sentiment analysis, as a classification problem, can be either binary or multi-class classification as the number of polarity labels might be. Binary polarity which consists of two classes is widely used in the classification of text documents. However, sentiments are more complex than just two classes in real life. The term multi-class is used when polarity consists of more than two classes, that is, from three above.

By the year 2001, it became obvious that sentiment analysis is indispensable. Several studies dealing with opinion mining have increased and since then, several publications have been made on the topic. With advancements in the field of machine learning, many techniques have been developed for efficiently handling sentiment analysis problems. Several review papers have been published to evaluate existing sentiment analysis techniques and examine the progress made so far in this area.

2.1.1 Tasks of Sentiment Analysis

According to Pang et al. [46], the main tasks of sentiment analysis consist of four sub-tasks including opinion extraction, sentiment classification, polarity determination, and summarization. On the other hand, Liu and Zhang [47] enumerated the main tasks of sentiment analysis to include subjectivity and sentiment classification, aspect-based sentiment analysis, sentiment lexicon construction, opinion summarization, analysis of comparative opinions, opinion search, and retrieval, opinion spam detection, and quality of reviews. Ravi and Ravi

[48] listed seven tasks of sentiment analysis as subjectivity classification, sentiment classification, review usefulness measurement, lexicon creation, opinion word and product aspect extraction, opinion spam detection, and various applications of opinion mining. In our study, four tasks of sentiment analysis are considered as discussed below and shown in Figure 2.1.

2.1.1.1 Opinion Polarity Classification

Opinion polarity classification aims to categorize a piece of textual data such as a document, sentence, or aspect, into some polarity labels. As noted, the labels could be binary such as positive or negative, or multi-class such as fine-grained or scaled ratings [7,35,49,50]. Polarity classification is the primary and fundamental task in sentiment analysis which has been extensively researched in recent years. The techniques for achieving polarity classification are classified into lexicon-based and machine learning-based approaches. In early studies when the main focus was just to identify sentiment orientation from words and phrases, the most dominant approaches were pure lexicon-based. However, recent studies make use of pre-developed sentiment lexicons as features and not for determining the sentiment polarity of a target [43,51]. Since lexicon-based approaches involve substantial human effort in the lexicon generation process, many researchers have opted for machine learning approaches because they detect sentiment polarities automatically [46,52-55]. Recent studies have focused on deep neural network models rather than supervised learning approaches because of the improvements in computational speed and power. [51,56-58]. As this task is going to be a centerpiece of this thesis

2.1.1.2 Subjectively Classification

The subjectivity detection task checks to see whether a text is subjective or objective. A text is said to be objective if it carries some factual information such as “the

weather is hot". On the other hand, subjective texts convey an individual's personal opinion or views, such as "I like hate hot weather". Subjectivity classification is the task that determines whether a sentence is objective or subjective. Subjectivity classification differs from polarity classification because the former judges whether some textual data such as a sentence, clause, or phrase is opinionated or not. It was Wiebe et al. [59] who originated the term "subjective". Subjective can be determined by an understanding of whether the intent or purpose behind a sentence is to be factual or otherwise. According to Liu et al. [42], subjectivity classification tasks determine whether a sentence is objective or subjective. This means that the task of subjectivity classification could be said to be more like the task of a sentence-level binary polarity classification. In related literature, Maas et al. [37] employed an unsupervised probabilistic model with a supervised sentiment component computed by LR predictor to perform sentence-level subjectivity detection for movie review data. Similarly, Nakagawa et al. [45] developed an unsupervised probabilistic model with Conditional Random Fields (CRFs) and hidden variables for the classification of subjectivity at the phrase and sentence levels, including the detection of polarity reversals in data from different sources. Wang et al. [60] utilized the improved Fisher's discriminant ratio-based feature selection method to perform subjectivity classification. The proposed set of features together with words appearing in positive and negative texts were deployed as train sets for the Support Vector Machine (SVM). Relatedly, Benamara et al. [61] proposed subjectivity classification for discourse-based sentiment analysis at the segment level. Rustamov et al. [62] applied a Fuzzy Control System and Adaptive Neuro-Fuzzy Inference System for sentence-level subjectivity detection in a movie review. In a related study, Chenlo and Losada [63] presented used polarity and subjectivity classification to study different features

of a sentence. The study reported that when unigrams or bigrams are combined with sentiment lexicon features, improved performance for subjectivity and polarity classification is achieved. In another research, Khan et al. [64] used SVM to develop a new semi-supervised framework for subjectivity detection which can determine feature weights and a lexical approach for simultaneous selection of polarity on different parts of speech. Hathlian et al. [65] developed another model where sentiment analysis is combined with subjective analysis on Arabic social media posts to determine whether or not people are interested in a defined subject. It could be observed from these studies that the task of subjectivity classification is considered more as a preprocessing phase for further sentiment analysis and not as an independent task.

2.1.1.3 Opinion Summarization

This task is meant to summarize a large group of opinions on a topic. Usually, the opinion encompasses a variety of perspectives, aspects, and polarities. This task is important in situations when there is the need to make a decision but a single opinion is not reliable. The main features are extracted from one or several documents and the corresponding sentiments [66]. There are two aspects to this: single-document and multi-document summarization. In single-document summarization, the internal facts present in the document are analyzed to extract the pieces of text which describe the entire text better. Multi-document summarization on the other hand groups the different sentences which express sentiments related to those entities or features after they have been identified. The final summary is then displayed in form of a graphic or a text to show the main features/entities with the level sentiment around each of them. Some studies on opinion summarization focus on the extraction aspect of the task to cover terms and phrases [67,68], while others focus on the

aspect-level polarity classification. For each product feature extracted, Hu and Liu [69] return every negative and positive sentence while assigning a count that tracks the number of positive and negative opinions corresponding to each feature. Meanwhile, Meng and Wang [70] considered the most repeated terms or phrases to represent the summary of a product feature, while Lu et al. [71] addressed the issue of aspect ratings for each product. On the other hand, Nishikawa et al. [72] developed an algorithm for the summarization of opinions. The algorithm simultaneously takes cognizance of content and coherence. Their proposed algorithm works by directly searching for the optimum sequence of sentences and extracts and orders the sentences which are present in the set of the input document. In another study, Condori and Pardo [73] developed another strategy for content selection for the production of extractive summaries. The authors presented a template based on the Natural Language Generation (NLG) system which can generate abstractive summaries of opinions. It is believed that extreme reviews have a great influence on customer decisions, and this makes extreme opinions to be very valuable in the task of opinion summarization. This is largely because the main aim of the task is to summarize views regarding an object for decision-making.

2.1.1.4 Emotion Recognition

Emotion recognition is an extension of sentiment polarity classification which serves in analyzing more fine-grained emotional states. The sentiments in some kinds of textual data may not fit well into binary categories such as positive or negative in real-life applications. Rather, some research use labels of emotions such as disgust, anger, sadness, fear, surprise, and happiness [74] or the personality traits namely, neuroticism, agreeableness, extraversion, conscientiousness, and openness [75], for analysis. The following activities are involved in such studies: recognizing emotion

in speech signals [76], detection of moods in lyrics [40], analysis of the mental state in diaries [77], detection of personalities in essays [44], recognition of human emotions in audio-visual information [78], identification of emotion from body movement [79], or detection of emotion by eye-tracking [80]. Some of the studies take cognizance of linguistic and psychological factors in the process of analysis and make applicable contributions rather than technical contributions.

2.1.2 Difficulties in Sentiment Analysis

The main aim of sentiment analysis is to reveal the excitement, attitude, expression, opinion, viewpoint, expression, and emotion towards a particular entity. Nevertheless, the writer or author's emotional attitude and state of mind usually influence the outcome of sentiment analysis. Over 7000 research projects and articles have been written on the topic of sentiment analysis. However, many challenges still abound in this area and some have been identified. Some of the challenges identified include:

- At times, the negative sentiments are expressed in a sentence without making use of known any negative words. According to Wawre and Deshmukh [81], using a particular opinion word for a positive case in one context and then using the same opinion word for a negative case in another context is a challenge in sentiment analysis. Similarly, Weitzel et al. [82] agreed that most times, individuals express their opinions differently as the need arise, and different meaning can be attributed to the opinion words. According to Boldrini et al. [83], having blended text formats, numerous data sources and domains, using informal language, multilingual resources, incorrect grammatical spellings, and using slang are some of the challenges in sentiment analysis.

- It is difficult to handle sarcasm because some reviews express dissatisfaction regarding a product or service sarcastically, though with positive language. Though some researchers have attempted to address the issue of sarcasm [84], this concept is yet to be properly incorporated in sentiment analysis systems. In many cases, it can be seen that sarcasm reverses the polarity of the sentiment words used in a document. Maynard and Greenwood [85] investigate the effect of sarcasm on sentiment analysis and show that correctly identifying sarcasm improves the performance of sentiment analysis systems. Weitzel et al. [82] opined that one of the challenges of the sentiment analysis field is Figurative Language. Some examples of Figurative Language include sarcasm, irony, analogy, and ambiguity. The Figurative Language extends the meaning of a language by deviating from the original usage of a word based on the author's perspective. Some reviews consist of a large number of comparative opinions, which poses a challenge to sentiment analysis. Instead of offering a review directly on an entity in question, some users choose to express a comparison between various entities. [86]. For example, "The original version of the software was more robust." At face value, one would think the review has a positive polarity; however, further analysis would show that the user has a negative polarity against the new version of the software compared to the original version. Comparative opinions are often useful in identifying the strengths and weaknesses of products or services to enable an improvement to be made on future versions of the products/services.

- Sentiment analysis is equally confronted with the challenge of intensifiers and downtowners. An Intensifier is a word that causes the sentiment score to increase in the case of positive opinions and causes the sentiment score to decrease in the case of negative opinions. An example is “Extremely” as used in a review such as “Extremely great software.” On the other hand, a downtoner is a word that moderately reduces the sentiment score or moderately increases the sentiment score in the case of positive opinions and negative opinions, respectively. An example includes using “quite” in case of positive opinions in a review such as “quite good software.” Taboada et al. [87] opined that the level at which intensifiers intensify varies from one case to another. The intensity of the intensifier is affected by the sentiment score of the word being intensified.
- Though sentiment lexicons are important features in sentiment analysis, creating them is a challenging task. It is difficult to use human annotators to annotate semantic intensity scores built with a lexicon because of the problem of consistency. Although various lexicon sources are available, it is difficult to determine which one offers the most reliable semantic scores.
- Another challenge encountered with sentiment analysis is that the same product can be referred to by several names even within the same document. Though automatic name entity resolution attempts to solve this issue, it is yet to be effectively resolved. It is a challenge to accurately handle anaphora resolution in text mining, and this is one of the challenges in sentiment analysis.

- There is the need to identify relevant text to each entity being referred to in a document. However, the current methods used in identifying relevant text in a document do not offer very good accuracy.
- Handling text with spelling, grammatical, and punctuation mistakes, missing punctuation, and slang is also among the challenges faced in sentiment analysis systems.
- Several statements about factual entities convey sentiments; however, only subjective statements are often considered in the current methods of sentiment analysis. There is, therefore, a need to handle factual statements in such cases.
- Ambiguity in comments is a challenge in sentiment analysis. Some authors and fond of using vague and confusing words, and this poses a challenge. It is often difficult to determine the meaning of such ambiguous statements.
- The issue of translation is another challenge in sentiment analysis. Some reviews are made in other languages and need to be translated into English. This may pose a challenge as Western, Asian, and African sentiments are significantly different.

2.3 Levels of Sentiment Analysis

Sentiment analysis is commonly categorized based on granularity. There are typically four levels of granularity involved: concept level, aspect level, sentence level, and document level, as shown in Figure 2.1. The source of a sentiment (document, sentence, aspect, or concept) determines the particular technique to be implemented for sentiment analysis. The following subsections explain in detail the levels of sentiment analysis.

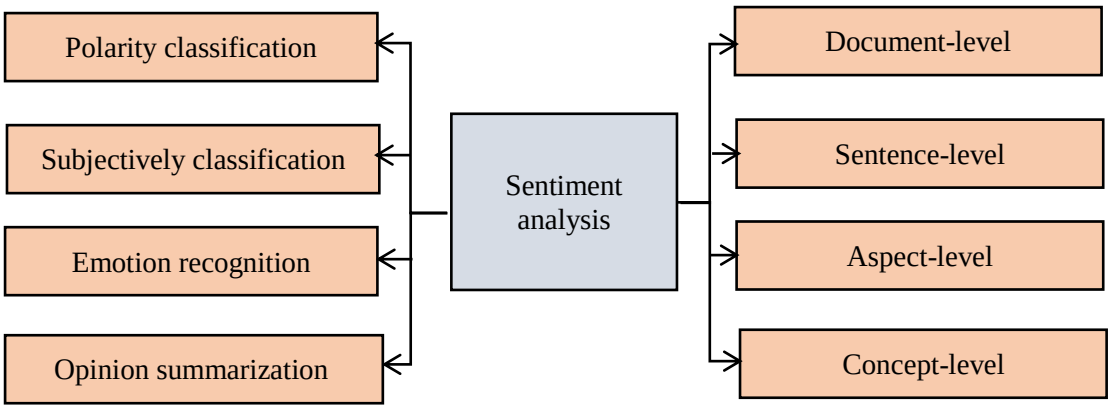


Figure 2.1: Sentiment analysis tasks and levels

2.3.1 Sentiment Analysis at the Document Level

The target at this level is the entire document, where a single sentiment is assigned to the whole document [37,50,88]. This follows the assumption that only a single sentiment is expressed in the entire document. It is argued that it is not usual for a document to carry only one opinion, and as a result, the analysis should be done at the component levels [32]. Some researchers argue in favor of this assumption, especially for reviews where a concluding remark is required about a product which is often a summary of different opinions. This assumption is also considered valid in the case of the financial news domain where the news consisting of positive or negative sentiment is usually reflected in a buy or sell outcome. Studies have indicated that an enhanced performance could be achieved in sentiment analysis at the document level if the focus is directed on diverse attributes in both short and long documents alike. It is assumed that short documents such as Tweets discuss a single topic due to their word limit. As a result, more attention is paid to identify those expressions that convey opinions or sentiments [43,57]. To determine the sentiment polarity of long documents, including lyrics [40], essays [44], or diaries [77], the sentiment analysis systems focus on exploring how the sentiment in each factor

contributes to the overall sentiment polarity of the document. Such individual factors include, for example, topics [77] or writers [44].

2.3.2 Sentiment Analysis at the Sentence Level

The task of sentiment analysis at the sentence level is to classify the different opinions contained in each sentence. These individual opinions are usually identified by punctuations: the question mark, the full stop, the exclamation marks, etc. This level of sentiment analysis assumes that each sentence conveys just one sentiment. Not all the sentences in a text are subject to this assumption, as some sentences do not convey a particular sentiment. There are two classification tasks for sentence-level sentiment analysis [47]. The first is subjectivity classification which aims is to differentiate sentences that convey information and facts (objective sentences) from those that convey views and opinions (subjective sentences). The second category is termed polarity classification of the sentences into either positive or negative classes. As noted, sentiment analysis at the document level usually produces general and non-specific information. This makes it necessary for a more specific analysis at the level of individual sentences. Several studies analyze opinions based on the individual sentences contained within the document [47,89-99]. Sentiment analysis at the level of sentences helps in eliminating noise and polarity shifts from a document and is the most preferred method when there is the need to capture the different opinions contained in one document.

2.3.3 Sentiment Analysis at the Aspect Level

It has been argued that the assumption of document-level sentiment analysis where an entire document is said to contain only a single sentiment is not realistic [67, 100, 101]. At the aspect level, sentiment analysis identifies each aspect contained in the text, and then the sentiments and or opinions are classified based on each aspect.

Aspect level sentiment analysis assumes that all opinions are targeted at specific topics or objects. This is a distinctive feature between this form of sentiment analysis and the others. For instance, aspects in movie reviews could be in the form of music, actors, or lights. Consequently, customers expressing opinions concerning a movie do so using these aspects such as their opinion about the actors or choice of music. There are two tasks involved in aspect-level sentiment analysis, including aspect extraction and sentiment classification. Aspect extraction makes use of probabilistic models such as Linear Discriminant Analysis (LDA) [101], or regression analysis [100]. On the other hand, sentiment classification uses tools such as neural network models [102]. Most studies on aspect-level sentiment analysis usually focus on the review domain since it is typical for review data to involve several aspects such as price, model, and color for a review of an android phone; and they assign a corresponding rating for each aspect which serves as a label for evaluation [100-102].

2.3.4 Sentiment Analysis at the Concept Level

Concept-level sentiment analysis aims to conduct sentiment analysis beyond the usual word-level analysis. Its objective is to ensure an efficient transition from unstructured textual information to structured data that can be processed by a machine regardless of the domain involved. This method uses semantic networks or web technologies for text analysis to ensure that conceptual and affective information associated with natural language opinions are aggregated. This approach relies on semantic knowledge bases instead of keywords and word co-occurrence counts. For instance, the approach relies on the implicit features of natural language concepts, unlike syntactical techniques. Furthermore, concept-based approaches are capable of extracting sentiments that are implied and cannot stand alone but are only linked to

other concepts. The method of opinion mining based on concepts was presented by Cambria et al. [103]. Concept-level analysis of emotions makes inference using conceptual information relating to emotions and sentiments inherent in natural language. Poria et al. [104] proposed a new approach that improves the accuracy of polarity detection where comments are analyzed at the conceptual level. The approach integrates machine learning techniques with linguistic and common-sense computing. The proposed approach appears to produce higher accuracy than the usual statistical methods. Tsai et al. [105] built a concept-level dictionary using common-sense knowledge. Several related studies, such as [106-112], have focused on concept-level sentiment analysis.

2.4 Approaches to Sentiment Analysis

The decision of selecting a technique to use in sentiment classification is a vital step in opinion mining. Polarity classification (also sentiment classification) involves the process of deciding the polarity of an object, such as document, sentence, etc., to determine if sentiment expressed towards a subject is positive, negative, or neutral. Polarity classification has been widely used in social media, blogs, product reviews, news articles, forums, etc. Three methods of sentiment classification appear in the literature. These include machine learning approaches, hybrid approaches, and lexicon-based approaches, as shown in Figure 2.2. The approaches are discussed in the following section:

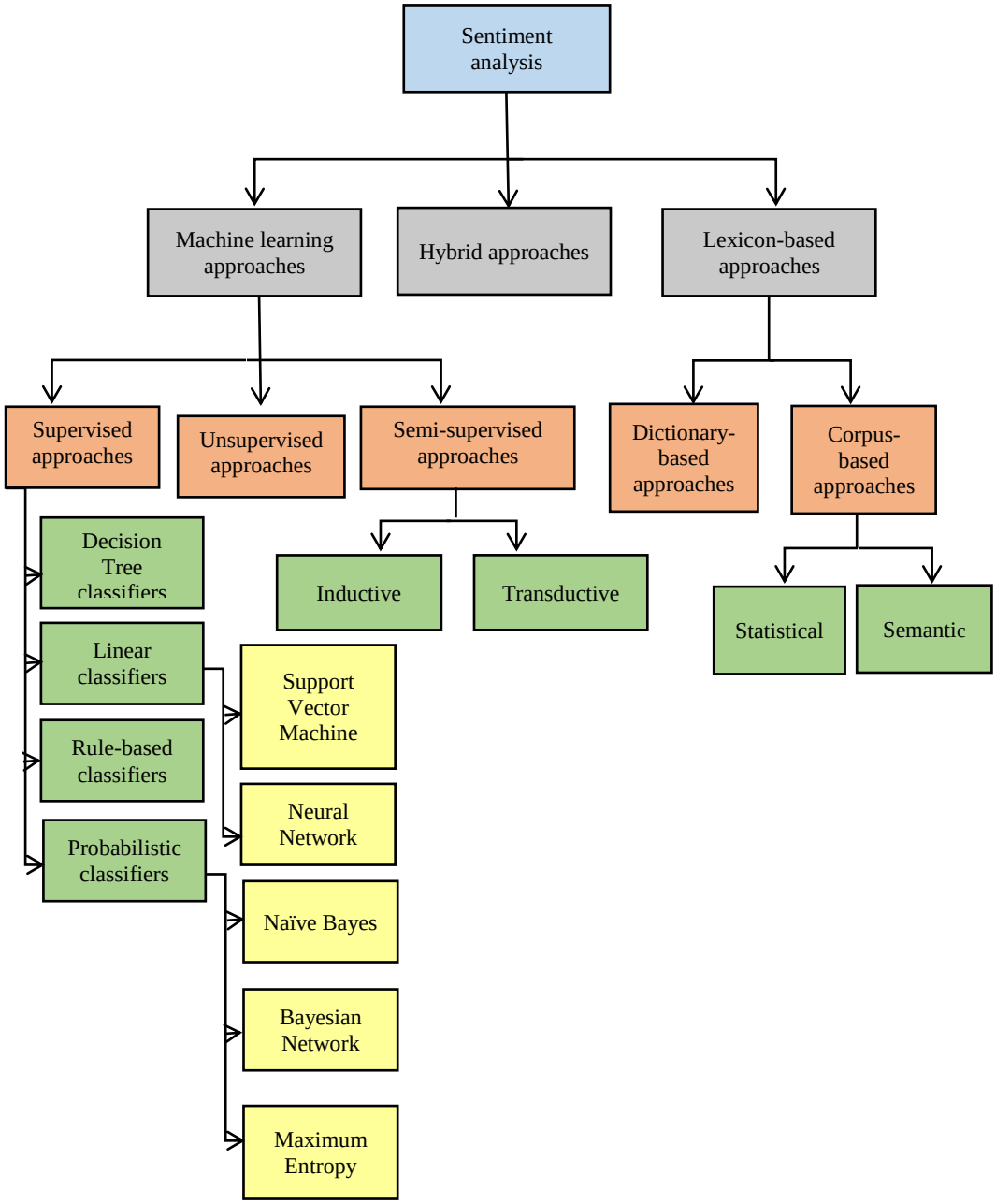


Figure 2.2: Sentiment analysis approaches [115]

2.4.1 Machine Learning Approaches

These approaches are the most used in sentiment analysis. In this approach, the sentiment analysis task is defined as a classification problem. Then machine learning algorithms are utilized to choose linguistic features that help to classify the given text

into various classes. The machine learning approaches can be classified into three main categories: supervised, semi-supervised, and unsupervised learning approaches. The main distinction among these approaches is whether the training data is labeled or not. Natural language processing plays a vital role in the process of feature extraction. Some important features used include (i) part-of-speech information such as adjectives, adverbs, and nouns [113,114]; (ii) syntactic dependencies [47,63,87]; (iii) terms, that is words or n-grams and their frequency; and (iv) negations which change the meaning of any sentence. Figure 2.3 gives the types of supervised learning-based approaches to sentiment classification. The three main categories of machine learning-based approaches are discussed in the following subsections.

2.4.1.1 Supervised Learning-Based Approaches

These approaches make use of labeled training documents to classify text automatically. Supervised learning approaches deal with constructing classification models that can predict the category of documents based on predefined class labels.

There are four basic types of supervised learning classifiers as discussed below:

- Decision tree classifiers: These algorithms build a tree-like structure of hierarchies with true/false queries in line with the categorization of the training document.
- Linear classifiers: They determine suitable separators which can separate the space into different classes in an optimum way.
- Probabilistic classifiers: They classify documents based on maximum probability. Examples include Naive Bayes, Bayesian network, and maximum entropy.
- Rule-based classifiers: They divide the data into segments bases on a set of rules. In the training phase, the rules, which are usually in the form of

"IF condition THEN conclusion" statements are generated for classifying documents into annotated categories.

Pang et al. [50] deployed NB, Maximum Entropy (ME), and SVM to classify movie reviews into binary classes in pioneer research involving sentiment analysis. The authors concluded that SVM produced the highest accuracy. The SVM is considered the most appropriate method for text classification [116] because of its strong theoretical base and its success in sentiment classification [117]. The performance of SVM and NB were compared with that of Artificial Neural Network (ANN) by Moraes et al. [118] for sentiment analysis using both balanced and unbalanced datasets. It was found that the performances of NB and SVM were affected by the unbalanced data. In another research, Bilal et al. [119] examined the performances of three algorithms, namely, NB, DT, and NN, in the classification of Urdu and English opinions in a blog. They found that NB performed better than the other two. Relatedly, Bhavitha et al. [120] conducted experiments to compare the performances of these algorithms: NB, SVM, and KNN; with the outcome showing that with a large feature set, SVM yielded the highest accuracy than the others; while Naïve Bayes produced the highest accuracy with a small feature set. In a separate study, Poornima and Priya [121] evaluated the classification accuracies of SVM, Multinomial NB, and LR. The results showed that LR had a higher accuracy than the other algorithms in the task of Twitter sentiment analysis.

The study by Yasen and Tedmori [122] involved eight algorithms, namely, NB, KNN, Bayes Net (BN), Ripper Rule Learning (RRL), SVM, RF, SGD, and DT. It was concluded that each algorithm proved some level of efficiency and reliability in the task of sentiment analysis.

2.4.1.2 Unsupervised Learning-Based Approaches

Unsupervised learning methods do not use training data in the process of classification, neither do they depend on existing data labels. It is difficult and expensive to access training data in some cases, and so unsupervised learning methods overcome this challenge. These methods consider a set of training samples where information regarding the output is unknown and only the input value is specified. There is no need for large human-annotated training data when using unsupervised learning methods. Syntactic patterns and lexical-based methods [123] are the widely used strategies under the unsupervised learning methods. Lexical-based methods consist of a set of predefined words which are associated with a particular sentiment. Turney [7] presented an unsupervised learning algorithm that could classify reviews into two classes, recommend/recommend, based on the mean number of positive/negative phrases present in the review.

On the other hand, the lexicon-based method employs a dictionary of words, phrases, their associated strength, and orientations and then computes sentiment score by incorporating intensification and negation [87]. Originally, the method was deployed for sentence and aspect-level classification of sentiment [69,124,125].

2.4.1.3 Semi-supervised Learning-Based Approaches

These approaches appear between unsupervised learning and supervised learning [126,127], and they infer a function from both labeled and unlabeled data. Unlabeled data is very useful [128], such that when used together with a small amount of labeled data, the accuracy and precision of the model are enhanced. The structure and feature space are some of the properties of unlabeled data exploited for classification tasks. Based on an assumption, one or more regularizer is formed over the unlabeled

data and then used to train and evaluate the model. Semi-supervised learning can be used for classification [129], feature reduction [130], hashing techniques [131], regression [132], and clustering [133,134] tasks. The approaches can also be used for metric learning [135], such as the research by Hoi et al. [136], where semi-supervised learning was used to evaluate a distance metric for image retrieval.

Although supervised learning models produce a good performance for sentiment analysis, the requirement for training data is a challenge. This is why several studies on supervised learning methods use the same datasets over and over. There are large amounts of social network unlabeled data available, although with noisy labels. Consequently, semi-supervised learning methods are becoming popular because of the availability of these unlabeled data from social networks. There are two categories of semi-supervised learning approaches, including self-training or co-training-based and graph-based approaches. Self-training and co-training approaches build models using a small amount of training data while cashing in on a larger volume of unlabeled or noisy labeled data. With this approach, the size of the training data can be extended without human interference. Meanwhile, the graph-based approach utilizes series of models which use graph structure to handle sentiment classification problems. Among the graph-based algorithms, label propagation is one of the widely used.

2.4.2 Hybrid Based Approaches

Hybrid methods combine more than one machine learning approach to form a single classifier. For instance, Prabowo and Thelwall [137] developed a hybrid classifier with five different classifiers including General Inquirer, Rule-Based, Statistics-Based, Induction Rule-Based, and SVM.31. In another research, De Albornoz et al.

[138] developed a hybrid classifier using machine learning techniques and lexical rules. The classifier is meant for the classification of sentences' polarity and intensity. The approach can determine the polarity class and intensity of each sentence, with the capability to address word ambiguity. The hybrid approach introduced by Ghiassi et al. [139] analyses sentiments in Twitter text using n-gram, a Dynamic Artificial Neural Network (DAN2), or SVM algorithm. In another research, Poria et al. [104] developed a hybrid approach made up of linguistics, commonsense computing, and machine learning to handle concept-level sentiment analysis. Furthermore, Appel et al. [96] proposed another hybrid method using semantic rules, improved negation management, and an enhanced sentiment lexicon to identify sentiment polarity at the sentence level. The system computes the intensity of sentiment polarity with fuzzy sets. In the approach proposed by Al Amrani et al. [140], sentiments are classified using Support Vector Machine, Random Forest, and RFSVM algorithms. Equally, Asghar et al. [141] developed a framework for the analysis of tweets using a slang classifier (SC), emoticon classifier (EC), and general-purpose sentiment classifier (GPSC). On the other hand, Rajeswari et al. [142] presented a model that combines a lexical approach (SentiWordNet) with algorithms such as SVM, DT, LR, and NB for sentiment analysis.

2.4.3 Lexicons Based Approaches

Opinion words, also known as sentiment words are an integral part of sentiment analysis because they are what algorithms need to mine positive or negative opinions. As an example, awesome, splendid, good, amazing, and fantastic are positive, while awful, bad, worse, poor, and disgusting are negative. In some cases, more than just words is combined to express positive or negative opinions such as in

quite amazing, very awful. The list made of such words and phrases is called a sentiment lexicon (or opinion lexicon) [46].

In some cases, the lexicons comprise of Part-of-Speech in which the words are segmented into adjectives, adverbs, nouns, and verbs Turney [7]. Ding et al. [124] introduced a lexicon approach that improves on the lexicon-based method developed by Hu and Liu [69]. The approach makes use of the information content from several sentences and just one sentence. This strategy determines the polarity of opinion words by making use of some linguistic properties of natural language expressions, and user inputs or domain knowledge is required in advance. The approach addresses the challenge of multiple and conflicting opinion words by evaluating the distance between opinion words and the product feature. Takamura et al. [143] proposed a method of polarity extraction for phrases where lexical networks that connect similar words with two links were built. That is, words linked with the same and different polarity. The method is capable of classifying adjective-noun phrases with unseen words. In another research, Mohammad and Turney [144] developed a lexicon that combines sentiment polarity with any of these eight emotion classes including anger, anticipation, disgust, fear, joy, sadness, surprise, and trust for each word. On the other hand, Lin et al. [145] proposed a framework for cross-language opinion lexicon extraction which uses the mutual-reinforcement label propagation algorithm. In related research, Zhang and Singh [146] developed a framework with the ability to generate a domain-specific sentiment lexicon. It is to be pointed out that adjective and adjectival phrases are the most influential words in sentiment analysis, and this is why most early research paid more attention to the use of qualities [69,147,148]. Furthermore, Nasim et al. [149] used TF-IDF and lexicon-based features to develop a

model that can conduct sentiments analysis expressed by students in their textual feedback. Equally, Han et al. [150] developed a lexicon-based framework for sentiment analysis. The system uses the domain-specific sentiment lexicon from the proposed domain-specific sentiment lexicon generation method. In another research, Rezaeinia et al. [151] developed a model that improves the accuracy of an existing model of sentiment analysis which was based on pre-trained word embeddings. The model combines four approaches including lexicon-based, POS tagging, word position algorithm, and Word2Vec/GloVe. A study by Machová et al. [152] proposed a lexicon-based sentiment analysis method that uses nature-inspired optimization algorithms to check out for optimal polarity values inherent in lexicon words.

2.5 Ensemble Learning Approaches

Ensemble learning approaches are also called learning multiple classifier systems and are used to handle classification tasks [153]. In this approach, several models are fit on the same training data to develop a system to perform classification and predictions. Each machine learning algorithm and model has its limitation, and that is why it is important to combine (ensemble) several models to complement each other. The goal of combining learning algorithms is to achieve higher accuracy compared to using just one classification model [154]. Though, it has been argued that it is not always true for an ensemble algorithm to perform better than a single learning algorithm [154,155]. The various types of ensemble learning algorithms are discussed below:

- Sequential methods: Under these methods, base learners are generated sequentially, consisting of data dependency. That is, each data depends

on the previous data in the base learner. An example of this method is "boosting".

- Parallel methods: These methods generate the base learner in a parallel order without data dependency. An example is "stacking".
- Heterogeneous ensemble methods: These methods use the same data to build different types of classifiers, and the method of feature selection differs for the same data. The final result is obtained by taking the mean of the models. An example of this method is "stacking".
- Homogeneous ensemble methods: These methods build the same type of models with different datasets for each model. The results from each model are then aggregated to form a single model. The method is appropriate for large datasets, and the method of feature selection is similar for all data. Examples of these methods are "bagging" and "boosting".

Ensemble learning algorithms can be divided into two categories: the common and combining methods. The details of each method are described below.

2.5.1 Sentiment Analysis through Most Popular Ensemble Methods

Some commonly used ensemble methods are discussed in this subsection. They employ a subset of training data in the process of classification such as bagging [156], boosting [157], and RF [158].

- Bagging/bootstrap aggregation: It is a method used in decision trees to decrease the rate of variance in learning algorithms. For instance, a single tree will have a higher variance than the average prediction of combined trees. Bootstrapping is a resampling technique where random

samples of smaller and same sizes are obtained from the dataset. In the process of bagging, classification models are built using the bootstrap samples of the training set and the individual outputs from each classifier then combined by a plurality vote [159]. In research by Qadir and Riloff [160], a bagging algorithm was deployed to classify five classes of Twitter hashtags including affection, anger/rage, fear/anxiety, joy, and sadness/disappointment. The hashtags were manually selected to represent the emotion corresponding to each class. Ten hashtags were learned in hundred iterations by the bagging algorithm which was then used to search the seed hashtags from tweets for labeling. In another study, Prusa et al. [161] examined the performance of ensemble algorithms (bootstrapping and bagging) and feature selection or resampling with Twitter data. Two feature selection methods namely, the chi-square (CS) and the Receiver Operating Characteristic (ROC) were used to select the important attributes for model construction. Furthermore, other models were constructed and evaluated using random re-sampling (cross-validation). It was found that the combination of bagging with feature selection produced better sentiment classification of tweets than the resampling method.

- Boosting: This method is used basically for converting weak learners to strong learners. In the learning process, the models are constructed iteratively by choosing the training subset of the current model based on the performance of the previous model. The model with higher weights is considered the misclassified model. Some examples of boosting algorithms include AdaBoost and Stochastic Gradient Boosting (SGB).

In the study by Celikyilmaz et al. [162], the sentiments in tweets were classified into two groups: polar and non-polar based on positive or negative sentiments, respectively

- Tree ensembles: These algorithms were first developed in 1995 by Ho [163] and are a combination of decision trees. The Boosted Trees developed by Friedman [164] is one of the widely used ensembles of trees. Random Forest, a tree bagging algorithm developed by Breiman [158] combines several random trees through bagging and voting [163]. In the training process, several random decision trees are generated and data is randomly selected on which trees are fit. At the end of the training, each tree cast a vote and the predicted class is that which has the majority vote. Random Forests use two methods namely, bagging and random subspace projection [165]. In the latter method, features are used instead of data [166], and at each split, a random subset of features is selected to curtail overfitting.

2.5.2 Sentiment Analysis through Simple Ensemble Methods

These methods combine the outputs from multiple models to predict the class as in majority voting and weighted voting.

In majority voting [167], all outputs from each classification model are taken as input and the label with the majority of votes is predicted as the class. The decisions on whether to place an object into a class are taken when more than half of the classifiers vote for it. If not, the input is rejected. In a related study, Gryc and Moilanen [168] classified the sentiments from a dataset into three classes namely, positive, neutral, and negative using three features. The features include the unigram

bag-of-words, the social network feature, and the feature for sentiment scoring. Two statistical classification methods: Naïve Bayes Multinomial (NBM) and LR were used for prediction. Furthermore, two ensemble learning algorithms: stacking and majority voting were employed for prediction. It was found that stacking yielded a lower accuracy compared to using a single machine learning algorithm and vice versa for majority voting. In a related study, Wan and Gao [169] developed an ensemble classification model for sentiment analysis where random resampling was run to obtain balanced classes. Information gain was deployed to choose the first 656 most important features, and five learning algorithms: NB, SVM, Bayesian Network, C4.5 Decision Tree, and RF formed the ensemble methods. Relatedly, Chalothom and Ellman [170] conducted different tests involving base learners, sentiment lexicons, and ensemble methods. Majority voting and stacking methods were tested and it was found that majority voting with three models produced better scores for tweets.

In weighted majority voting, a weight is assigned to each classifier relative to its performance to correct predictions. Weights are utilized during the aggregation of votes and the impact is increased or decreased to reflect correct/wrong predictions by each classifier. There are two methods of choosing weights: as a constant for each classifier or different weights for each class and each model corresponding to accurate predictions. In a related study, Aziz and Dimililer [171] employed weighted majority voting in research that sought to enhance the predictive accuracy of sentiment analysis in Twitter. Six classifiers were used as base classifiers including LR, NB, DT, SGD, SVM, and RF. Usually, weights are chosen as accuracy and can be static or dynamic depending on the classified instance [172]. In a study by Kolyal

et al. [173], weighted majority voting was deployed to determine the association between the event's time and document creation time after extracting the events. Research has shown that the classification accuracy of ensemble classifiers in majority voting can be enhanced using weighted voting approaches. A related example is a study by Nazeer et al. [174] where weighted majority voting was used to identify sentiments within tweets. Classifiers including NB, RF, and LR were employed and their performances examined.

Apart from the majority voting strategy, there are other forms of voting approaches known as unanimity and plurality voting [153]. Unanimity voting is where each pattern is specifically labeled as a class provided all the ensemble members agree on the label. Further, plurality voting requires that half of the members plus one additional member agree on a class label.

2.5.3 Sentiment Analysis through Meta-Classifier Ensemble Methods

The commonly used meta-learning methods are described in this section.

- Stacking [175] combines classifiers built by different inducers and attempts to distinguish which classifiers are the most reliable. It is aimed at achieving the best generalization accuracy. In stacking, the original input attributes are not utilized but rather, the predicted values of the model are used as the input attributes without altering the target attributes. Each of the base classifiers classifies the test instance at the initial stage. Then, the classifications are forwarded to a meta-level training set where a meta-classifier is generated. The original dataset is partitioned into two subsets where one of the subsets is used as the meta-dataset while the second is for building the base-level models. The

consequence is that the meta-classifier makes predictions that can be generalized for the base-level learning algorithms. To improve the performance of stacking, the output probabilities corresponding to every class label generated by the base-level classifiers are used. This requires that the number of input attributes is multiplied by the number of classes. According to Wolpert [175], stacking is meant to utilize the output from the base classifier to serve as input for the meta-classifier. In a study by Džeroski and Ženko [176], the performance of stacking was evaluated by fitting ensembles of classifiers which showed that stacking performs better compared to the ensemble by cross-validation. To improve on stacking, the authors proposed a multi-response model tree that can learn at the meta-level. The results showed that the system performed better than normal stacking approaches and better than selecting the best classifier by cross-validation.

- Arbiter Trees are built following a bottom-up approach [177]. Pairs of classifiers are induced and a new arbiter is induced from the output of two other arbiters. That is, given any *k* classifiers, there consist log 2(k) levels corresponding to the generated arbiter tree. The following steps are followed to create an arbiter. Given any pairs of classifiers, their training dataset union is classified by the two classifiers. The predictions of the two classifiers are compared by a selection rule to select instances that form the training set for the arbiter. The set with the same learning algorithm is considered for inducing the arbiter. The arbiter provides an alternate classification in situations where *n* there are diverse classifications by the base classifiers. The arbiter then provides an

arbitration rule which serves as the final classification outcome. Figure 2.3 demonstrates an arbiter tree generated for k = 4. $Trainset_1$−$Trainset_4$ are the initial four training datasets from which four classifiers ($Classifier_1$−$Classifier_{4)}$ are created simultaneously. $Trainset_{12}$ and $Trainset_{34}$ are the training sets produced by the rule selection from which arbiters are produced. $Arbiter_{12}$ and $Arbiter_{34}$ are the two arbiters. Similarly, the root arbiter ($Train_{14}$ and $Arbiter_{14}$) are generated and the arbiter tree is completed.

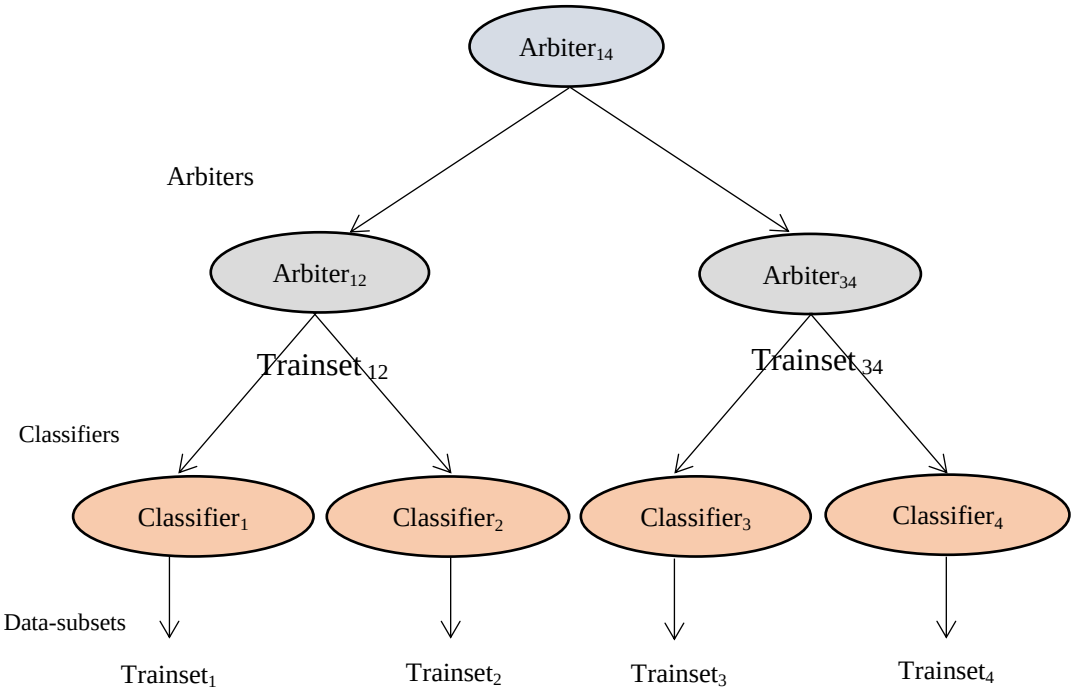


Figure 2.3: Arbiter tree sample

- Combiner Trees are similar to arbiter trees and are also generated through a bottom-up approach. The main difference is that a combiner is placed in each non-leaf node of a combiner tree [178] and the meta-learner's training set is anchored on the classifications of the learned base classifiers. The content of training observations is determined by a

composition rule. To classify an observation, the composition rule is used to generate the classifications. The results from two base classifiers and a single combiner are shown in Figure 2.4. There are two schemas: the stacking schema and stacking with additional input attributes. A study by Chan and Stolfo [179] proved that the stacking schema does not perform as well as the second schema. It was shown that information is lost as a result of data partitioning, but combiner trees preserve a good accuracy level achieved by a single classifier.

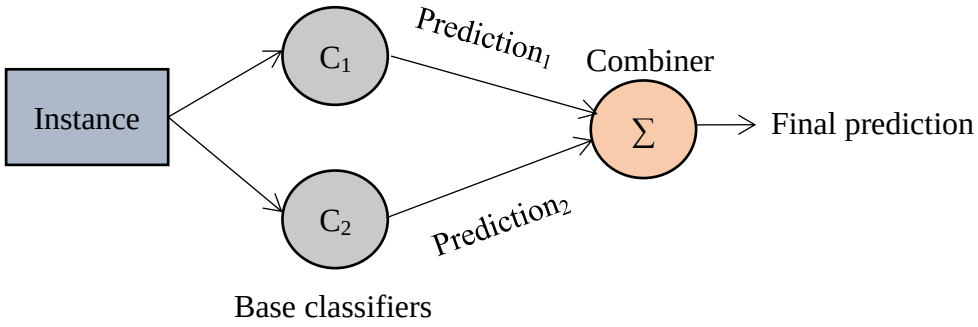


Figure 2.4: Prediction from two single classifiers and a single combiner

- Grading is a technique that employs "graded" classifications as meta-level classes [180]. Graded is used to distinguish correct from incorrect classifications. The classifications made by the *k* different classifiers are transformed into *k* training sets by this method. This is done by making use of the instances k times and then attaching the same to a new binary class in each occurrence. The class shows whether or not the k^{th} classifier produced a correct/incorrect classification. One meta-classifier is learned for each classifier responsible for classifying when the base classifier misclassifies. During classification, each base classifier

predicts the unclassified instance. The classifications are done by the base classifiers which are confirmed correct by the meta-classification schemes from the outcome. Voting is performed in situations where there are conflicting classification results from several base classifiers. To some extent, grading is a generalization of cross-validation selection [181] where the training data is divided into k subsets and classifiers are built on $k - 1$ set and tested on one set. The classifier with the least misclassification rate is returned by the procedure. The strategy considers separately each instance in such a way that only classifiers that predict correctly a particular instance are considered. Grading differs from combiners because grading does not change the instance attributes by replacing them with class predictions. In addition, grading creates several sets of meta-data for each base classifier, and then meta-level classifiers are learned from the sets. On the other hand, arbiters use information about the disagreements of classifiers to determine the training set while grading makes use of disagreement to select the training set.

2.6 Optimization of Ensemble Classifier

The subset of base models with the highest performance can be selected to fit an ensemble. However, this method of selecting a subset of models is proving to several challenges such as over-fitting, sensitivity to noise, and the possibility of selecting similar base models. The performance of ensemble learning depends largely on the base models and so it is important to pay attention to the process of base models' selection. It is important to evaluate the performance of each base model individually and how they contribute to the ensemble model. The diversity of the base models

should be ensured and each should have a reasonable contribution to the overall performance of the ensemble model. Against this backdrop, this research proposes to develop an optimized ensemble classifier algorithm that can search and select the best base models that will result in an ensemble model with high performance. The optimized algorithm will employ the Genetic Algorithm in the search process to ensure that only the base models with high performance are selected for ensemble learning. This approach has several advantages such as curtailing over-fitting and reduction of model complexity. The 25 base classifier models shown in Table 6.3 of Chapter 6 are optimized using the Genetic Algorithm such that different parameter settings and feature subset combinations are used.

2.6.1 Optimized Classifier Selection Criteria

One of the most important steps in the design of an optimized classifier is how to select the appropriate base classifiers from a list of classifiers. There are two types of methods for classifier selection. These include static classifier selection and dynamic classifier selection. Each method is concerned about optimizing classification accuracy as much as possible. The same set of classifiers is used for the prediction of unseen samples in the case of static classifier selection, while in dynamic classifier selection, a set of different classifiers is selected to form the ensemble. Furthermore, the base classifiers are trained on the training data after which the results of the combination from the development data are used to select the subset of classifiers with the optimum performance. For a corpus without development data, k-fold cross-validation is applied to the training set to search for the optimum classifier subset for testing the unseen data. There are several approaches for implementing the dynamic classifier selection method. One of these is to dynamically determine the candidate classifiers based on similarity in the performance on similar input values in the

training data [153]. Another method is to select the single best-performing classifier in the neighborhood of the unseen data.

2.6.2 Optimized Ensemble Classifier using Search Algorithm

It is required to explore all the possible candidate classifier combinations to arrive at the optimum classifier ensemble. There are different methods for performing the search, such as using Single Best Search, N Best Search, Forward Search, Backward Search, Exhaustive Search, and Evolutionary Search algorithms. The Single search algorithm considers the best performing classifier, while the N Best Search algorithm considers the N best performing classifier(s). Furthermore, the Forward and Backward Search algorithms terminate when the optimization function has been exhausted and are referred to as greedy search algorithms. On the other hand, the Exhaustive Search algorithm works on the assumption of a small number of the candidate classifier ensembles, which makes it unlikely for an increment in the number of the base classifiers. The Evolutionary Search works better for a large number of classifiers, unlike the greedy search algorithms. Several studies have implemented this algorithm for the process of classifier selection [182]. Some prominent examples of the Evolutionary Search algorithms include the following: Genetic Algorithm, Bee Colony, Firefly, and Ant Colony. Several studies have implemented the Genetic Algorithm as an Evolutionary Search approach [182,183]. In this thesis, one of our proposed methods for sentiment analysis in Twitter employed the Genetic Algorithm as part of the designed architecture, as discussed in Chapter 6. The Genetic Algorithm is a model that mirrors a natural evolutionary system.

2.6.3 Principle of Genetic Algorithm

Genetic Algorithms have been applied in engineering and science for solving problems involving computational modeling of natural evolutionary systems [184]. It involves a natural selection process in which the fittest individuals are chosen for reproduction to produce offspring(s) for the next generation. It was invented by John Holland in the 1960s and was later developed by Holland and his students and colleagues in the 1960s and 70s at the University of Michigan [184]. They are based primarily on the Darwinian concepts of survival of the fittest [185]. Genetic algorithms are inspired by genetic functions that attempt to find potential solutions to a given problem iteratively. In the process, new populations are reproduced, and the optimal solution is targeted. The main concept in genetic algorithms is population because many objects in the solutions space are searched in parallel sets of genetic operators aside from the other search functions. Genetic algorithms are very effective, especially when optimizing function spaces that calculus-based algorithms cannot handle. They work by the repeated evaluation, selection, and reproduction, and the process continues until the convergence of the population of solutions is optimal. Figure 2.5 shows the steps taken by a genetic algorithm. The phases of the genetic algorithm are described below:

2.6.3.1 Initial Population

The initial population constitutes the first set of potential solutions which are usually generated heuristically or randomly with the population is made a set of individuals. An individual consists of a set of parameters referred to as Genes. A combination of Genes form a chromosome and several chromosomes make up the genetic algorithm represented by binary-encoded values which form candidate solutions to the optimization problem [186]. A candidate solution is normally encoded as an array of

parameter values corresponding to the given problem. Formally, given a problem with *n* dimensions, each chromosome is encoded as an *n*-element array.

$$\text{Chromosome} = [G_1, G_2, \ldots, G_n]$$

where each G_i constitutes an actual value of the i^{th} parameter.

2.6.3.2 Encoding

Each member of the population is represented as a binary bit string with a fixed length in a genetic algorithm. The parameters of the problem are encoded as strings and referred to as a chromosome or DNA and can be decoded to the original values of the parameters. It is usual to refer to the encoding as genotype, while the decoding as phenotype.

2.6.3.3 Evaluation or Fitness Calculation

To test and evaluate the performance of a potential solution, the fitness function is used. The ability of a chromosome to compete with other chromosomes is referred to as its fitness and is assigned a probability of survival to indicate how fit a chromosome is. Evaluation is conducted immediately after creating the initial population to find the fitness level of the constituents of the population. To determine the members that will participate in the next round of selection and reproduction, those with higher levels of fitness are considered. The chromosome or DNA is used in computing the fitness which serves as input to the objective function.

2.6.3.4 Selection

In the selection phase, the fittest chromosomes to be used for reproduction are selected using the criteria defined by the user. The chromosomes transfer their genes from one generation to the other. It is assumed that only those chromosomes which are fit are selected in the previous generation to produce offspring. Selection approaches including roulette wheel selection, tournament selection, rank selection,

and elitism are used to select pairs of parent chromosomes. Part of what takes place at the selection phase is to determine the number of offspring each individual will produce in the next generation. This is referred to as the target sampling rate, and it is not a whole number that must be changed to an integer. Individuals exhibiting higher fitness are preferred over those with less fitness.

2.6.3.5 Reproduction

At this phase, two objects are randomly selected from the mating pool. Following the selection, the genetic functions are then applied to their genetic attributes to produce new members that form the next generation. The process continues till the completion of the next population. Crossover and mutation are the main operators in the recombination phase.

2.6.3.6 Crossover

This is the phase where the selected chromosomes reproduce and pass on their genotype to the next generation. A crossover point is chosen for the pairs of chromosomes by default or randomly among the genes. Parents exchange their genes to produce offspring until a crossover point is attained. When the selected crossover point is attained, the tails of the parent chromosomes are swapped to produce new offspring. The crossover probability (between 0.5 and 1.0) determines how the crossover occurs and is generated by the crossover function.

2.6.3.7 Mutation

This is a secondary genetic operation which task is to maintain diversity in the population. It safeguards the population against pre-mature convergence on one solution in addition to creating genetic codes absent in the current population. The mutation changes the genotype of an individual when applied at a frequency known as the mutation rate which is usually less than 0.05.

2.6.3.8 Accepting

This is the point where the stopping condition is reached after the series of genetic operations. It is the point where an optimal solution is reached and all genetic operations terminate.

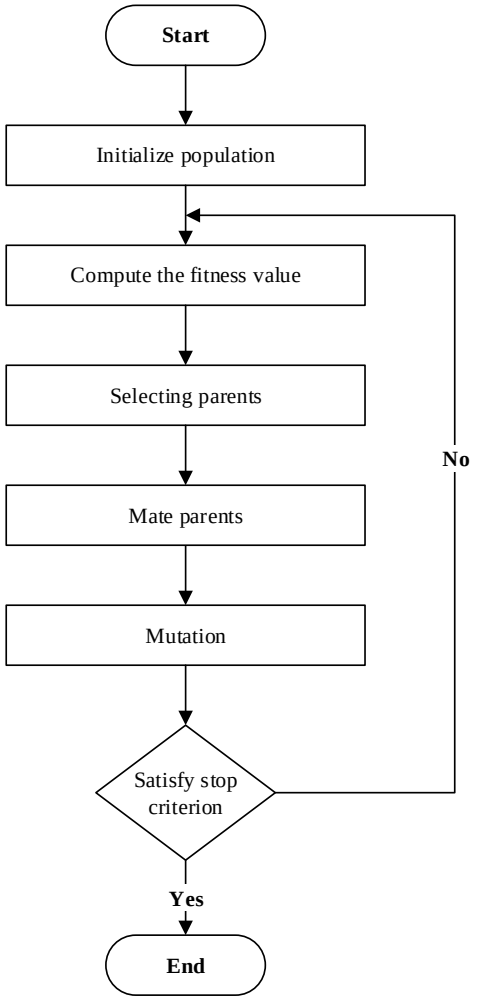


Figure 2.5: Schematic flowchart of the Genetic Algorithm (redrawn from [187])

Chapter 3

RELATED WORK

This chapter provides the prominent related work being carried out in the area of sentiment analysis. First, we discuss the studies which are concerned with sentiment analysis. In the last decade, sentiment analyses have been broadly considered the most active research area in the field of text mining. There are many applications and enhancements on sentiment analysis algorithms that were proposed in the last few years. Researchers used different techniques to extract features and applied various machine learning techniques to perform sentiment analysis. Early works on sentiment analysis mostly tried to analyze if the overall sentiments of a review are positive or negative. First works on sentiment analysis beginning with the seminal work of Pang et al. [50], Turney [7], Dave et al. [188], Yu and Hatzivassiloglou [189], and Pang and Lee [190]. They considered the reviews as Bag of Word and focused on analyzing their sentiment of them as positive or negative using machine learning classifiers. Later works, researchers are now focusing on the direction of improving the sentiment analysis process. This chapter consists of three different sections: Section 3.1 reviews related literature on sentiment analysis using different datasets. Section 3.2 reviews related works regarding SemEval-2017, Task 4, Subtask A, B, and C datasets [17]. Finally, Section 3.3 reviews some relevant research works that were used Stanford Sentiment Treebank (SST-2 and SST-5) datasets [191].

3.1 Sentiment Analysis using Different Datasets

This section reviews different methods and approaches in sentiment analysis. We briefly review some major relevant works that focused on key aspects of our work such as machine learning techniques, ensemble learning techniques, weighted voting approaches in ensemble learning methods, and optimization algorithms in sentiment analysis. In most prior works, machine learning techniques have been extensively used in sentiment analysis. By extension, current works addressing sentiment analysis based on various machine learning approaches have been indicated. Most of the existing approaches solve the problem as a text classification problem. The simplest way has used in sentiment analysis is using the lexicon-based approach [192], which calculates the total number of positive and negative sentiment words appearing in the given document to determine the overall sentiment of the review. The drawback of this method is poor recognition of affect when negation word is appeared in the text [193].

Nowadays, most researchers have applied many supervised methods to sentiment analysis; these methods are mainly based on supervised learning approaches which rely on manually labeled sample data (opinions) with the intelligent design of different set efficient feature engineering to obtain a good classification performance [194]. This idea is to find some informative features to reflect the sentiment expressed in a given document. Meanwhile, some researchers proposed several unsupervised learning methods by utilizing sentiment lexicons [195] containing sentiment words along with their manually assigned polarity. In recent years, the majority of works on sentiment analysis have been employed based on supervised learning approaches. In supervised learning techniques, a set of labeled data is

utilized for training a classifier then the trained classifier is used for classifying unlabeled data. Choosing appropriate features for representing the sentiment words for the classifiers is one of the most significant tasks in this approach. Many methods have been proposed for features that are frequently used in machine learning techniques [32,196,197]. Some major relevant works on sentiment analysis are briefly described below:

Madasu et al. [198] provided sentiment evaluation of different feature selection methods for sentiment analysis performed on Yelp, Amazon, and IMDB reviews datasets. Term Frequency-Inverse Document Frequency was utilized as a feature extraction method for creating feature vocabulary. After that, different feature selection methods are tried to choose the best subset of features from the feature vocabulary. Next, several base learning classifiers SVM, LR, NB, and DT were trained on selected features. Lastly, Random Subspace and Bagging ensemble approaches are applied to classifiers to improve the performance efficiency of sentiment analysis.

Kumar et al. [199] proposed a hybrid feature extraction method for sentiment analysis. The method was performed on the IMDB movie review dataset. In this work, both lexicon and statistical techniques are performed for feature extraction steps. Features from both techniques are combined to form a single feature set. Then it is trained on different supervised learning classifiers such as SVM, NB, Maximum Entropy (MaxEnt), and KNN. The experimental results are highly promising to increase the performance of sentiment analysis in terms of accuracy.

Hassonah et al. [200] introduced a hybrid filter and evolutionary wrapper system to improve sentiment analysis. The system was applied on four different Twitter social

media datasets including public opinion, product, restaurant, and movie reviews. In this system, two different feature selection methods such as Multi-Verse Optimizer (MVO) and Relief were used to extract features. Then, the obtained features from both methods were combined to form a new feature set. After that, they developed classification methods based on using SVM. The results show that their proposed system is most prominent for sentiment analysis tasks in terms of accuracy.

Afzaal et al. [201] proposed a predictive schema for aspect-based sentiment analysis, allowing users to analyze sentiments related to different aspects of tourist reviews. In this system, the semantic relations were used among phrases of review for extracting infrequent and implicit aspects to improve the predictive accuracy of sentiment analysis. The proposed framework was performed on hotel and restaurant reviews. Five commonly used machine learning classifiers, namely SVM, NBM, Fuzzy Lattice Reasoning (FLR), Random Forest Tree (RFT), and MaxEnt are used to build their system. The experimental results demonstrate considerable enhancement in the performance of the proposed framework using the NBM classifier.

Naresh et al. [202] proposed an Optimizing Supervised Machine Learning Approach (SMODT) for sentiment analysis on Airline reviews. In this work, Sequential Minimal Optimization (SMO) algorithm was used to extract the best features from the preprocessed data. After that, different supervised learning classifiers KNN, SVM, and DT have been employed in their schema to classify the updated training data into their classes. The results show that SMO can improve the efficiency of sentiment analysis accuracy when combined with DT.

Recently, most existing approaches have solved and improved the sentiment analysis problem based on the ensemble learning approach to obtain a more robust classification for sentiment analysis. The performance of sentiment analysis tasks can be improved through ensemble learning approaches [203,204]. Kilimci et al. [205] proposed an ensemble learning approach to enhance sentiment analysis. The authors combined ensemble learning algorithms with the word embedding approach to increase the performance of sentiment analysis in short texts by extending feature spaces. In this study, the authors focused on enhancing the feature space to improve the short text classification due to the limited size of expression opinions on social media such as Twitter. The authors conducted experiments using Twitter datasets to show the efficiency of their proposed model. The outcomes demonstrate that the word embedding-based proposed classifier outperforms the traditional ensemble classifiers for sentiment analysis.

Akhtar et al. [206] presented a stacked ensemble approach for the intensity prediction of emotion and sentiment. In this work, the intensity predictive outcomes obtained from classical feature based on Support Vector Regression (SVR) classifier and three different deep learning models, namely Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), and Gated Recurrent Unit (GRU) were combined through using Multi-Layer Perceptron (MLP) model. The proposed model was evaluated on the Emolnt 2017 and SemEval 2017 dataset. The experimental results show that the proposed model highly performs in identifying the intensities of sentiments and emotions in the messages.

Khan et al. [207] proposed a novel ensemble method called EnSWF for sentiment analysis. The proposed EnSWF system can extract appropriate features and reduce

the high dimensionality of features space by selecting the best meaningful and effective subset of features from Amazon Product and Cornell Movie Reviews datasets. POS, Unigram, and Bigram feature sets were used with an ensemble of multiple filter base-features selection and ensemble classifier to generate a robust sentiment classification system in the proposed EnSWF approach. In this work, the simple majority voting method was used in both filter based-features selection and classification. In the phase of features selection, five different filter based-features selection methods: Minimum Redundancy and Maximum Relevance (MRMR), Gini Index (GI), Information Gain (IG), Gain Ratio (GR), and Chi-square (CHI) were used and combined as an ensemble method. At the phase of classification, majority voting with three different classifiers: SVM, Generalized Linear Model (GLM), and NB were used. The authors found that the proposed EnSWF successfully reduces the high dimensionality of feature sets and increases classification accuracy.

Pong-Inwong et al. [208] introduced an ensemble framework method for sentiment analysis in educational teaching. In this work, the student's opinions data toward their instructors in the Network Management subject was used, consisting of 400 samples with 20 attributes. The number of attributes was reduced using the CHI feature selection technique and trained with the voting ensemble method. The results show proposed approach is significantly efficient.

Khalid et al. [209] presented an ensemble classifier for sentiment analysis called GBSVM which performs voting from Gradient Boosting (GB) and SVM classifiers. In this work, Term Frequency and three variants of Term Frequency- Inverse Document Frequency such as Unigram, Bigram, and Trigram are used as features to train the classifiers. The proposed framework was performed on the Google App

dataset. The results reveal that the proposed GBSVM with Unigram performs better than Term Frequency, Bigram, and Trigram.

Saleena [210] proposed a weighted ensemble learning model for sentiment analysis in tweets. The main aim of this work is to improve the performance and accuracy of sentiment analysis in tweets. In this work, base classifiers including NB, SVM, RF, and LR are combined into one classifier, and the Bag of Word feature extraction technique is used to extract features and convert them into vector space. The proposed ensemble classifier has been trained and tested on four English sentiment datasets such as Stanford-Sentiment140 corpus, Health Care Reform (HCR), First GOP debate Twitter sentiment, and Twitter sentiment analysis datasets. The results prove a higher performance of the proposed ensemble model compared to the base models and the majority voting technique.

Yueyang et al. [211] presented an ensemble framework for sentiment analysis based on the weighted voting algorithm. In this work, the features from the NLPCC benchmark dataset were extracted using Syntactic, lexicon, and semantics information. The IG method was used to find the optimal suitable subset of features for training by NBM, SVM, and Conditional Random Field (CRF) classifiers. The proposed ensemble model combined these three classifiers based on weighted voting and simple majority voting algorithms. The experiments prove that weighted voting is more effective in improving sentiment analysis efficiency than simple majority voting.

Araque et al. [212] proposed combining traditional machine learning techniques with deep learning techniques for sentiment analysis through several ensemble learning

models. In which several sentiment models were trained with the combination of different types of feature extraction methods. The proposed model was tested on two different domain datasets: movie review and twitter. The results confirm that the proposed model surpasses the performance of sentiment analysis.

Alrehili et al. [213] presented a voting ensemble method for sentiment analysis to classify customer reviews into positive and negative. The voting model combined five classifiers, including SVM, NB, RF, Boosting, and Bagging. The outcomes demonstrate that the voting algorithm provides the best performance for sentiment analysis.

In recent works, other techniques like computational intelligence are also actively used in this area, including Evolutionary Computing, Rough Sets, Swarm Intelligence, Fuzzy Logic, Neural Networks, etc. Genetic algorithm is probabilistic search techniques belong to the class of evolutionary algorithms. It is majorly used to optimize the solution from the set of feasible solutions. Nowadays, genetic algorithm has been applied to different domains, including signature verification, scheduling, timetable, image processing, robot control, routing, information retrieve, machine learning, etc. [214, 215]. Only a few attempts in the literature have been studied Genetic Algorithm to enhance sentiment analysis problem researches. The recent studies in sentiment analysis have been applied the Genetic Algorithm to feature selection or combined this algorithm with existing machine learning techniques to achieve better classification accuracy. For instance, Ishaq et al. [216] presented an efficient classification framework for sentiment analysis using CNN and Genetic Algorithm. In this work, three different operations have been combined: First, semantic features from movie, automobiles, and hotel reviews datasets have been

extracted and then transforming to vector space by using word2vec. Next, CNN was used to extract opinions. Lastly, the parameters of CNN have been tuned with a Genetic Algorithm to obtain optimum values. The experimental results reveal that the proposed approval provides better results for sentiment analysis.

Cahya et al. [217] proposed a feature-weighted method to enhance the Complement Naïve Bayes (CNB) classifier based on the Genetic Algorithm to analyze sentiment in tweets. In this work, term frequency weighting was used to extract features from preprocessed data to produce a document term matrix then fed it to the Genetic Algorithm to select the optimum combination of feature weights based on the correlation between features and class labels. After assigning weights to features, the NB classifier was developed using a training set and produced weights. Finally, the testing set is classified to identify the sentiment in the tweets., Experiments were conducted on the twitter airline dataset in order to validate the proposed method. The outcomes show that the proposed model improves the sentiment classification ability.

Iqbal et al. [218] developed a hybrid framework for sentiment analysis by combining machine learning classifiers with lexicon-based approaches to classify reviews datasets from the UCI repository. A new genetic algorithm was proposed to reduce the feature set size by developing a modified fitness function in this framework. SentiWordNet dictionary was used in the fitness function to compute the polarity difference between the feature vector and class label. The experiment results show that the hybrid proposed framework improves efficiency and sustaining the scalability of sentiment analysis.

Fatyanosa et al. [219] employed the Genetic Algorithm as a feature selection process to reduce the features in the sentiment analysis. The experiment was conducted using a Twitter 5-point scale dataset related to self-driving cars. In this work, NB was trained then used to classify testing data. The result of the F_1-score was used as a fitness function for each population in each generation. The results demonstrate that the combining algorithms with genetic algorithm improve the ability of classifiers and recognition of minority the classes significantly.

Keshavarz et al. [220] introduced a model named Adaptive lexicon learning by genetic algorithm (ALGA) to classify the polarity of sentiment based on genetic algorithm. In this work, a Genetic Algorithm was incorporated to create lexicons but the calculation of fitness in this method was time-consuming. The authors proposed a novel parallel approach for calculating the fitness of ALGA efficiently on Healthcare Reform (HCR), Obama McCain Debate (OMD), Sanders–Twitter Sentiment Corpus, SemEval datasets. The results demonstrate that the ALGA model achieves better performance in terms of time, speed and complexity.

Saidani et al. [221] presented a weighted genetic algorithm to optimize the process of feature selection in analyzing sentiment in tweets. The authors combined a supervised weighting method with a stochastic search method to generate a feature subset that can select and extract the most efficient features. The outcomes reveal the efficiency of their proposed model.

3.2 Sentiment Analysis using SemEval-2017 Task 4 A, B, and C Datasets

Freely existing datasets permit evaluation of the proposed methods in all fields of text mining. SemEval organization provides a gold annotated sentiment analysis dataset based on Twitter for researchers working in this area. Specifically, several research studies have been conducted on sentiment analysis using SemEval-2017 on sentiment analysis Twitter dataset which contains tweets annotated for the sentiment on 2 point, 3 point, and 5-point scales. In this subsection, we review some relevant works on sentiment analysis related to the use of SemEval-2017, Task 4 for sentiment analysis in the Twitter dataset.

Baziotis et al. [222] introduced two deep learning models for sentiment analysis. LSTM classifier was used and augmented with 2-types of attention mechanisms. In the first model, Bidirectional LSTM was used and was equipped with an attention mechanism to address the message level sentiment analysis problem. In the second model, Siamese LSTM has utilized a context-aware attention mechanism to address the topic-based sentiment analysis problem.

Cliche [223] presented an ensemble system with deep learning techniques for sentiment analysis. In this system, 10 CNN and 10 LSTM with different parameters and different pre-training schemes were combined. The main aim of this work is to experiment with deep learning classifiers to generate the best sentiment classifier system to classify the polarity of sentiment in tweets.

Kolovou et al. [224] introduced the combination of several classification systems for sentiment analysis. The systems compute vectors with different semantic and statistical features: Word2vec, Webis, etc., then train them using different classifiers: NB and CNN, and take the average result. The proposed system's object is to experiment with combining different mathematical and linguistic methods to improve the performance of sentiment classification in the tweets.

Symeonidis et al. [26] proposed an ensemble sentiment classification scheme based on the majority voting algorithm. The voting schema combines supervised machine learning classifiers: Passive-Aggressive, SGD, and SVC with other linguistic features like sentiment lexicon and Bag of Word to select and identify an optimum subset of base learning classifiers to classify sentiments in the tweets.

Onyibe et al. [225] proposed a system to predict the sentiment of the tweets based on using optimized Conditional Random Fields (CRF++) and lexical features. First, seven lexical features: Unigrams, Tweet length, Tweet length binned, Bigrams, SentiStrength, Removed URL and Stopwords were combined to identify the optimal combination of features. They explore that the combination of unigram and SentiStrength features with tune CRF++ parameters provide the best performing result.

Zhang et al. [226] proposed a multi-channel CNN-LSTM model, which comprises the combination of multi-channel CNN and LSTM for classifying sentiments in Twitter. The authors used a multi-channel strategy in the CNN layer where several filters of different lengths are adopted to extract n-gram features in different scales, and then these features have been composed sequentially by applying LSTM.

González et al. [227] implemented a model based on deep learning approaches to address the classification of sentiment tasks. They utilized 3 Convolutional Recurrent Neural Network (CRNN) and the combination of specific and general word embedding with sentiment dictionaries for high-level abstraction learning from representations that have some noise. The model has three inputs in- / out-domain embeddings and sequences of word polarities. The outputs of these three network models were combined and fed to a fully-Connected Multi-Layer Perceptron (MLP). The outcomes of the proposed method are very promising.

Lozić et al. [228] introduced a model for sentiment analysis based on using the SVM classifier with a linear kernel. In this schema, a set of basic features including word embedding, Term Frequency-Inverse Document Frequency, counting features, user information, sentiment polarity lexicons, and more specific features including nostalgia features, rating features, and recent deaths were used classify the sentiment of tweets. The classification efficiency of their model was proven by the results of the experiments.

Gupta et al. [229] introduced a system based on the detection of sarcasm to improve sentiment classification tasks. In this work, a feature set is proposed named an Affect Cognition Sociolinguistics (ACSs) feature and trained with the SVM classifier to detect sarcastic in tweets. A two-level cascade classification system has been developed for sentiment prediction and observed that sarcasm detection derived features consistently benefited key sentiment analysis.

Rozental et al. [230] presented two supervised training methods to perform sentiment analysis on Twitter data; the first was based on RNN architecture, and the other used

LR, NB, and Feed-forward Neural Network. The authors produced Parse Tree for each sentence of the document and gave 5-label prediction results. Then, they extracted a sample of 20000 tweets from Twitter randomly, where each tweet of the Parse Tree was labeled for its sentiment.

Wang et al. [231] used a simple CNN to perform sentiment analysis toward sentence and topic levels. They used six layers in the CNN architecture: convolutional, topic embedding, input, output, max pooling, and concatenate layers.

Rajendram et al. [232] presented a Gaussian Process model to classify sentiment in tweets. They used the Bag of Word feature extraction method with fixed rule Multi-Kernel learner to develop the Gaussian Process model. The experiments show that Multi-Kernels are more effective compared to Single-Kernel in sentiment classification.

Li et al. [233] developed a model for sentiment analysis using Word Embeddings (WE) to learn features from general tweets, Sentiment Specific Word Embeddings (SSWE) to learn features from a distance supervised tweets, and a Weighted Text feature Model (WTM). In this system model, they combined WE, SSWE, and (WTM). The WTM produces two feature sets: the first set is the negation feature which counts the number of negation sentiments in the tweets without utilizing a sentiment lexicon. The second set is generated using Cosine similarity and Term Frequency-Inverse Document Frequency model to compute the similarity between the tweet and each of the polarity types represented by Pseudo Centroid tweets learned from the train set. Then it is fed to the classification algorithm.

Dovdon et al. [234] proposed a framework for Twitter sentiment classification based on the supervised machine learning technique. They incorporated MaxEnt with different feature sets, including Bag of Word, Bigram, Punctuation based features, Lemmas, and Lexicon based features to classify the overall massage polarity and topic-based message polarity of tweets.

Laskari and Sanampudi [235] implemented a simple Word2Vec feature extraction method with a Gradient Boost Tree ensemble classifier to classify the polarity of sentiment in tweets. In this work, they applied the Gradient Boost Tree ensemble classifier with a parameter optimization method to enhance sentiment classification accuracy.

3.3 Sentiment Analysis using Stanford Sentiment Treebank Datasets

A variety of approaches have been proposed to sentiment analysis tasks that use Stanford Sentiment Treebank (SST-2 and SST-5) datasets. Some of these approaches use predefined lexicon and traditional machine learning approaches that consider sentiment analysis as a kind of classification. Some of the others are based on deep learning approaches. This subsection reports some of the important work that used machine learning techniques and deep learning techniques to classify tweets' sentiments. In recent years, many attempts have been made to improve the performance of supervised learning classifiers and deep learning classifiers in analyzing sentiments in different ways. For instance, a pioneering work proposed by Lei et al. [236] introduced a model named Sentiment Aware Attention Network (SAAN) based on using polarity lexicons to improve the attention mechanism in neural network sentiment classification. Firstly, they identified sentiment words in all sentences based on using sentiment polarity dictionaries. After, they used the LSTM

model to learn the feature vector to every sentence based on its sentiment words. This feature vector was concatenated with the hidden feature vectors which were learned from another LSTM for improving the attention weights calculation. However, the sentiment word polarities in this method are neither considered nor exploited.

Yu et al. [237] proposed a sentiment word embeddings model for improving the sentiment polarity detection by refining the existing pre-trained word embeddings using the sentiment intensity scores from sentiment lexicons. This method improves word vectors such they are closer in both sentimentally and semantically similar words in the lexicon. The results prove that the proposed refinement model enhances traditional word embeddings and existing proposed word embeddings for Binary, Ternary, and Fine-grained sentiment analysis performance.

Lu et al. [238] presented a new method for sentiment analysis that integrates sentiment lexicon with attention-based bidirectional long short-term memory (BiLSTM).

Sadr et al. [239] combined both Convolutional and Recursive Neural Networks with pre-trained Word Vector to propose a new robust sentiment analysis classifier that consists of four layers: Convolutional, Recursive, Embedding and Classification layers. This work aims to use the RNN as an additional pool layer to reduce the loss of local data and capture long-term dependency. Furthermore, they pointed out that the CNN structure itself is the chief reason for the network to extract Multi-level and Multi-scale features.

Chen et al. [240] proposed a new method to enhance the performance of sentences level sentiment analysis by performing the machine learning approach with a deep neural network model. First, they employed Bidirectional LSTM-Conditional Random Fields (BiLSTM-CRF) to extract target expression in the sentences, after classified them into different types of sentences according to the number of targets extracted from them and then fed each group of sentences to 1-dimensional CNN separately to speed up the process of sentiment classification.

Baktha et al. [241] investigated the performance of three variant RNNs in predicting the sentiment of reviews namely GRU, vanilla RNNs, and LSTM. First, pre-trained word vectors were fed as input to the structure of RNN to analyze three hidden layers. Then, the results of RNN were fed to a dense layer that predicts the output. The outcomes depict that GRU achieved the highest accuracy in sentiment classification.

Kim et al. [125] designed a CNN model for textual sentiment analysis. This model consists of different layers to classify sentiments over the review of the text. In this model, the authors first initialized an embedding layer in which words semantically reside in space and finalized through the process of training. After, they utilized 2 consecutive convolutional layers, one of them used to store local information, and the other one used to get features from contextual words from the first layer. Next, they used the max-pooling layer to get obvious features. Lastly, they calculated the value of probability for each class with a fully connected layer and soft-max activation function. The conclusion depicts that the successive convolutional layers can provide superior performance on the classification of long text.

Hiyama et al. [242] introduced a neural sentiment classification model based on using an attention mechanism. This method consists of four layers. First, they used a word embedding layer for getting word vectors from each word of sentences. Then, they accepted word vectors as input and used the Bidirectional LSTM layer to generate a new word vector considering surrounding words. Next, they used the attention layer to estimate the importance of word vectors which are strongly related to the sentiment polarity of the sentences, and build sentence vectors based on their importance. Lastly, they used the classification layer which uses the sentence vectors to predict the sentiment polarity. The experimental results reveal using the attention mechanism in neural sentiment analysis works well and achieves higher performance compared with using neural sentiment analysis without the attention mechanism.

Li et al. [243] presented a sentiment analysis model based on deep learning techniques by combining 2-Channel CNN-LSTM with CNN-BiLSTM in a parallel manner. The outcomes indicate the superiority of the proposed model in predicting the sentiment polarity of review text.

Hassan et al. [244] proposed a neural sentiment model named “ConLstm” by combining CNN and LSTM models and used pre-trained word vectors to represent the review sentences. In this framework, they utilized the LSTM model as a pooling layer to support convolutional layers for capturing Long-Term dependencies in the sequence of sentences more efficiently. The results demonstrate that the proposed method achieves good performances with fewer parameters on sentiment analysis tasks.

Dong et al. [245] proposed a Capsule Network framework named “caps-BiLSTM” based on using the BiLSTM model for sentiment analysis. First, caps-BiLSTM used a convolution layer to convert the word vectors to hide vectors. Then, entered these vectors were into Capsule Network to find similarities between inputs and outputs. Lastly, the resultant vectors of the capsule were entered into the BiLSTM model to predict the sentiment labels of the text. Experimental outputs reveal that caps-BiLSTM has a favorable performance in the sentiment analysis.

Chapter 4

EXPERIMENTAL SETTINGS

To improve on the performance of the tasks in sentiment analysis, it is important to choose a particular classification algorithm, fine-tune the parameters, and select the most important features required to achieve the most desired performance. Usually, there are a large number of classification algorithms and datasets normally come with a large number of features. This makes it difficult to decide on the algorithm and feature subset to use. A possible solution to this challenge is to design an ensemble classifier and then optimize it such that good performance can be achieved. With this approach, all or a subset of the individual classifiers work together to classify an input. Literature evidence shows that optimized ensemble classifier or classifier ensemble methods exhibit a better accuracy than the individual members that make up the ensemble [246,247]. The experimental setup used for our proposed methods is discussed in this Chapter. The benchmark datasets utilized for the main experiments to evaluate the performances comparatively are described in Section 3.1. Section 3.2 discusses the stages of text preprocessing while Section 3.3 discusses feature extraction. Furthermore, Section 3.4 is about the machine learning classifier approaches deployed for sentiment classification in the proposed methods. Finally, Section 3.5 discusses the approaches used for joining the decision of individual classifiers to form the proposed ensemble method. The evaluation metrics used to evaluate the performances of proposed methods are explained in the next section.

4.1 Sentiment Datasets

The datasets used for conducting experiments to evaluate the effectiveness of the proposed methods are presented in this section. The datasets include the following:

4.1.1 SemEval 2017 Task 4 (Sentiment Analysis in Twitter)

The datasets, SemEval 2017 is accessible for research purposes regarding Twitter sentiment analysis. The dataset, SemEval-2017 task 4, was used in this study and it has 5 subtasks, A to E Subtasks "A", "B", and "C" are related to the sentiment classification task and subtasks "D" and "E" are related to sentiment quantification [17]. The dataset is made up of user's annotated tweets containing sentiment labels in multiple scales relating to the topic of reference in the message. In our work, task 4 subtasks "A", "B" and "C" are used to train and test the proposed model. It should be noted that subtask "A" contains 20632 tweets with a 3-point scale for classifying a message in positive, negative, or neutral classes. On the other hand, subtask "B" contains 10551 tweets, with a 2-point scale for classifying a message in either the positive or negative class according to the topic. Furthermore, subtask "C" contains 20632 tweets which classify the message on a 5-point scale, namely strongly positive, weakly positive, neutral, weakly negative, and strongly negative, according to the topic. Table 4.1 gives a brief statistical summary of the datasets.

4.1.2 Stanford Sentiment Treebank (SST)

The Rotten Tomatoes dataset is made up of movie reviews that were extracted from the original Rotten Tomatoes page files. Several studies have made use of this dataset. The SST dataset [191] which consists of 11855 sentences extracted from the Rotten Tomatoes dataset was annotated by Stanford University. Each sentence in the dataset was parsed into multiple phrases using Stanford Parser, resulting in 215154

single phrases and annotated based on their sentiment. There are two categories of the SST dataset including:

- SST-2 has binary labeled categories excluding the neutral category, and positive and very positive classes are merged to form the positive category. Similarly, negative and very negative categories are merged to form the negative category. The dataset has9613 sentences, divided into train, development, and test sets with 8544, 1101, and 2210 sentences, respectively.
- SST-5 is a fine-grained labeled category containing 11,855 sentences divided into train, development, and test sets with8544, 1101, and 2210 sentences, respectively. Each of the sets is annotated with fine-grained labels such as very positive, positive, neutral, negative, very negative.

Table 4.1 gives the detailed statistics about SST-2 and SST-5 datasets.

4.1.3 Yelp Challenge Dataset

Our study makes use of the diverse dataset known as the Yelp Challenge dataset. It consists of business, review, user, and check-in data as separate JSON objects. A business object provides information about the type of business, location, rating, categories, business name, and a unique id [248]. A review object consists of review text and a rating associated with a specific business id and user id. Different businesses are described in this dataset such as restaurants, shopping, hotels, and travel, etc. For this study, the restaurant reviews domain is considered from the Yelp Challenge dataset. The dataset is labeled by considering 1 or 2 stars as negative sentiment while 4 or 5 stars as positive polarity. Furthermore, the neutral polarity is not considered in the scope of our study. A total of 5,000 reviews are randomly

selected for each class label to evaluate the proposed methods. The breakdown of the dataset used in this study is given in Table 4.1.

4.1.4 Movie Review (Sentiment Polarity Version 2.0) Dataset[2]

The sentiment polarity dataset of movie reviews constructed by Pang and Lee was used for experiments to evaluate our proposed method. The dataset comprises of movie reviews by users through tweets having binary sentiment polarity labels, positive or negative. Further, the dataset serves as a benchmark for sentiment classification. It has 32937 positive and 31783 negative document reviews, and each is split into sentences having lowercase normalization. Table 4.1 gives the statistics about Movie Review (Sentiment Polarity Version 2.0) datasets.

4.1.5 Stanford Sentiment Gold Standard (STS-Gold)

This dataset was introduced by Saif et al. [249] and was collected to complement the processes of Twitter sentiment analysis evaluations. The dataset was constructed by Saif et al. [249] from 180K tweets from the original Stanford Twitter corpus. The dataset has 2,034 tweets, made up of 632 positive and 1402 negative as presented in Table 4.1. Furthermore, 58 entities were manually annotated by three different human evaluators with the aid of an instructed booklet. To control for noise, the entities and tweets selected to form the dataset were jointly agreed on by the three human evaluators in terms of the sentiment labels. Similarly, targeted entities and polarities were used to interpret the dataset.

4.1.6 Sentiment Labeled Sentences (SLS) Dataset

This dataset is made up of the following files (amazon_cells_labelled.txt, imdb_labelled.txt, yelp_labelled.txt). The amazon_cells_labelled dataset consists of reviews and scores for products sold and was retrieved from amazon.com in the cell

[2] https://www.kaggle.com/nltkdata/movie-review

phones and accessories category. The dataset forms part of the dataset collected by McAuley and Leskovec [250] for benchmarking sentiment analysis. It consists of 1000 reviews, each of which has a binary sentiment label: positive or negative. There are all 500 positive sentences and 500 negative sentences. The statistical summary of the dataset is presented in Table 4.1.

Table 4.1: Detailed summaries of the datasets related to sentiment analysis

Datasets	Class	SP	P	Neu	N	SN	Total
SemEval-2017 4A	3	-	7059	10342	3231	-	20632
SemEval-2017 4B	2	-	8212	-	2339	-	10551
SemEval-2017 4C	5	382	7830	10081	2201	138	20632
SST-2	2	-	4963	-	4649	-	9612
SST-5	5	1852	3111	2242	3140	1510	11855
Yelp Challenge	2	-	5000	-	5000	-	10000
Movie Review	2	-	32937	-	31783	-	64720
STS-Gold	2	-	632	-	1402	-	2034
SLS (Amazon)	2	-	500	-	500	-	1000

[*SP: Number of total strongly positive tweets in the data set, P: Number of total positive tweets in the data set, Neu: Number of total neutral tweets in the data set, N: Number of tweets belonging to negative tweets in the datasets, SN: Number of tweets belonging to strongly negative tweets in the datasets.*]

4.2 Data Preprocessing Methods

Data processing is carried out to clean and normalize the text such that irrelevant text is removed. This process converts the input text (document) to a different output format. Data preprocessing consists of operations that are simple and rule-based. Apart from removing some features during preprocessing, there are some instances when some features can be added to make documents richer. Data preprocessing operations make use of linguistic algorithms as well as external models or datasets. Preprocessing steps are used done in a chain of operations a simple input-output function. Steps can easily be inserted or removed during the implementation of data

preprocessing operations. It is required that steps that have to do with removing features (simplifying, normalizing) are executed before the steps meant to add features. The following sections describe the preprocessing procedures employed in this study and how they impact on text classification accuracy of the proposed methods as discussed in Chapters 5 and 6, respectively.

4.2.1 Normalization

All the common regular expression-based operations are included in this step. There are several sub-steps involved including lowercase conversion, number removal, punctuation mark removal, white space removal, all websites and targets in tweets were changed to placeholders "URL" and "@" respectively, abbreviation expansion, word replacement, and reverting words that contain repeated letters to their original form [251]. These steps render the documents ready for tokenization by removing noise that could negatively affect predictive accuracy.

4.2.2 Tokenization

In this process, the document is split into words, and a list of words is returned. Some examples of the tasks that are handled during the tokenization step are given below:

- What to do with hyphens? Is *"very-elegant watch"* two tokens or three?
- There are entities, which should be one token, but using simple rules they might be split up. IP numbers, car model names, phone numbers... Entity recognition is always a domain-related problem.
- This also can be a language-specific problem. For example, the German language uses a lot of compound nouns, such as

"Rechtsschutzversicherungsgesellschaften" This can be solved through stemming.

It is possible to configure advanced tokenization algorithms; as a result, a use-case is needed when implementing the tokenization step.

4.2.3 Removal Stopwords

It is important to filter out stopwords to reduce noise and render the documents more specific. A stopword does not convey much meaning and is usually domain-specific and language-specific. It is a good strategy to filter out stopwords when carrying out classification tasks with statistical methods. Nevertheless, in some cases, it is not appropriate to drop stopwords. For example, if stopwords are filtered from the sentence "The movie was not good at all.", it will result in "movie good". The latter has altered the meaning in the sentiment analysis perspective. In this study, a list of stopwords with excluding sentiment words such as "against", "love", "like", "happy", "not", etc. are created. Appendix B consists of the list of stopwords.

4.2.4 Stemming (Lemmatization)

The stemming step removes every word affix and retains only the very root of the word. Lemmatization is a similar approach, where words are reduced to the dictionary form known as a lemma. There are two important points to note:

- The first point is that this process is not very important in the English language because there are relatively small agglutinative or conjugation tendencies in the English language. Consequently, the dictionary form is very often used as it is. In languages having enormous word modification tendencies such as Slavic or Latin, this step is important. Some researches were conducted to note separately the effect of

stemming on English and non-English documents, including Indonesian [252] or Arabic [253].

- The second point is that lemmatizing two different words are capable of reducing them to the same base form. The solution is to reduce the high dimensionality of the feature space in text classification [253]. This can help in some situations but also can be problematic in others especially when different forms convey important features for the document classification.

4.3 Feature Extraction Methods

Several disciplines, including machine learning, pattern recognition, and data mining make use of feature extraction [255]. In the process of feature extraction, a subset of features is extracted from the complete set using functional mapping [256]. The process generates all the possible combinations of features to find an optimum subset that can produce more accurate results [257]. Starting with an initial set of data, the process of feature extraction derives a subset of features that are more informative and discriminative, which are then used for learning and generalization tasks. The features extracted for the modeling task are usually those containing relevant information from the input data, which can enhance the accuracy of the targeted task. Feature extraction reduces computational complexity, dimensionality, and overfitting [256]. When a classification model is only able to correctly classify data points that are very closely related to the training data and cannot classify other data correctly, it is said to overfit [258]. Feature extraction will be performed in this study to select features that will be used with the different classifiers that form the ensemble learning model. In this study, single features are extracted as well as combinations of features. The following subsections describe the most important feature extraction

methods relevant to the experiments conducted in our proposed methods, which will be presented in Chapter 5 and Chapter 6 of this thesis.

4.3.1 Bag of Words (BoW)

The BoW is used in NLP and data mining for addressing the unstructured nature of sentences that make up paragraphs in a document. A bag is referred to as a set that allows repetition among its constituent members. The concept of the BoW enables the words that make up a document or a text to be represented in an unordered manner. In this way, the bag of words model does not consider the grammatical structure, semantic meaning, and word order of the document. The number of times each word occurs is of high significance. The BoW model forms a list of its vocabulary from the documents from which the total number of times each word occurs in that document is noted. An illustration of the concept of BoW is given below. Consider the following two text documents:

$Document_1$ = "John likes to watch football. His sister likes football too."

$Document_2$ = "John also likes to watch movies."

Based on the above documents, a bag of words model will create a list of vocabulary as follows:

{" John", "likes", "to", "watch", "football", "his", "also", "movies", "games", "sister", "too", "romantic"}.

The BoW model has been used extensively as a technique for feature generation in text classification tasks. How frequent a word occurs (term frequency) in a given document is of paramount importance in the process of creating features with the bag of words model. From the examples given above, the number of times each word appears can be shown in a feature vector for $Document_1$ and $Document_2$ as follows:

[1, 2, 1, 1, 2, 1, 0, 0, 0, 1, 1, 0] representing $Document_1$

[1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0] representing$Document_2$

Each number in the list represents the number of times the corresponding word from the vocabulary list occurs in the first sentence. It could be observed that words such as “likes,” “football” appears two times and the word “His”, “too”, etc. only once in the first sentence. However, other words such as “also” and “movies” from the vocabulary list are missing in the sentence.

4.3.2 Term Frequency and Invert Document Frequency (TF-IDF)

The TF-IDF statistic assigns weights to terms by combining the frequency of a term is in a document (TF) with how rare the term appears in the entire document set (IDF) [259]. TF-IDF is calculated as:

$$TF - IDF\ (d,t) = TF(d,t)\ \times IDF(t) \tag{4.1}$$

Where d represents a document, t represents a term, TF is the term frequency and IDF is the inverse document frequency.

TF is the number of occurrences of a term (feature) in a document and is calculated as:

$$TF(d,t) = \sum_{i\ \in d}^{|d|} 1\{d_i\ = t\} \tag{4.2}$$

Document Frequency (DF) keeps track of the number of documents that contain a particular term. IDF [63] evaluates the importance of terms concerning the total number of documents and the number of documents containing that term. IDF is an improvement over DF since the latter is not a good discriminator. IDF is calculated as:

$$IDF(t) = \log \frac{1 + |d|}{|d_t|} \tag{4.3}$$

where d is the total number of documents and d_t is the number of documents containing the term t . The relevance of a term in a document is determined by the TF-IDF weight which is assigned to each unique term. The terms are ranked from the highest to the lowest according to the weight value. A threshold k is defined for selecting the top k terms.

4.3.3 Term Presence and Frequency

Another method for weighting features is the term presence and frequency. This method represents a piece of text as a feature vector where each entry corresponds to individual terms. The presence of a term is represented by binary values, where 1 represents the presence and 0 represents the absence. Where a term (feature) occurs in the document or sentence, it is assigned the weight value 1; else the value 0 is assigned. The formula for the Terms presence and frequency is given in 4.4.

$$TF(t) = 1 + log\big(f(t,d)\big) \tag{4.4}$$

where, $f(t,d)$ is the count of term t in given document d.

A study by Pang et al. [6] achieved a higher accuracy using presences as features values than with frequencies. According to the study, term presence is more important to sentiment analysis than term frequency. The presence of a strong sentiment-bearing word is capable of changing the overall polarity of a sentence in sentiment analysis, unlike in text classification tasks. A previous study has shown that the occurrence of rare words is more informative than frequently occurring words. This is referred to as “hapax legomenon.” Similarly, Paltoglou et al. [260]

found that it is more beneficial to use binary features than raw term frequency (TF); however, scaled TF values proved to be effective as binary values.

4.3.4 *n*-gram Features

Several studies have deployed the *n*-gram model in data mining in areas such as language modeling, information retrieval, information filtering, and information extraction. The *n*-gram is a sequence of words or characters generated from a document when a window of size *n* is moved [261]. The *n*-gram is mostly used for text representation of the tokenization in the feature selection technique in sentiment classification. It is the process of breaking down a piece of text into different segments, with *n* indicating the number of words contained in one segment. Unigram keeps only one word per sentence, and at times, this cannot keep track of the significant emotion indicators within the text. On the other hand, the bigrams are capable of intercepting the negations within the given text because it considers two words as one unit. For example, "I am not happy with the flight", usually unigram will take 'not' and 'happy' into consideration separately where bigrams can capture the term 'not happy' that satisfies the negative expression in the original orientation of the sentiment. The positions of the term are very vital in a document representation in sentiment analysis. Consequently, it is important to choose an appropriate *n*-gram model for the sentiment classification task.

The *n*-grams are very beneficial to capture some dependencies between the words and the importance of each phrase in a sentiment. The trigrams take three words into account to constitute an attribute; four-grams take four words as one unit, etc. Usually, *n*-grams adopt n number of words from the text, which serve as one entity for classification considerations. While it is true that higher *n* performs better, it is

also true that when n is high, the level of detail within a text is decreased. For this reason, the choice of n-gram tokenization should be taken care of with caution. In a study by Dave et al. [49], it was established that in some situations, bigrams and trigrams perform better. The choice of n-grams is specific to the type of problem at hand. The disadvantage of using *n*-grams is that more features are created which are capable of affecting performance. In the experiments conducted in our study, *n*-gram length was set to 2; that is, bigrams were used for feature vectors.

4.3.5 Part-of-Speech Tags (PoS tags) Feature

PoS information has been widely used in the literature for sentiment analysis tasks. PoS tags feature also referred to as lexical tags or morphological classes, assigns the parts of speech to each word, including nouns, verbs, adjectives, adverbs, etc. The process gives a breakdown of the structure of the document and information regarding the words and the neighboring words. When the PoS tagging system is used, the ambiguity of the word is decreased [262]. Annotating a word with its PoS tag helps to increase the confidence of the NLP system. The advantage of this concept is that the correct meaning of words in morphological languages such as English is easily determined. It is shown that some adjectives (e.g. lovely, awful), nouns (e.g. concern, hope), verbs (e.g. love, hate), and adverbs (e.g. gently, harshly) convey sentiment. A related study by Turney [7] used the PoS tags feature for adjectives and adverbs to obtain the sentiment orientation at the document level. Some authors believe that the addition of PoS information about the words can improve the performance of classifiers. Some studies have employed PoS tags the syntactic function of a word, as features of state-of-the-art sentiment analysis [46,50,263-265]. In this thesis, Penn Treebank PoS Tags [266] is used to extract

features as described in Appendix A. Table 4.2 below shows the annotated words of "The food was pretty good" with its POS tags.

Table 4.2: An example of PoS tag features

Tokens	PoS tagged sentence
['The', 'food', 'was', 'pretty', 'good']	[('The', 'DT'), ('food', 'NN'), ('was', 'VBD'), ('pretty', 'RB'), ('good', 'JJ')]

4.3.6 Sentiment Lexicon Features

A sentiment lexicon has been widely used for sentiment analysis in many languages [267]. Sentiment lexicons are very vital for both lexicon-based and machine-based learning approaches [268]. Some researchers have leveraged sentiment lexicons to produce unsupervised sentiment models, while some have deployed them to train machine learning algorithms in supervised approaches [269]. A sentiment lexicon collects all words (also known as polar or opinion words) associated with their positive or negative sentiment orientation [267]. Words such as wonderful, beautiful, and amazing are positive sentiment words. On the other hand, words such as awful, poor, and bad are negative sentiment words. It turns out that only a few sentiment lexicons are available and accessible on the Web [270]. Some sentiment lexicons are contained in a single file with a list of words and their associations with negative or positive sentiments. The file has two columns, with the first column containing the words (or terms) while the second column indicates the polarity which can be in the form (positive, negative), (0, 1), or $(1, -1)$. In some situations, the word strength is also included. In another format, sentiment lexicons are divided into two individual files, where one contains positive words and the other contains negative words. Some

of the ways of representing sentiment orientation (polarity value) are given as follows:

- A real value which represents the strength of the sentiment in the range $(-1,+1)$
- Static categories of positive or negative
- Some sets of ranking consisting of strongly positive, positive, neutral, negative, and strongly negative.
- In some situations, sentiment lexicons provide the PoS related to each word, while others give information relating to the strength of the polarity.

The AFINN lexicon is used in this study to assign words with a score ranging from -5 and +5 (most negative up to most positive). A negative score represents negative sentiment while a positive score represents positive sentiment. The total score each for the "positive" and "negative" opinion words are summed for each sentence and the total score represents the overall sentiment of the text. In situations where the total score is positive, then the sentiment of the text is positive otherwise negative. For example,

"I do not like reading all of the negative tweets"

Sentiment words	**Scores**
Not	-5
Like	+4
Negative	-3

The sentiment word (*not) which appears* before (*like)* leads to a negative score instead of a positive score. Aggregating all the positive and negative scores (-5 + (+4) +(-3)) produces a total sentiment score of -4 for the text.

4.4 Base Classifier Algorithms

After the transformation of the text reviews into vectors of number, they need to be processed using different machine learning techniques to obtain the classification result. In this thesis, the most efficient and frequently employed seven classifiers in sentiment analysis have been used to classify the sentiment review datasets as discussed in Section 3.1. The details of these base classifiers are explained as follows:

4.4.1 Support Vector Machine (SVM) Classifier

SVM classifier [270] is among the most commonly applied machine learning classifiers in sentiment analysis. The main idea behind the SVM classifier is to use a set of Hyper-planes to separate different classes. At the training stage, SVM tries to find the best Hyper-planes by maximizing the distance from the closest data point of each class to the Hyper-plane, thus achieving a better generalization over the unseen data. At the testing stage, SVM classifies input vectors as positive or negative based on the side of the Hyper-plane to which are mapped. Furthermore, SVM computes the separating Hyper-plane by using a kernel function that transforms the current features into higher dimensional feature spaces. Data that is separated linearly is classified using Linear kernel and data which is separated non-linearly is classified utilizing Radial Basis Function (RBF) kernel. In addition, SVM use some other kernel functions are Polynomial, Sigmoid, and Gaussian kernels. It uses the following discriminant function:

$$f(x) = W^N g(x) + b \tag{4.5}$$

where W represents the vectors weight, and $g(x)$ denotes a non-linear mapping between input feature to high dimensional feature, b presents the term of bias. Figure 4.1 illustrates a Hyper-plane that separates two classes linearly into two-dimensional spaces.

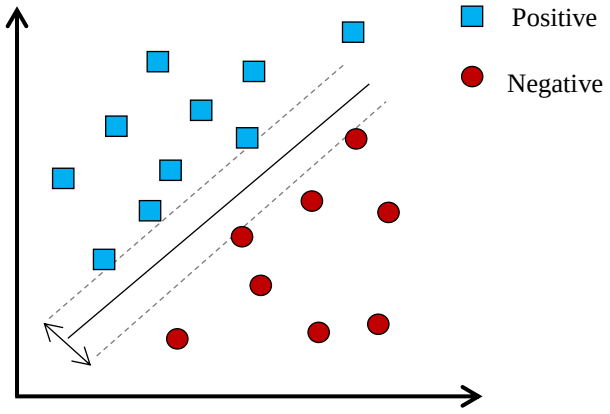


Figure 4.1: An example of a Hyper-plane linearly separate two classes (redrawn from [140])

4.4.2 Naive Bayes (NB) Classifier

NB classifier [272] is a conditional probability algorithm belonging to the probabilistic-based classifiers family based on Bayes' theorem which is mostly applied in sentiment analysis. NB classifier assumes that the presences of features are strongly uncorrelated with other features and computes the probability. In short term, NB classifier determines the probability of class C given the document that input vector X will occur, thus all features in input vector X are assumed mutually independent, and therefore NB is formulated as:

$$P(C_i\,|X) = \frac{P(C_i)\,P(X|C_i)}{P(X)} \tag{4.6}$$

where C_i defines the classes and X denotes the input vector spaces, thus $P(C_i)$ and $P(X)$ are the prior probability of class i and text vector in given document respectively, accordingly $P(X|C_i)$ is the likelihood which represents the probability of input vectors appearing in given class i.

Carrying with the equation (4.6), the probability of text vector in the given document is often shunned, thus the final representation of classifying function can be reduced as:

$$Y^{\wedge} = argmax_{i\in\{1,\dots,j\}}\, P(C_j) \prod_{i=1}^{k} P(X_i|C_j) \tag{4.7}$$

4.4.3 K-Nearest Neighbors (KNN) Classifier

This model classifies a document by evaluating its distance from other documents. [273]. All the *K* neighbors in the training documents are computed and the category with the highest number of *K* neighbors determines where to assign the document [274]. When the value of *K* is small, it indicates that noise will significantly influence the result, and a large *K* reduces the effect of noise. In this research, the *K* value is set to 5. Figure 4.2 demonstrates the basic principle behind the KNN classifies used to categorize unseen data into already observed classes based on its neighbors.

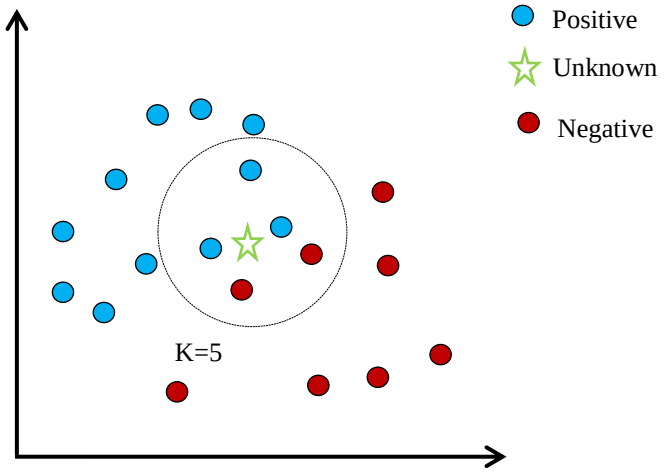


Figure 4.2: KNN classifier principle.

4.4.4 Logistic Regression (LR) Classifier

LR classifier is a more efficient and flexible statistical analysis algorithm that belongs to the generalized linear family of models. LR also is the expansion of linear regression methods used for situations where outcomes are categorical variables. LR classifier has been extensively employed in sentiment analysis problems where it defines the response variables with more than one predictor variables [275]. Furthermore, LR utilizes a logistic function to model the probabilities which define the prediction of output. Thus, it attractively predicts the categorical dependent variable through analyzing the relationship between one or more existing independent variables. LR classifier is formulated in the following form:

$$P = \frac{1}{1 + e^{-(b_0 + b_1 X_1 + \cdots + b_n X_n)}} \tag{4.8}$$

where P is the predicted probability which the output is present, b_i for $\{ i = 0, 1, \ldots, n\}$ represents the regression coefficients, X_j for $\{ j = 1, 2, \ldots, n\}$denotes different independent variables.

4.4.5 Stochastic Gradient Descent (SGD) Classifier

SGD classifier is an efficient and easy algorithm for implementation which improves many loss functions like linear SVM and LR classifier [276]. This classifier is generally utilized for optimizing the linear function and here the concept of stochastic is introduced based on the roots finding nature of the optimization task. In SGD, a term of the batch is used at each iteration to select the number of samples randomly instead of the entire dataset and these batches are used for calculating the gradient for each iteration. SGD formula can be presented as follows:

$$W_{t+1} = W_t - \alpha \frac{\partial L}{\partial W} \tag{4.9}$$

where, W_{t+1} and W_t are current and old weight respectively, $\frac{\partial L}{\partial W}$ is the current gradient multiplied by some factor α called the learning rate used to update W_{t+1} .

4.4.6 Decision Tree (DT) Classifier

DT classifier [277] is one of the most well-known learners that is perfectly applied in sentiment mining as it can order classes on a precise level. DT classifier does not require any domain knowledge for its construction. Furthermore, it can handle both categorical and numerical text data and is also able to handle high dimensional and noise data. DT constructs classification models in the form of a tree structure in which data points are broken down into smaller subsets and gradually an associated DT is incrementally constructed [277]. The result of this process shows a tree with decision and leaf nodes, the top of the decision node in a tree is called the root node which corresponds to the best predictor as shown in Figure 4.3. It also depicts that if a particular sequence of outputs has occurred then which decision node has the high probability to occur and what class label will be assigned for that sequence. The main idea behind DT is the use of the Iterative Dichotomiser 3 (ID3) algorithm which

utilizes IG and Entropy function for constructing a DT. The following formula shows using the concepts of Entropy function to find the split point and the feature to split on, and mathematically it can be written as:

$$E(\delta) = -\sum_{i}^{n} P(C_i)\, log_2\, (P(C_i)) \tag{4.10}$$

Where i represents the number of features; $P(C_i)$ is the probability of class C_i in a dataset; and δ present target class.

Figure 4.3 represents the DT classifier structure in which the root node represents a test on a feature and each decision node represents an outcome of the test and each leaf node denotes a class label.

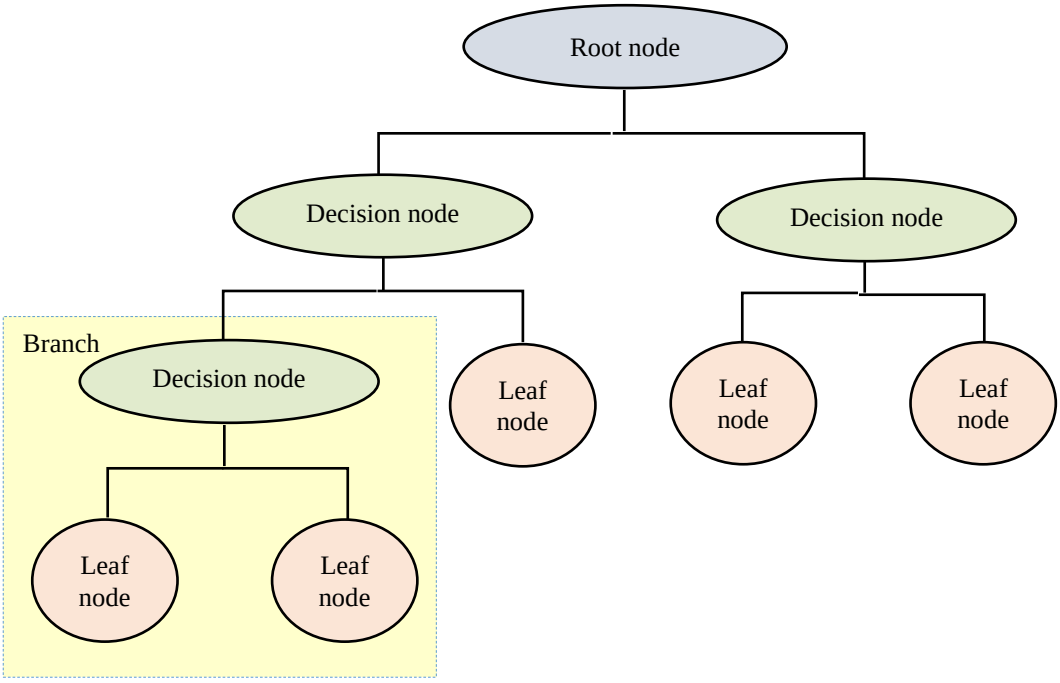


Figure 4.3: DT classifier diagram

4.4.7 Random Forest (RF) Classifier

RF classifier is also known as a random decision forest. It is one of the ensemble learning algorithms which can be used for both regression and classification tasks. RF algorithm constructs a multitude of DT models based on the combination of those multiple DT models, resulting in a forest of a tree, therefore it can be termed as the collection of tree-structured classifiers. In the beginning, RF trains many DT classifiers where each tree is constructed using a random subset of different vector features. After that, the sequence of vector features and their values generate a route to leaves which represent the decisions. Then the decisions of all trees are fitted into a meta-estimator to make a forest. RF uses a majority voting algorithm to derive the resultant class label from the generated classes through similar subsampled trees generated as the RF outcome as shown in Figure 4.4. In RF at training time, the decision node values are updated to reduce a cost function that estimates the performance of the trees. In addition, RF decreases variance through training different samples of the dataset and utilizing a random sample of different vector features [278, 279]. Furthermore, the use of more trees in the RF algorithm generally corresponds to better performance and produces effective predictive outcomes [279]. RF model uses the Gini Index formula when performing it to solve the classification problems to decide how nodes branch in a DT. This formula utilizes both class and probability to define the Gine of each branch on a node, determining which of the branches is more likely to occur. The mathematics formula behind RF can be represented as follows:

$$Gini = 1 - \sum_{i=1}^{n} (P_i)^2 \tag{4.11}$$

where P_i denotes the relative frequency of the class observed in the dataset, n is the number of classes.

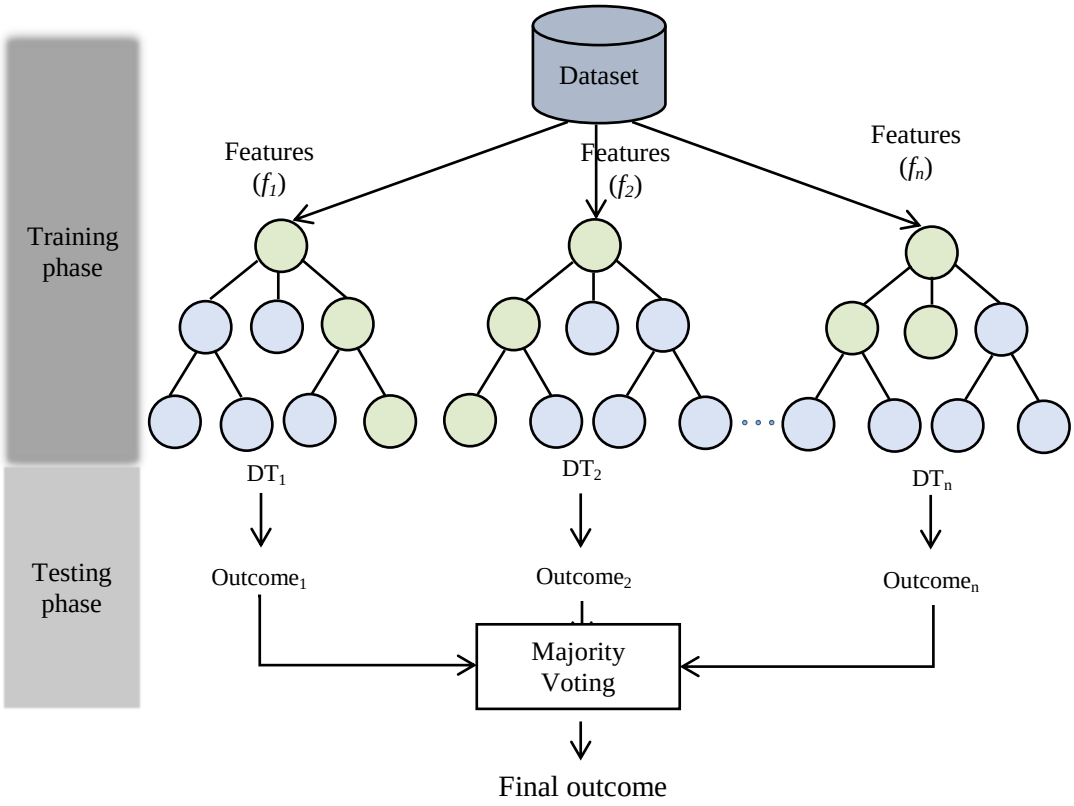


Figure 4.4: Structure of RF classifier (redrawn from [279])

4.5 Ensemble Classifiers

Ensemble learning uses multiple classifiers for data training and final predictions and performs better than a single classifier [154]. There is, however, no guarantee that ensemble learning algorithms will always perform better than a single, trained, machine learning algorithm [154,155]. There are two types of ensemble learning methods, namely common ensemble methods (Section 2.5.1) and combining ensemble methods. The combining ensemble methods consist of two methodologies which are: simple combining (Section 2.5.2) and meta-combining methods (Section 2.5.3). For this thesis, Bootstrap Aggregation (or Bagging) and Boosting are used from the common combining methods, while Simple Majority Voting and Weighted Majority Voting are used from the simple combining methods for sentiment analysis. These methods are described in the sections that follow.

4.5.1 Bootstrap Aggregation

Bootstrap Algorithm (with Aggregation and Bagging) has been widely used in text classification. With this method, resampling is used to generate multiple base classifiers in parallel where training data subsets are drawn and replaced randomly and base learners are then trained on all the subsets. To make a final decision, the results of each model are aggregated together [55].

4.5.2 Boosting

Boosting is an ensembling approach that uses a set of low accuracy classifiers to create a high accuracy classifier. Boosting is based on the idea of sequentially training weak learners, each of which tries to improve on its predecessor [280]. The algorithm improves on the previous performance by correcting the wrong predictions in the next iterations. This classifier has subtypes such as AdaBoost (Adaptive Boosting), Gradient Tree Boosting, and XGBoost [55]. All these three variants are applied in our experiments.

4.5.2.1 AdaBoost

AdaBoost, short for "Adaptive Boosting", is an ensemble learning algorithm that constructs a set of weak classifiers through multiple iterations. Each instance in the training set has a weight, and the weights of the instances which were misclassified are increased and used for the next weak classifier in the next iteration. A new weak classifier is added to the classifier set after each iteration. The process continues till a small error rate accuracy is obtained or a maximum iteration time is reached [281].

4.5.2.2 Gradient Tree Boosting

This algorithm develops a predictive model based on the boosting and decision tree learning algorithms. It takes its roots from a statistical framework known as the Adaptive Reweighting and Combining (ARC) algorithm which was introduced by

Breiman [282]. Usually, decision tree algorithms grow a single large tree to fit the data and this leads to overfitting and high variance. As a remedy to such problems, the boosting algorithm is designed in a way that decision trees minimize the variance with Gradient Tree Boosting. Gradient Tree Boosting makes use of the long learner tree which is grown sequentially where it learns iteratively while fixing the error of previous iterations [283]. This results in an output with low variance and error.

4.5.2.3 eXtreme Gradient Boost (XGBoost)

XGBoost is short for eXtreme Gradient Boosting [284] which is based on the gradient boosting framework. Gradient Boosting is a tree ensemble boosting approach which combines a group of weak models to produce a robust classifier. The robust classifier is trained iteratively starting with a base classifier. Both XGBoost and Gradient Boosting follow the same principle. The main differences between them are in the details of implementation. XGBoost provides better performance by managing the complexity of the trees using different regularization approaches. XGBoost consists of a set of DT that uses the Gradient Descent technique to reduce the errors of weak estimators, using the training loss and regularization term as the objective function. In boosting, new models are sequentially added to the errors made by existing models until no more improvements are feasible. In the gradient boosting approach, new models are constructed to predict the residuals (or errors) of previous models, and then all models are used together to make the final prediction. The name gradient boosting means that the technique uses a gradient descent algorithm while adding new models to minimize loss. This approach can be used for both regression and classification modeling problems [285].

XGBoost uses two additional techniques besides regularization to improve the model's performance. The reduction of weights is the first approach, which is accomplished by scaling newly added weights with parameter η, also known as the learning rate. This reduces the influence of an individual tree and gives room to future trees to enhance the model. Another approach for improving the model is feature sub-sampling. It operates similarly to Bagging does in the RF algorithm by choosing sub-samples of features for each tree. This is done to decorrelate features, decrease bias, and keep the ensemble model from overfitting. Furthermore, compared to other ensemble models, the XGBoost approach provides many computational advantages such as cache-aware settings, block structure for parallel learning, and out-of-core computations [284]. The advantage of XGBoost is its speed and performance which can be attributed to its utilization of parallel computing that makes learning faster. Regardless of the size of the data or the number of machines, XGBoost runs relatively faster than other algorithms. XGBoost has been shown to run over ten times faster than other algorithms and outperform them [285]. Both XGBoost and Gradient Boosting are ensemble tree techniques that employ the gradient descent architecture to boost weak learners. However, XGBoost improves upon the base Gradient Boosting framework through systems optimization and algorithmic enhancements [284]. The system optimization such as:

- **Parallelization:** XGBoost uses parallelized implementation to tackle the process of constructing sequential trees. This is conceivable because of the interchangeable nature of loops used to structure base estimators; the outer-loop enumerates the leaf nodes of a tree and the inner-loop calculates the features. This nest-loops limits parallelization due to the outer-loop cannot be started before the inner-loop is completed. As a

result, the order of loops is swapped utilizing initialization via a global scan of all instances and sorting using parallel threads to optimize run time. This swap increases the algorithmic speed by balancing any parallelization overheads in computation.

- **Tree Pruning:** the greedy stopping criteria for tree splitting in the Gradient Boosting framework is based on the negative loss criterion at the split point. XGBoost first utilizes the "max_depth" parameter instead of criterion then starts pruning trees backward. This "depth-first" method greatly enhances computing performance.
- **Hardware Optimization:** This method was created to make the best use of available hardware resources. This is done by cache awareness, which involves each thread creating internal buffers to hold gradient statistics. Further improvements such as "out-of-core" computing optimize available disk space while handling large data frames that do not fit in memory.

The algorithmic enhancements such as:

- **Regularization:** To avoid overfitting, it penalizes more complicated models using both LASSO (L1) and Ridge (L2) regularization.
- **Sparsity Awareness:** By automatically 'learning' the optimum missing value based on training loss, XGBoost naturally allows sparse features for inputs and handles different forms of sparsity patterns in the data more efficiently.

- **Weighted Quantile Sketch:** The distributed weighted Quantile Sketch technique is used by XGBoost to determine the best split points across weighted datasets.
- **Cross-validation:** The technique includes a built-in cross-validation procedure at each iteration, eliminating the need to implement this search directly and to define the precise number of boosting iterations needed in a single run.

4.5.3 Simple Majority Voting

The Simple Majority Voting algorithm is an ensemble method that combines the predictions made by several classifiers [167]. The algorithm can equally train a set of classifiers with good performance so that the outcome of each classifier is enumerated to improve their weaknesses. Majority voting can be computed as follows:

$$y(x) = \{h_1(x), h_2(x), \dots, h_n(x)\} \tag{4.12}$$

where $h_1(x), h_2(x), \dots, h_n(x)$ are n classification rules, the value of each x predicts to the class with the highest number of votes.

4.5.4 Weighted Majority Voting

This is a combiner algorithm that is used in the general voting category. It is intended to produce a meta-learning classifier that is associated with a specific weight for confidence. The predictive performance of the individual base classifier determines how weights are assigned. The weights are considered during vote collection when the impact of the base classifiers' prediction is increased and decreased [286]. There are two ways of using weights in this method: either the weights are set as a constant or each base classifier has a separate weight per class to correspond with the strength of that classifier in prediction. When the latter is the case, it is called class-based

weighted majority voting [287]. Mathematically, the weighted majority voting algorithm is written as in the following formula:

$$y^{\wedge} = argmax_i \sum_{j=1}^{n} W_j X_A(C_j(X) = i) \quad (4.13)$$

In this formula, W_j is the weight of the j^{th} base classifiers, C_j is an ensemble, $y^{\wedge}$ is the predicted class of the ensemble classifiers, X_A represents the characteristic function$C_j(X) = i \in A$, and A connotes the set of unique class labels.

4.6 Evaluation Metrics

Classifier evaluation in sentiment analysis focuses on the effectiveness of the classifier rather than the efficiency [287]. The focus is on how well the classifier makes predictions and not on the computational complexity. The following metrics are widely used in text mining: accuracy, precision, recall, and F_1-score. These metrics are obtained from a confusion matrix that records the correct versus wrong classified cases per category [287, 288]. Table 4.3 gives the confusion matrix.

Table 4.3: Confusion matrix

Predicted Class	**Actual Class**	
	Class Positive	Class Negative
Predicted Positive	TP	FN
Predicted Negative	FP	TN

From Table 4.3, TN stands for true negatives, this is the number of negative cases correctly predicted by the classifier. TP stands for true positives which is the number of positive classes correctly predicted by the model. FP stands for false positives which is the number of negative classes wrongly predicted as positive classes. Finally, FN stands for false negatives, which is the number of positive classes wrongly predicted as negative. Using these metrics, the performance of the proposed

methods (as described in Chapter 5 and Chapter 6) on all datasets is computed from the confusion matrix as shown below:

- Accuracy: total number of correctly predicted documents divided by the total number of documents. The formula for computing accuracy is given by:

$$Accuracy\ (Acc) = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (4.14)$$

- Precision: the number of true positives out of all positively assigned documents. It is computed as:

$$Precision\ (Pre) = \frac{TP}{TP+FP} \times 100 \quad (4.15)$$

- Recall: the number of all the true positives out of the total actual positive documents. It is computed as:

$$Recall\ (Rec) = \frac{TP}{TP+FN} \times 100 \quad (4.16)$$

- F_1-score: this is a tradeoff between precision and recall. It is computed as:

$$F_1\text{-}Score = 2 \times \frac{Pre \times Rec}{(Pre+Rec)} \times 100 \quad (4.17)$$

- Rec_{macro1} score: it gives the macro-averaged score of Recall among all classes, both positive and negative.

$$Rec^{Positive} = \frac{TP}{TP+FN} \quad (4.18)$$

$$Rec^{Negative} = \frac{TN}{TN+FP} \quad (4.19)$$

$$Rec_{macro1} = \frac{1}{2}(Rec^{Positive} + Rec^{Negative}) \times 100 \quad (4.20)$$

- Rec_{macro2} score: it gives the macro-averaged score of Recall in all classes, which is the positive, negative and neutral class.

$$Rec^{Positive} = \frac{TP}{TP + FN + FU} \quad (4.21)$$

$$Rec^{Negative} = \frac{TN}{TN + FP + FN} \quad (4.22)$$

$$Rec^{Neutral} = \frac{TU}{TU + FP + FP} \quad (4.23)$$

$$\text{Rec}_{macro2} = \frac{1}{3}(Rec^{Positive} + Rec^{Negative} + Rec^{Neutral}) \times 100 \quad (4.24)$$

- $F_{1\text{-}macro}$: this is the macro-averaged score F_1-score for both the positive and negative classes. The $F_1^{Positive}$ for the positives is obtained by calculating the corresponding precision ($Pre^{Positive}$), where $Pre^{Positive}$ represents the ratio of correctly predicted positive messages. The $F_1^{Negative}$ for negative is obtained by calculating the corresponding precision ($Pre^{Negative}$), where $Pre^{Negative}$ stands for the ratio of correctly predicted negative messages.

$$Pre^{Positive} = \frac{TP}{TP + FP} \quad (4.25)$$

$$Pre^{Negative} = \frac{TN}{TN + FN} \quad (4.26)$$

$$F_1^{Positive} = 2 \times \frac{Pre^{Positive} \times Rec^{Positive}}{Pre^{Positive} + Rec^{Positive}} \quad (4.27)$$

$$F_1^{Negative} = 2 \times \frac{Pre^{Negative} \times Rec^{Negative}}{Pre^{Negative} + Rec^{Negative}} \quad (4.28)$$

$$F_{1-\text{macro}} = \times \frac{1}{2}(F_1^{Positive} + F_1^{Negative}) \quad (4.29)$$

- Macro-averaged Mean Absolute Error (MAE^M): this computes the Mean Absolute Error (MAE) separately for each class after which the average is taken for all classes, hence all classes are treated equally.

$$MAE^M(h, Te) = \frac{1}{|C|} \sum_{j=1}^{|C|} \frac{1}{|Te_j|} \sum_{X_i \in Te_j} |h(X_i) - y_i| \quad (4.30)$$

- Micro-average Mean Absolute Error (MAE^{μ}): this takes the aggregates of all contributions by each class and then computes the average.

$$MAE^{\mu}(h, Te) = \frac{1}{|Te|}\sum_{X_i \in Te} |h(X_i) - y_i| \quad (4.31)$$

In the equation, y_i stands for the actual target of X_i, while $h(X_i)$ is its predicted target, Te_j is the set of test documents with the actual class, $C_j, |h(X_i) - y_i|$ is the distance spanning the predicted class $h(X_i)$ and actual class y_i .

It should be noted that the MAE^{M}, is more appropriate for measuring the classification accuracy of systems having imbalanced datasets than the MAE^{μ} [17].

Chapter 5

SENTIXGBOOST: ENHANCED SENTIMENT ANALYSIS IN SOCIAL MEDIA POSTS WITH ENSEMBLE XGBOOST CLASSIFIER

Sentiment analysis has been widely used in the area of text mining. This chapter reports on a novel framework developed to facilitate the implementation of an ensemble classifier approach for sentiment analysis tasks. The novel ensemble classifier employs XGBoost as a meta-classifier for stacked ensembling. The ensemble classifier framework employed in this work combines multiple feature sets with ensemble classification where multiple base classifiers which are weak learners are combined into an ensemble classifier. These feature sets include BoW, TF-IDF, PoS, *n*-gram, Opinion Lexicon, and Term Frequency. The use of XGBoost as a meta-classifier is a significant contribution to this work. The developed method through this framework combines several individual classifiers to form an ensemble. Two experimental settings were employed during the validation of our proposed method. Both settings provided good performance comparable to single base classifiers, different strategies of ensemble classifier techniques, and the existing methodologies. This has therefore justified the reliability of our approach. In this chapter, we describe the proposed SentiXGboost method to generate a novel ensemble classifier approach for the sentiment analysis task. This chapter is organized as in the following. The description of the proposed SentiXGboost method architecture is

presented in section 5.1. In section 5.2, we first present the details of the datasets employed and then discuss the experimental settings employed in this method. Finally, we discuss the results of the SentiXGboost method then compared its results with the other state-of-the-art methods.

5.1 Proposed SentiXGboost Method Architecture

The proposed framework relies on a combination of six base classifier concepts in machine learning used as input to the XGBoost algorithm. This is done to improve classification performance. In this section, the details of the proposed sentiment analysis method called "SentiXGboost" are presented. The system architecture of the proposed framework is depicted in Figure 5.1. The initial phase of the system handles the preprocessing of the train and test data. Activities carried out at this phase include tokenization, stemming, and removal of stop words before the feature extraction phase, as shown in Figure 5.1. BoW, PoS, TF-IDF, *n*-gram, Opinion Lexicon, and Term Frequency features are extracted at the feature extraction phase, and the combination of all features is used for training the base classifiers. In the proposed architecture, the most widely used and efficient classifiers employed for sentiment classification, namely DT, NB, RF, KNN, LR, and SGD [198,289-291] are trained as base classifiers. One of the methods that enhance classification accuracy is the ensemble learning method which combines the outcome of weak classifiers to form a single, robust classifier. In this method, the predictions of the base classifiers are combined and used as input to the XGBoost classifier, as shown in Figure 5.1. XGBoost, an advanced implementation of the Gradient Tree-Boosting algorithm, was implemented by Chen Tianqi in 2016 [284]. XGBoost is also known as Regularized Boosting technique because it contains several regularizations, which decrease over-fitting and improve the performance of classifications. Notably, it has higher

predictive power and is approximately ten times faster than the Gradient Tree-Boosting algorithm [292]. XGBoost classifier is trained as a meta classifier that combines weak learners to produce a robust learner. For the given training data X_i and their labels Y_i , XGBoost classifier utilizes individual classifiers to predict the outcome Z_i.

$$Z_i = \sum_{n=1}^{N} f_n (X_i), \tag{5.1}$$

where function f_n represents the n^{th} a DT that contains scores on its leaves. The following function calculates the score of each tree:

$$L^{(n)} = \sum_{i=1}^{k} l(Y_i , Z_i^{(n-1)} + f_n (X_i)) + \Omega f(n), \tag{5.2}$$

where l represents loss function, $Z_i^{(n)}$ denotes the prediction for sample X_i at n^{th} iteration and Ω is the regularization term, which prevents the score leaves from obtaining large values. $f_n (X_i)$ are inserted into the tree function to achieve the final classification tree. The parameters of the model, namely silent, scale_pos_weight, learning_rate, colsample_bytree, subsample, objective, n_estimators, reg_alpha, max_depth, and gamma, are empirically set to False, 1, 0.01, 0.4, 0.8, 'binary:logistic', 100, 0.3, 4, and 10, respectively. In the proposed method, we used the DT classifier as base_estimator, and at each iteration, a weak classifier is added to the classifier ensemble until the ensemble yields the correct classification.

In Algorithm 1, after extracting features from training data, the features are then used to generate individual classifiers which serve as input to the meta-classifier XGBoost. Each base classifier is trained using gold-labeled training data and the same combined feature set to determine the sentiment polarity of tweets. Next, outputs of all base classifiers are combined to form the sample distribution used for

training the meta-classifier XGBoost. In step 3, XGBoost trains several DT classifiers sequentially using the output of the six base classifiers as input sample distribution in such a way that each new DT classifier focuses on samples that were misclassified by the previous DTs. The new model is fitted to the residuals or errors generated from the previous prediction at each iteration. These models in combination with previous models perform the final prediction. Notably, XGBoost uses the gradient descent algorithm to minimize the loss whenever a new DT is added.

Algorithm 1: train proposed SentiXGboost classifier

Input: tweets and related sentiments as training data.

Method:

(1) Generate feature set: BoW, TF-IDF, PoS, *n*-gram, Opinion Lexical and Term Frequency

(2) Train base classifiers C_i, using the feature set

(3) Combine output of all classifiers C_i into sample distribution D;

- Input: D as sample distribution;
- $\mathfrak{I}$ as base classifier;
- T as a counter for learning rounds.
- Process:
  - $D_t = D$; initialization of D
  - While = 1 to T:
    - $h_t = \mathfrak{I}(D_t)$; A weak classifier is trained from D_t
    - $\epsilon_t = P_{x\sim Dt}\,(h_t\,(x) \neq y)$; the error h_t is evaluated
    - D_{t+1} = Adjust_Distribution(D_t, ϵ_t)

- End While

- Output: $H(x)$ = Combine_outputs ($\{h_1(x), \ldots, h_t(x)\}$)

(4) Save (the trained SentiXGboost classifier)

(5) Return (the saved classifier)

SentiXGboost classifier uses Algorithm 2 to predict the sentiment of a given tweet. SentiXGboost pre-processes the input and generates the feature set employed in the training phase. The base classifiers contained in the SentiXGboost method evaluate the sentiment polarity of the input, and meta-classifier XGBoost combines these predictions to form the final prediction result of SentiXGboost.

Algorithm 2: make a prediction using the SentiXGboost classifier

---

Input: tweets making up the test set.

Output: predicted tweet sentiments

Method:

(1) Generate feature set: BoW, TF-IDF, PoS, *n*-gram, Opinion Lexical and Term Frequency

(2) For each trained classifier C_i in Classifier ensemble do

- Make a prediction for the input sample using the feature set

(3) Aggregate the predictions to obtain the final prediction result for the XGBoost Classifier

(4) Return (final prediction)

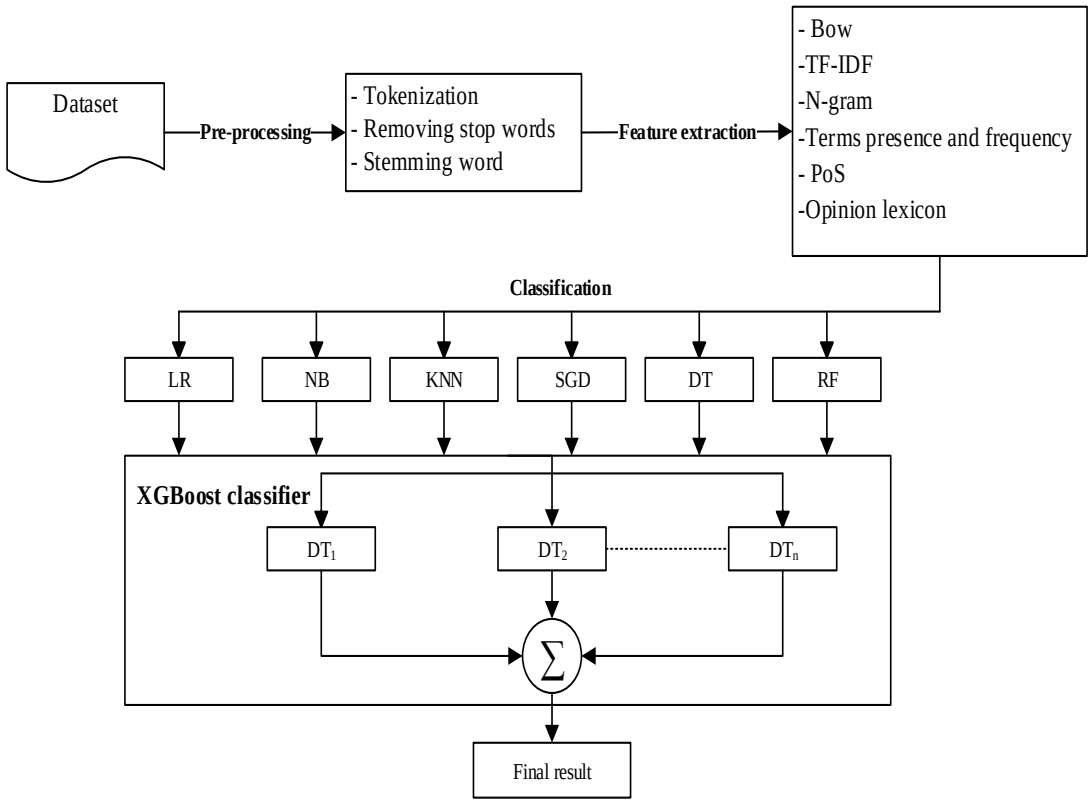


Figure 5.1: The proposed SentiXGboost method architecture

5.1.1 Individual Classifier Used

Supervised machine learning approaches have previously been utilized for the classification of texts according to defined classes. As described in Section 4.4, the following classification algorithms have been deployed for tweets sentiment mining in this study: NB, LR, KNN, SGD, DT, and RF. All these classifiers are trained with different parameter settings using different combinations of features. The details of all base classifiers with their parameter settings are presented in Table 5.1.

Table 5.1: Individual classifiers with their parameter settings

Individual Classifiers	**Parameter Settings**
NB (MNB)	*alpha* "1.0": the additive smoothing.
LR	*penalty* = "12": the normality of penalization, *multi_class* =" auto": binary versus multiclass label data.
KNN	*n_neighbors* = "5": number of neighbors.
SGD	*loss* = "*log*": set logistic regression as the loss function, *penalty* = "12", The penalty (a regularization term) to be used, "12" is the standard regularizer for linear SVM models, *alpha* = "*le*-3", a constant that multiplies the regularization term, *random-state* = "42": for reproducibility in controlling randomness of samples.
DT	*n-estimator* = "100": number of trees, *max_depth* = "3", maximum depth of the tree, *min_samples_split* = "2": minimum samples required to split the node, *min_samples_leaf* = "1": minimum number of samples to be at the leaf node, *random_state* = "0".
RF	*n_estimators* = "100": number of trees in the forest, *max_depth* = "3": maximum depth of the tree in the forest, *min_samples_split* = "2", *min_samples_leaf* = "1", *max_features* = "auto": number of features used for the best split, *random_state* = "0".

5.2 Experimental Results and Evaluation

This section discusses the datasets and experimental settings conducted to evaluate the performance of the proposed SentiXGboost method. The experimental results of the proposed SentiXGboost method were obtained through experiments and are compared with all individual classifiers used in the proposed architecture. The comparison is also done for the different strategies of ensemble classifier techniques in terms of classification performance and efficiency analysis. Furthermore, the

performance comparison of the proposed SentiXGboost method against other existing sentiment analysis methods using the same datasets is carried out.

5.2.1 Statistics on Datasets Used

The performance of the proposed SentiXGboost method for sentiment analysis is evaluated using the following sentiment datasets: SemEval 2017 Task 4B, SLS, STS-Gold, SST-2, Yelp Challenge, and Movie Review. These datasets, obtained from tweets, are currently the most comprehensive publicly available sentiment-related datasets for conducting sentiment analysis on in tweeter. Each of them is grouped into the training and test datasets. The details of the number of positive and negative samples contained in the train and test sets for each dataset are given in Table 5.2.

Table 5.2: Statistics of the datasets employed in this experiment

Datasets	Type	Positive	Negative	Total
SemEval-2017 Task 4B	Train	5779	1606	7385
	Test	2433	733	3166
SLS (Amazon)	Train	353	347	800
	Test	147	153	200
STS-Gold	Train	435	988	1423
	Test	197	414	611
SST-2	Train	3438	3290	6728
	Test	1525	1359	2884
Yelp Challenge Dataset	Train	34300	35700	70000
	Test	15700	14300	30000
Movie Review(Sentiment Polarity Dataset V2.0)	Train	26285	25491	51776
	Test	6652	6292	12944

5.2.2 Experimental Settings

In this work, we compared the proposed SentiXGboost method with all individual classifiers used in the proposed architecture, including NB, LR, KNN, SGD, RF, and DT (as described in Section 4.4). Further, different ensemble methods, namely

majority voting, bagging, and boosting (as described in Section 4.5) are used for comparison with the proposed method.

The majority voting is useful when combining a set of classifiers that compete well with each other such that the weakness of each is balanced out and the performance is improved [55]. In the experiments involving the majority voting, we employed the same base classifiers, namely DT, NB, RF, KNN, LR, and SGD, used as a part of the SentiXGboost. Simple majority voting is used to combine the predictions of these base classifiers into the ensemble prediction.

The BootStrap Algorithm is widely used in text classification where several base classification models are generated in parallel through resampling. Subsets from the training set are randomly drawn and replaced for use in training a different base learner from the set of ensemble learners. In the end, the final prediction is obtained by aggregating the results from individual models [55]. In the experiments for Bagging, the LR classifier was used as a base estimator, setting *max_samples* = 0.5 and *max_features* = 0.5.

Boosting works sequentially to produce a classifier with high accuracy from a set of classifiers with low accuracies. The learners are trained sequentially such that each learner attempts to improve on the previous learner [280]. The algorithm corrects the misclassifications of the previous iteration in the current iteration. Most common variants of boosting include AdaBoost (Adaptive Boosting), Gradient Tree Boosting, and XGBoost [55]. All these boosting methods are deployed in this study. The AdaBoost algorithm constructs a model in the initial iteration and another model is constructed in the second iteration by increasing the weights of the misclassified

observations. The process continues iteratively until an optimal model is obtained. For the experiments in this study involving the AdaBoost model, we employed as base classifier the DT classifier with settings *n_estimators*= 50, *learning_rate*= 1, and *random_state*= 0. For the Gradient Tree Boosting model, the misclassified observations of the previous model are used to train new models. The DT base classification model in the experiments relating to the Gradient Tree Boosting algorithm was set at *n_estimators*= 100.

The following gives an outline of the two experiments conducted to determine the performance of each classification model individually:

- The set of features, namely BoW, PoS, TF-IDF, *n*-gram, Term frequency, and Opinion Lexicon, were combined to train the following classifiers: NB, LR, KNN, SGD, RF, and DT. The training of the models was done on the training set, which constitutes 70% of the data. The remaining 30% was kept as the test set for validating the classifiers.
- The following ensembling techniques: Majority voting, AdaBoost, Gradient Tree Boosting, Bagging, and SentiXGboost were trained using 70% of the dataset, which constitutes the training set while 30% of the data was set aside as a test set. The same feature combinations as the first set of experiments were used for majority voting.

5.2.3 Analysis Results and Evaluations

This section discusses the classification performance of the proposed SentiXGboost method in terms of Accuracy, Precision, Recall, Average Recall, and F_1-Score. More details about these metrics are provided in section 4.6. These performance metrics are also being frequently used to evaluate NLP models, including sentiment analysis

tasks. Furthermore, we also compare the results of the proposed method with some existing methods that used the same dataset as well as compare to recent research that used various methodologies.

5.2.3.1 Analysis Results

The performance of SentiXGboost was evaluated using the following performance metrics: Accuracy, Precision, Recall, and F_1-Score. This section compares the results of SentiXGboost with those of all the base models used in the proposed architecture. Further, the performance of the proposed method is compared with the different approaches of the ensemble classifiers.

Table 5.3 presents the performances of classifications on SemEval-2017 Task 4 and Subtask B dataset, produced by each of the mentioned base classifiers used in the proposed SentiXGboost architecture and the ensemble classifiers. The results show that the proposed approach performs better than all the individual classifiers and the classifier ensembles.

Table 5.3: Performance of the individual classifiers and ensembling approaches using SemEval-2017 Task 4, Subtask B dataset

	Classifiers	**Acc (%)**	**Pre (%)**	**Rec (%)**	**F_1-Score (%)**
Individual classifiers	NB	86.3	85.4	96.8	91.4
	KNN	78.3	79.4	96.1	87.2
	LR	**88**	**89**	95.5	**91.9**
	RF	85.2	84.5	**96.9**	91
	SGD	87.8	88.7	93.8	91.5
	DT	82.4	87.4	90.2	88.8
Ensemble classifiers	Majority voting 6 classifiers	88.2	91.3	88.5	92.7
	AdaBoost	88.9	89.5	97.1	92.8
	Gradient tree boosting	88.8	89.4	97	93
	Bagging	85.8	84.3	97	91.5
	Proposed SentiXGboost	**90.8**	**92.7**	**98.1**	**94**

[*Acc: Accuracy, Pre: Precision, Rec: Recall.*]

The performance of the six classification models and the proposed method in terms of Accuracy, Precision, Recall, and F_1-Score are shown in Table 5.3. There could be seen that the LR classifier yielded better performance relating to Accuracy, Precision, and F_1-Score, which scored 88%, 89%, and 91.9%, respectively. On the other hand, the RF classifier achieved the highest Recall score, which is 96.9%. Furthermore, the lowest measures in Accuracy, Precision, and F_1-Score are for the KNN classifier, which are 78.3%, 79.4%, and 87.2%, respectively. On the other hand, the DT classifier yielded the lowest scores for the Recall metric, which is 90.2%. The implication is that the LR classifier performs better than other single base classifiers on the SemEval-2017, Task 4, and Subtask B datasets. A related study by Pranckevičius and Marcinkvičius [293] shows that the LR classifier performs better than all other individual classifiers in the analyses of Amazon and Heart Disease datasets. Similarly, the results of the study by Çığşar and **Ü**nal [294] corroborate our results which show that the LR classifier performs better than other classifiers on the Turkish Statistical Institute 2015 survey dataset.

The results in Table 5.3 show that all ensemble methods yield better performance compared to the base classifiers. In contrast, the performance of the SentiXGboost ensemble algorithm is enhanced on the SemEval-2017 Task 4 and Subtask B datasets. The experiments show that the SentiXGboost ensemble algorithm performs better in terms of Accuracy, Precision, Recall, and F_1-Score, which are 90.8%, 92.7%, 98.1%, and 94%, respectively. Meanwhile, the Bagging ensemble algorithm gave the lowest scores in Accuracy, Precision, and F_1-Score with values 85.8%, 84.3%, and 91.5%, respectively. AdaBoost and Gradient Tree Boosting ensemble algorithms were ranked as the second and third highest, respectively, while the

Bagging ensemble algorithm performed worst. The Majority voting yielded the least Recall score of 88.5%. The consequence of these results is that the SentiXGboost ensemble algorithm is more appropriate for the SemEval-2017, Task 4, and Subtask B datasets.

To further evaluate the performance of the SentiXGboost approach, we performed extensive experiments using five sentiment datasets, namely SLS (Amazon), STS-Gold, SST-2, Yelp Challenge, and Movie Review (Sentiment Polarity Dataset Version 2.0). Table 5.4 and Figure 5.2 depict the performances obtained by individual and classifier ensembles, including SentiXGboost, based on Acc. These results show that the highest accuracy is achieved by the SentiXGboost approach on all datasets. According to the results obtained from these experiments, the highest Accuracy values for SentiXGboost are 92.5%, 91.1%, 85.2%, 92.75%, and 76.5%. Conversely, the highest Accuracy is achieved by LR among all individual classifiers, which show 86.5%, 87.2%, 79.4%, 85%, and 71.7% on SLS (Amazon), STS-Gold, SST-2, Yelp Challenge, and Movie Review (Sentiment Polarity Dataset Version 2.0), respectively. Notably, the KNN classifier exhibits the lowest performance of 65.5%, 72.2%, 57%, 64%, and 53.7% on all datasets, respectively.

Table 5.4: Accuracy of the individual classifiers and ensembling approaches for SLS (Amazon), STS-Gold, SST-2, Yelp Challenge, and Movie Review (Sentiment Polarity Dataset Version 2.0) datasets

	Classifiers	SLS	STS-Gold	SST-2	Yelp	Movie review
Individual classifiers	NB	86	86.9	79.2	82	70.2
	KNN	65.5	72.2	57	64	53.7
	LR	**86.5**	**87.2**	**79.4**	**85**	**71.7**
	RF	82.5	81.8	74.1	77	64.7
	SGD	81.5	84.7	79.2	83.66	67.6
	DT	78.5	81.5	66.5	71.5	56.2
Ensemble classifiers	Majority voting 6 classifiers	85	85.1	79.6	82	69.4
	AdaBoost	90	89.4	79.8	83.33	70.2
	Gradient tree boosting	90	90.2	81.8	85	70.8
	Bagging	82	87.8	80	83.3	70
	Proposed SentiXGboost	**92.5**	**91.1**	**85.2**	**92.75**	**76.5**

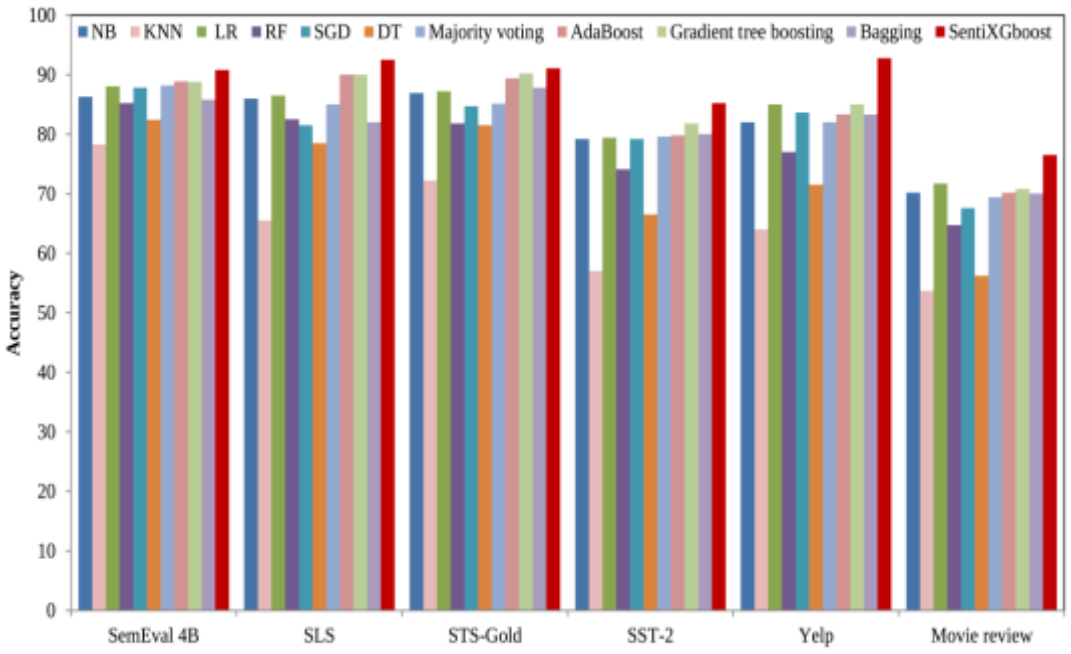


Figure 5.2: Comparison of accuracy of single and ensemble classifiers on sentiment labeled datasets.

Table 5.5 and Figure 5.3 present the F_1-Score performance achieved by individual and ensemble classifiers, including SentiXGboost. Notably, for all sentiment datasets, the highest F_1-Score for all datasets is obtained through the proposed SentiXGboost. In addition, it is evident that the gradient tree boosting classifier achieves the second-highest performance on all datasets, except the movie-review dataset, showing that boosting effectively increases the performance of the classifiers for the sentiment analysis task. The proposed method yielded an F_1-Score of 90% on the SLS dataset, which improves the second-best performance by 4.3%. The proposed method yielded an Accuracy of 84.7% on the STS-Gold dataset, while the second-best performance is improved by 2.4%. The proposed method has an F_1-Score of 86% on the SST-2, and the second-best performance is outperformed by 3.8%. The Yelp dataset performance of the proposed method is an F_1-Score of 93.75%, which improves the second-best performance by 8.05%. On the movie review dataset, the proposed method improves the performance of the gradient tree boosting algorithm by 3.8% and the second-best performance by 2.9% — still achieving the highest performance with an F1-score of 75.7%.

By examining the results in Table 5.5, it can be seen that the highest F_1-Score for individual classifiers obtained by the LR classifier are 85.4%, 77.6%, 80.9%, 85.58%, and 72.8% on the SLS, STS-Gold, SST-2, Yelp, and Movie review datasets, respectively. In addition, it is observed that the KNN classifier provides the lowest F_1-Score on all datasets. Notably, the difference in classification performance between LR and KNN ranges from 17.3% on the Movie Review dataset to 23.3% on the STS-Gold dataset.

Table 5.5: F_1-Score of the individual classifiers and ensembling approaches for SLS (Amazon), STS-Gold, SST-2, Yelp Challenge, and Movie Review (Sentiment Polarity Dataset Version 2.0) datasets

	Classifiers	**SLS**	**STS-Gold**	**SST-2**	**Yelp**	**Movie review**
Individual classifiers	NB	85.1	76.4	80.6	82.18	70.8
	KNN	62.2	54.3	61.5	63.63	55.5
	LR	**85.4**	**77.6**	**80.9**	**85.58**	**72.8**
	RF	79.2	60	75.2	75.3	66.5
	SGD	80.6	68.7	80.2	84.54	69
	DT	76.2	69.3	68.2	69.2	57.3
Ensemble classifiers	Majority voting 6 classifiers	82.7	70	81.88	81.05	69.7
	AdaBoost	85	79.9	80.5	82.75	70.9
	Gradient tree boosting	85.7	82.3	82.2	85.7	71.9
	Bagging	75.8	77.6	80.6	83.87	70.3
	Proposed SentiXGboost	**90**	**84.7**	**86**	**93.75**	**75.7**

The experiments show that the proposed SentiXGboost algorithm can enhance the general classification accuracy in Sentiment Analysis. Tables 5.4 and 5.5 illustrate that SentiXGboost is the best ensemble classifier because it yields better results for each of the SLS (Amazon) and STS-Gold, SST-2, Yelp Challenge, and Movie Review (Sentiment Polarity Dataset Version 2.0) datasets.

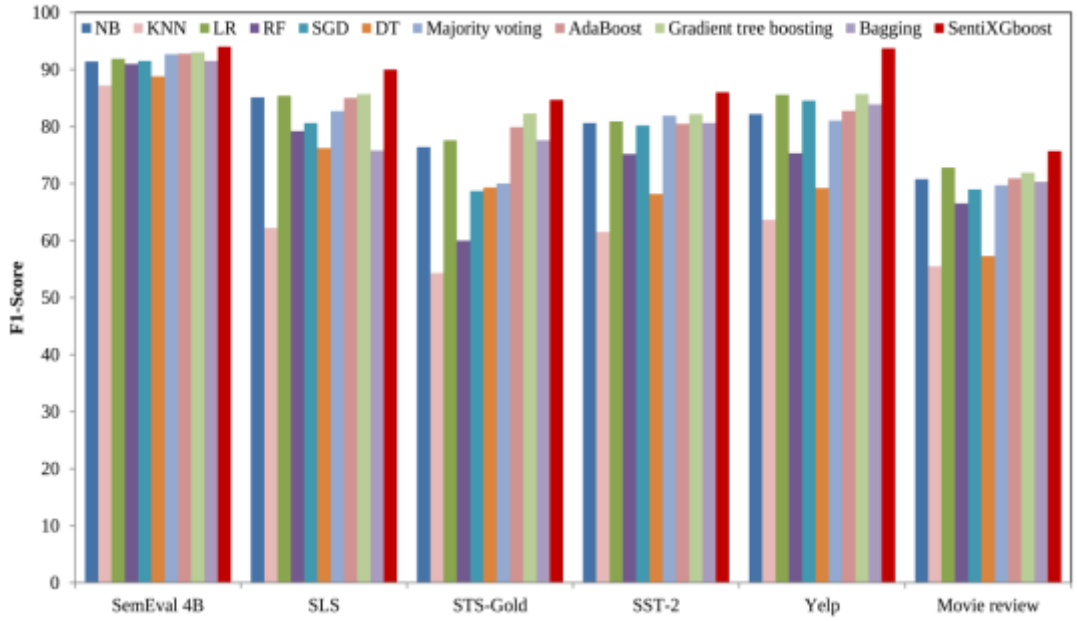


Figure 5.3: Comparison of F_1-Score of single and ensemble classifiers on sentiment labeled datasets.

5.2.3.2 Comparison of Results with Existing Methods

A comparative evaluation of the performance of the SentiXGboost method with other methods on the same datasets was conducted. The datasets include SemEval-2017, Task 4, Subtask B (Sentiment Analysis in Twitter), SLS (Amazon) and STS-Gold, SST-2, Yelp Challenge, and Movie Review (Sentiment Polarity Dataset Version 2.0) datasets. These datasets were employed for comparison since several approaches were proposed as part of the task. To this end, Table 5.6 – Table 5.11 shows that SentiXGboost performs better than all other published methods in terms of Accuracy, Average Recall, and F_1-Score on these datasets.

Table 5.6: Comparison results of our proposed method with other methods based on Accuracy, Average Recall, and F_1-Score for SemEval-2017, Task 4, Subtask B dataset

Studies	**Acc (%)**	**Rec_{macro1} (%)**	**F_1-Score (%)**
Cliche [223]	89	88	89
Symeonidis et al. [26]	60.7	60	60

Rozental and Fleischer [230]	80	82	80
Dovdon and Saias [234]	51.8	59.4	48.6
Wang et al. [231]	49.9	51.6	49.9
Laskari and Sanampudi [235]	41.2	48.3	37.2
Rajendram, and Mirnalinee [232]	51.8	58.6	49.4
Lozić et al. [228]	84	84.5	83.6
González, Pla, and Hurtado [227]	79	76.6	77.3
Korovesis [295]	86.1	80.8	85.4
Proposed SentiXGboost	**90.8**	**88.6**	**94**

[*Acc: Accuracy, Rec$_{macro1}$: Average Recall among positive and negative classes*.]

Table 5.7: Comparison results of our proposed method with other methods based on Accuracy and F_1-Score for SLS dataset

Studies	**Acc (%)**	**F_1-Score (%)**
Chen et al. [296]	88.6	88.4
Huang et al. [297]	88.4	-
Xu et al. [298]	88.2	-
Proposed SentiXGboost	**92.5**	**90**

[*Acc: Accuracy.*]

Table 5.8: Comparison results of our proposed method with other methods based on Accuracy and F_1-Score for STS-Gold dataset

Studies	**Acc (%)**	**F_1-Score (%)**
Saif et al. [299]	81.32	78.56
Kermani et al. [300]	85.92	74
Troussas et al. [301]	89.02	-
Yan et al. [302]	85.35	-
Kauer and Moreira [303]	84.5	84.3
Keshavarz and Abadeh [304]	76.68	-
Proposed SentiXGboost	**91.1**	**84.7**

[*Acc: Accuracy.*]

Table 5.9: Comparison results of our proposed method with other methods based on Accuracy and F_1-Score for SST-2 dataset

Studies	Acc (%)	F_1-Score (%)
Hiyama et al. [242]	73.7	-
Baktha et al. [241]	81.54	-
Chen et al. [240]	82.3	-
Giménez et al. [306]	82.45	-
Xu, Y. et al. [307]	81.8	-
Tripathi, S. et al. [308]	53.3	-
Park and Ahn [309]	80.9	-
Socher et al. [310]	82.4	-
Kim, Y. [311]	82.9	82.4
Sochar et al. [312]	82.7	-
Proposed SentiXGboost	**85.2**	**86**

[*Acc: Accuracy.*]

Table 5.10: Comparison results of our proposed method with other methods based on Accuracy and F_1-Score for Yelp Challenge dataset

Studies	Acc (%)	F_1-Score (%)
Guerreiro and Rita [313]	66.86	-
Potts, C. et al. [314]	-	73.1
Chen, R. [315]	72.83	67.9
Hemalatha and Ramathmika [316]	78.44	-
Rathee, N. et al. [317]	76	-
Ahmed and Ghabayen [318]	-	91
Zhu, Y. et al. [319]	82	-
Proposed SentiXGboost	**92.75**	**93.75**

[*Acc: Accuracy.*]

Table 5.11: Comparison results of our proposed method with other methods based on Accuracy and F_1-Score for Movie Review dataset

Studies	**Acc (%)**	**F_1-Score (%)**
Singh and Sachan [320]	71.3	-
Carvalho, F. et al. [321]	74.65	-
Korovkinas, K. et al [322]	72	70.11
Proposed SentiXGboost	**76.5**	**75.7**

[*Acc: Accuracy.*]

Chapter 6

SENTIGA: OPTIMIZED ENSEMBLE CLASSIFIER FOR SENTIMENT ANALYSIS USING GENETIC ALGORITHM

Ensemble learning is a subfield of machine learning that combines the predictions of multiple learning algorithms to create classification models with better predictive accuracy. It is crucial to identify the base learning algorithms that can accomplish the classification task as part of the ensemble. The choice of the combination scheme for base learning algorithms is also crucial in ensuring higher predictive accuracies. In this chapter, we introduce a novel optimized classifier ensemble framework denoted as SentiGA. This framework identifies the optimal subset of classifiers from a large pool of candidates for the sentiment analysis task. The proposed SentiGA uses a Genetic Algorithm to determine the classifiers that make up the ensemble. The Genetic Algorithm is an optimization technique that provides a range of options for dealing with the complexity between the search algorithm used and the solution found. To ensure the efficiency of the ensemble and achieve good performance, the ensemble classifiers must be constructed with well-performing base classifiers that complement each other. This is because the selection of the base classifiers and their performances influence the final performance of the ensemble classifiers. To obtain a robust ensemble classifier, it is required that the constituent base classifiers are tuned with variations of the parameter settings, and different feature subsets combinations are used for model training. This chapter describes the SentiGA framework to

facilitate the development of a novel ensemble classifier approach for the sentiment analysis task. The remainder of this chapter is structured as follows. The architecture of the proposed SentiGA framework is described in Section 6.1. Additionally, this section contains a general discussion on the different parts that make up the proposed SentiGA framework. We then discuss how the multiple base classifiers were combined by using a voting algorithm. It also consists of a discussion on how the optimized ensemble classifier was generated by applying Genetic Algorithms to further improve the performance of the proposed method. At the end of this chapter, the detailed evaluation of experimental results is compared with the state-of-the-art methodologies are discussed.

6.1 Proposed SentiGA Method Architecture

In this section, we present the detailed architecture of the proposed SentiGA method for sentiment analysis. The architecture of the proposed framework is depicted in Figure 6.1. The SentiGA framework addresses the selection of an optimized ensemble classifier from a pool of classifier ensembles. The SentiGA method makes use of the Genetic Algorithm optimization method to find an optimal solution for sentiment analysis of social media data through the evolution of various classifier ensembles. The classifier ensembles are represented in the SentiGA framework as chromosomes, where each bit denotes a classifier's participation in the ensemble.

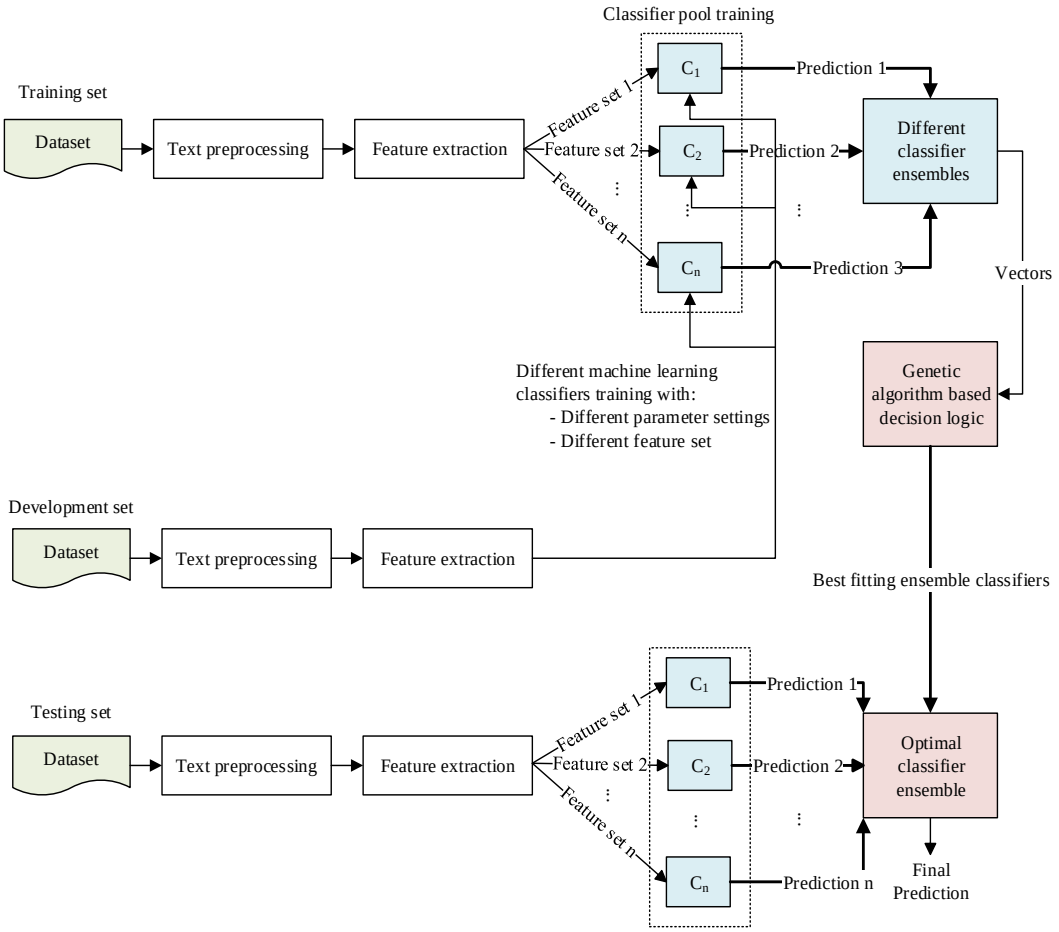


Figure 6.1: A block diagram of the proposed SentiGA framework

The genetic framework employed in this work consists of several steps which are described in the following. At the initial step of the proposed architecture, each sentence in the dataset passes through preprocessing. The data is cleaned, normalized, and made ready for the next steps at the preprocessing stage. There are four main preprocessing operations on the data, including normalization, tokenization, stemming, and removal of stopwords. The details on these techniques are given in section 4.2. Furthermore, during preprocessing, the datasets which have imbalanced distributions among the classes are balanced. For instance, the positive class and negative class vastly outnumber the strongly positive class and strongly negative class. The first step attempts to balance the distribution of the classes in the dataset. This problem is solved by increasing the number of the minority class by

oversampling them. After the preprocessing operations, the next step is feature extraction from the preprocessed dataset. Three techniques of feature extraction are used by the proposed method, and the extracted features are then combined and used as features for training the classifiers. The feature sets include BoW, TF-IDF, and Bigrams. Section 4.3 gives more details on these modules. The data is then split into three sets consisting of 50% training, 20% validation, and 30% testing sets. After this stage, a wide set of base classifiers are used to generate a pool of classifiers. The classifiers deployed in this framework include SVM, NB, LR, SGD, RF, and DT. Details of these classifier training are given in section 4.4. The pool of classifiers consists of twenty-five classifiers in our method, which makes the results more robust. The classifier pool is obtained by tuning various parameter settings and using different feature subsets for training classifiers. All classifiers in the pool are trained on the training set using different feature sets and model parameters, as shown in Table 6.2. The validation dataset is used to evaluate the ability of the trained classifiers to predict the sentiments in tweets accurately. Furthermore, different possible combinations of the classifier are generated from classifiers in the pool, using the weighted majority voting rule (this module is explained in section 4.5.4). Each base classifier with the corresponding classification accuracy in the validation dataset is selected as a weight. The Genetic Algorithm is employed to generate or evolve the optimum classifier ensemble from a large pool of classifiers at the optimized ensemble classifiers step. The general principle of Genetic Algorithms is explained in section 2.6.3. The concept of Genetic Algorithm concerning the proposed SentiGA framework is described below:

1) **Initial population creation:** at this step of the Genetic Algorithm, we initialize the random base population consisting of objects of a different

combination of machine learning models, i.e., SVM, NB, LR, RF, etc. For each classifier model, we create its instances for all possible combinations of its parameters and then add them to the initial collection of the base classifiers. Next, a genetic code, or chromosome, is designed to represent a combination of classifiers. This genetic code is a bit string whose length corresponds to the number of base classifiers in the collection mentioned previously, and each bit of the string, or gene, corresponds to one of the base classifiers. In a chromosome, the indexes of the base classifiers used in the classifier ensemble are set to 1, and the indexes of the classifiers that will not participate are set to 0. For example, the encoding of chromosomes for the object comprising SVM2 and RF is shown in Figure 6.2. For each chromosome, or classifier ensemble, in the population, the predictions of the base classifiers (marked by 1 in the chromosome) are combined using weighted majority voting and then its fitness is calculated by computing its accuracy. The highest accuracy is regarded as the fitness level.

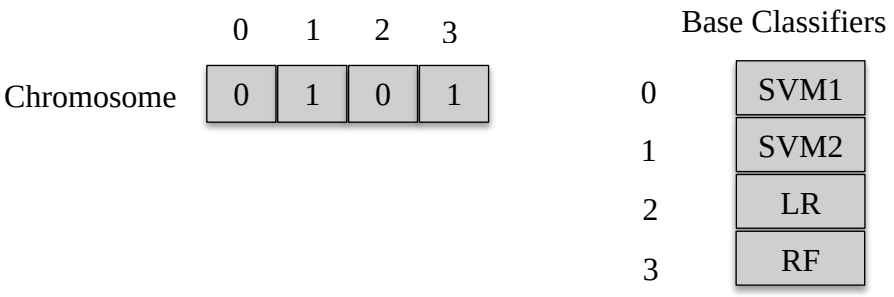


Figure 6.2: The encoding of a chromosome

2) **Next Population:** from the initial population, a new population is reproduced for the next generation through genetic procedures of

selection, *crossover,* and *mutation*. This process will continue until a stopping condition is reached or the solution converges at an optimal solution when the operations terminate. The process of genetic operations is terminated as follows:

- ***Selection*:** The algorithm first iterates through the population and finds the chromosome with maximum fitness in the selection process. Then it applies the *accept-reject* algorithm for selection. In this technique, a random chromosome from the population is selected after which another random number is selected ranging from 0 to maximum fitness. If this second random number is less than the index of the randomly selected object from the population then it will be accepted otherwise rejected. Thus, there will be a high probability for the chromosome with maximum fitness to be selected as a parent for the next generation.
- ***Crossover*:** the two selected parents pass through the genetic function of crossover. In the crossover, a new child has produced then a random index in the chromosome is selected as the mid-point. The genes from parent *A* up to the mid-point and the genes from parent *B* onward from the mid-point are combined to form the chromosome of the new child.
- ***Mutation*:** in the mutation, the genes of the chromosome are randomly altered based on the mutation rate. If a randomly generated number is less than the mutation rate, a gene at a particular position is replaced with another random bit to produce some diversity.

- ***Genotype or Decoding*:** the list of newly created chromosomes is further decoded to produce genotypes of the offspring. In this step, an ensemble classifier is created as per the genetic codes of the newly produced offspring. Suppose a bit at a particular index in the chromosome is 1. In that case, the corresponding base classifier is added to the estimator's list for ensembling. For the ensemble, the weighted majority-based voting rule is used.

3) **Replace:** the population is replaced with the newly generated population.
4) **Elite Count**: some chromosomes with the best fitness values in the current generation are guaranteed to survive to the next generation. These chromosomes are called *elite children*.
5) **Fitness:** the fitness for each object of the new population is calculated. Different matrices can be used to find the fitness of a classifier e.g., accuracy, precision, recall, and F_1-score, which are commonly used for classification. In this study, accuracy is used.
6) **Test:** terminate the loop if the stopping condition is reached, and return the best solution.
7) **Loop:** Go to step *5*

The flowchart of the proposed architecture and the pseudo-code for using Genetic Algorithm in the proposed SentiGA method is given below:

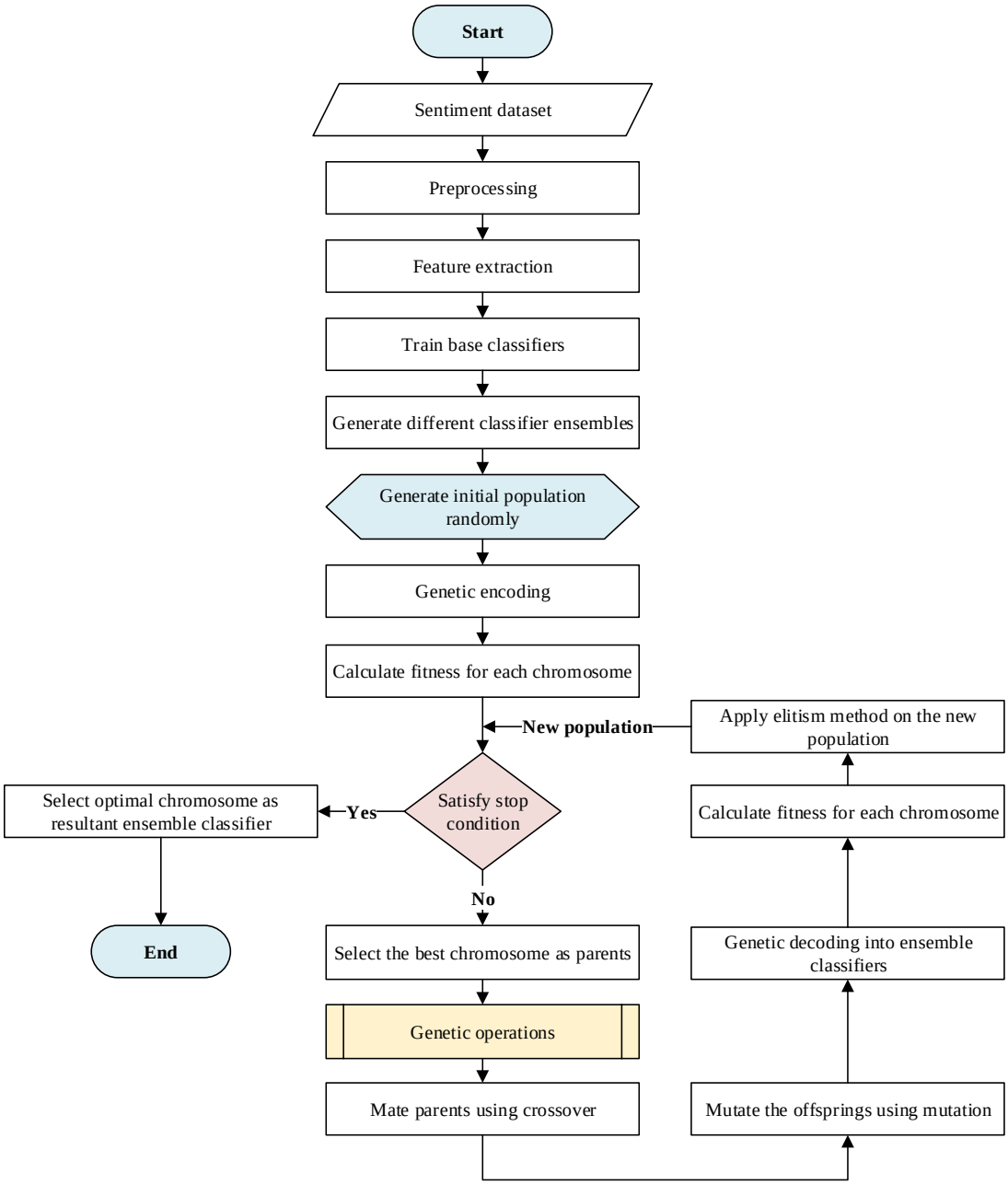


Figure 6.3: The flowchart of the proposed SentiGA scheme

Algorithm: Ensemble Classifier Optimization using Genetic Algorithm for Sentiment Analysis

Require: (Sentiment Data)

Ensure: Optimal Performance with Ensemble Classifier

%Step#1 Preprocessing

[Data] ← PreProcess (Sentiment Data)

%Step#2 Feature Extraction

Splitting into training, validation and testing data ([X_train], [X_val], [X_test], [y_train], [y_val], [y_test]) ← featureExtraction (extractor, [Data])

%Step#3 Classifier Pool Generation

Train base classifiers Ci ← using different parameter settings and different combination of feature set

[classifiers] ← classifier ([X_train], [y_train])

%Step#4 Ensembling Classifiers

[classifiers] ← many combinations of predictions [ensemble classifiers]

%Step#5 Initial Population

[ensemble classifiers] ← [ensemble classifiers] ([X_train], [y_train])

% Encoding

For each cls **ϵ** ensemble classifiers **do**

[baseChromosome] ← getChromosome ([ensemble classifiers], indexes)

End For

```
    % Population Object
    population ← Population ([ensemble classifiers],
    [baseChromosome])
    % Initial Fitness
    population → calculateFitness ()
    return population
%Step#6 Next Population
For each gen ϵ generation do
    % Step#7 test
    If Stopping condition
        Break and return optimalPerformance
    End If
    For each classifier ϵ population do
        % Selection
        PartnerA ← acceptReject(maxFitness)
        PartnerB ← acceptReject(maxFitness)
        % Crossover
        Child ← PartnerA → crossover(PartnerB)
        % Mutation
        Child → mutate (mutationRate)
        %OffspringsChromosome
        [OffspringsChromosome] ←[Child]
    End For
    % Decoding or Genotype
    For each item ϵ population do
```

%Ensemble Classifiers

[Offsprings] ← getPhenoType

([OffspringsChromosome], ensemble classifiers)

End For

% Replace

population → chromosome ←[OffspringsChromosome]

population → ensemble classifiers ← [Offsprings]

% Step#8 Elitism

keep best chromosome → population

% Step#9 Fitness

Population → calculateFitness ( )

End For

6.2 Methods

Let it be recalled that the experimental setup used for the proposed SentiGA method was discussed in Chapter 4. The datasets, the steps for data preprocessing (normalization, tokenization, removal of stopwords, and stemming), the feature extractions, the base classifiers, and ensemble learning classifiers used in this framework have been discussed in detail in Chapter 4. Based on the feature extraction methods described in section 4.2, we constructed four feature sets based on the three different feature extractor categories as defined in Table 6.1. For instance, subset A consists of the combinations of three different feature categories.

Table 6.1: The set of features used for training base classifiers

Feature sets	BoW	TF-IDF	Bigram
A	X	X	X
B	X	X	-
C	-	X	X
D	X	-	X

6.2.1 Classifier Pool Generation

Six base classifiers from the previous experimental setup (Chapter 4) were selected for use in this experiment to generate a pool of classifiers: SVM, NB, LR, SGD, RF, and DT. These classifiers were selected because most of them performed well in terms of Accuracy, as demonstrated in the previous chapter. Table 6.2 shows the details of these base classifiers and their parameter settings. The step on classifier pool generation aims to generate a pool of candidate base classifiers comprising N classifiers that are both accurate and diverse. The classifier's diversity is the main component in the optimized classifier ensemble method because it is impossible to enhance the predictive performance of classification when combining classifiers with the same outputs. The diversity of classifiers can be achieved by using different classifier algorithms and for a given classifier algorithm by training it with different parameter settings and different feature set combinations. Consequently, each base classifier is trained using different parameter values to ensure the uniqueness of each classifier in at least one of the parameter properties in the pool. For instance, the SVM classifier can be trained using different values of parameter settings for the degree of the polynomial kernel, linear kernel, and radial basis function (RBF). Three different feature engineering methods which are frequently used for sentiment analysis are considered in this study. These feature engineering methods include BoW, TF-IDF, and Bigram, which are used in different feature combinations to train

the base classifiers as illustrated in Table 6.2. For example, classifiers SVM1 and SVM2 belong to the same classifier type but they use a different combination of feature sets. Similarly, classifiers SVM8 and NB are different classifiers but they use the same combination of feature types.

Table 6.2: Presenting the complete detail on the base classifiers and their parameter settings with feature set engineering methods used for training the base classifiers

No.	Classifiers	Parameter Settings	Feature subsets
1	SVM1	kernel=set(['linear']), degree=set([3]), gamma=set(['auto'])	A
2	SVM2	kernel=set(['linear']), degree=set([3]), gamma=set(['auto'])	D
3	SVM3	kernel=set(['linear']), degree=set([8]), gamma=set(['auto'])	C
4	SVM4	kernel=set(['linear']), degree=set([8]), gamma=set([' scale'])	A
5	SVM5	kernel=set(['linear']), degree=set([8]), gamma=set([' scale'])	B
6	SVM6	kernel=set(['rbf']), degree=set([3]), gamma=set(['auto'])	A
7	SVM7	kernel=set(['rbf']), degree=set([8]), gamma=set(['auto'])	D
8	SVM8	kernel=set(['rbf']), degree=set([3]), gamma=set(['scale'])	C
9	SVM9	kernel=set(['rbf']), degree=set([3]), gamma=set(['scale'])	B
10	NB1	-	A
11	NB2	-	B
12	LR1	penalty = set(['l2']), random_state=set([0])	A
13	LR2	penalty = set(['l2']), random_state=set([1])	D
14	LR3	penalty = set(['l2']), random_state=set([2])	B
15	LR4	penalty = set(['l2']), ,random_state=set([2])	A
16	RF1	n_estimators=set([100]), max_depth=set([3)], random_state=set([0])	A
17	RF2	n_estimators=set([100]), max_depth=set([5]), random_state=set([1])	D
18	RF3	n_estimators=set([200]), ıax_depth=set([3]), ındom_state=set([0])	A

19	RF4	n_estimators=set([200]), max_depth=set([5]), random_state=set([1])	D
20	SGD1	loss=set(['log']), penalty=set(['l2']), max_iter=set([3])	A
21	SGD2	loss=set(['log']), penalty=set(['l2']), max_iter=set([5])	D
22	SGD3	loss=set(['hinge']), penalty=set(['l2']), max_iter=set([5])	C
23	SGD4	loss=set(['hinge']), penalty=set(['l2']), max_iter=set([8])	B
24	DT1	n_estimators=set([100]), max_depth=set([3])	A
25	DT2	n_estimators=set([200]), max_depth=set([5])	A

6.3 Experimental Results and Evaluation

Evaluation is important to assess the credibility of the framework and ascertain the possible improvements required. This section presents a detailed analysis and evaluation of results obtained from the experiments with the proposed SentiGA method. Two sets of experiments were conducted in this study to assess the SentiGA method's feasibility and performance. The results of both experiments are presented and discussed to show that the SentiGA method is significantly more efficient than the single best classifier and the ensemble classifier containing all classifiers in the pool. In addition, this section presents the development environment including the datasets used, experimental setup, and evaluation experimental results. The performance of the SentiGA method compared to some existing sentiment analysis methods using the same datasets is presented.

6.3.1 Datasets Used

In this section, we introduce the datasets used in our experiments. For training and testing the proposed SentiGA method and evaluation of the results, five publicly available Twitter sentiment datasets are used. These include SemEval 2017 Task (4A, 4B and 4C), SST-2 and SST-5 datasets. More details on these datasets are described in section 4.1. Each dataset is split into three partitions: training, development, and test data sets. The number of each class sample in each partition across the datasets used in the experiments is shown in Table 6.3.

Table 6.3: Statistics on employed datasets

Datasets	Type	SP	P	Neu	N	SN	Total
SemEval-2017 Task 4A	Train	-	3972	5790	1791	-	20632
	Development	-	988	1452	449	-	
	Test	-	2099	3100	991	-	
SemEval-2017 Task 4B	Train	-	4572	-	1336	-	10551
	Development	-	1164	-	313	-	
	Test	-	2476	-	690	-	
SemEval-2017 Task 4C	Train	205	4383	5632	1259	74	20632
	Development	54	1101	1409	308	17	
	Test	123	2346	3040	634	47	
SST-2	Train	-	2759	-	2623	-	9612
	Development	-	679	-	667	-	
	Test	-	1525	-	1359	-	
SST-5	Train	1072	1717	817	1271	1761	11855
	Development	238	436	217	307	462	
	Test	542	958	476	664	917	

[*SP: Number of total strongly positive tweets in the data set, P: Number of total positive tweets in the data set, Neu: Number of total neutral tweets in the data set, N: Number of tweets belonging to negative tweets in the datasets, SN: Number of tweets belonging to strongly negative tweets in the datasets.*]

6.3.2 Experimental Procedures

In this section we provide details about experimental settings for two series of experiments: (i) the ones concerned with the selection of the optimal subset of classifiers from the large pool of classifiers using Genetic Algorithm; and selection of the single best classifier in the pool, and (ii) the second assessment of the

proposed optimized classifier ensemble with full ensemble classifiers containing all classifiers in the pool.

In the experiments, five different datasets are used for evaluating the proposed SentiGA method. The characteristics of these datasets are presented in Table 6.3. The experimental datasets cover two binary class problems, one ternary class problem, and two 5-point class problems. The experimental workbench is Jupyter Notebook, a common group of machine learning software written in Python which supports a wide range of workflows in machine learning and data mining tasks. In the beginning, the datasets are split into three disjunctive groups: 50% training set, 20% validation set, and 30% testing set. In this framework, six different classification models are used to generate ensemble classifiers. Each classification algorithm is trained with different parameter settings and different combinations of feature sets to generate multiple diverse classifiers as base classifiers. Table 6.2 presents the settings used in this experiment. The weighted majority voting method is used in this framework to compute the performance of classifier ensembles to calculate the fitness of chromosomes. Each chromosome is designed to represent the different classifier ensembles. The initial population of chromosomes in the Genetic Algorithm is generated randomly. The population size in the simulation experiments is set to 100 chromosomes; each of them is represented by bit strings of length 25, containing the voting bits. This implies that a hundred different ensemble classifiers evolve at the same time. The algorithm is run for 1000 iterations. The accuracy performance given by the weighted majority voting from the combination rule is used to determine the fitness of each chromosome. The number of generations for the Genetic Algorithm evolution is set at 100. In every new generation, the Tournament

Selection method is employed to select the pair of chromosomes with the highest fitness values from a randomly selected subset of the population. After selecting the pairs of chromosomes, they are then passed through the crossover (mid-point) and mutation processes at a rate of 0.5 and 0.1 respectively. These two processes are carried out to increase the population's diversity, thus, increasing the chances of preventing a convergence to the local optimum. In this framework, the Elitism method is employed where 10% of the best chromosomes from the previous generation are propagated to the new generation, which is not already present in the new generation. This method can increase the performance of the Genetic Algorithm rapidly because it avoids losing the fittest chromosomes over the entire population. After a series of evolution and many generations when the termination condition is met, the population's fittest chromosome is considered as the best-optimized ensemble classifier solution.

6.3.3 Results and Discussion

This section presents the results and discusses the classification performance of the SentiGA proposed method in terms of Accuracy, Precision, Recall, Average Recall, F_1-Score, Average F_1-Score, Macro-average Mean Absolute Error, and Micro-average Mean Absolute Error. The description of these metrics is provided in section 4.6. These performance measures are also commonly used to assess sentiment analysis models. Furthermore, the results of the SentiGA method with the other state-of-the-art methods that used the same dataset are compared.

6.3.3.1 Experimental Results Evaluation

To assess the efficiency of the proposed SentiGA schemes, the experiments were conducted on five widely used datasets in sentiment analysis. Table 6.4 – 6.8 depicts the comparison results of the best single classifier, full ensemble of all classifiers as

well as the proposed SentiGA method. The evaluation results are reported in Table 6.4 – 6.8 to demonstrate the effectiveness of the proposed SentiGA method over other existing methods. It should be pointed out that the weighted voting technique was used for all the cases. The best results obtained by a specific classifier with the datasets appear in boldface, while the second-best results appear in italics.

Table 6.4: Classification results obtained by the single best classifier, full ensemble containing all classifiers, and proposed SentiGA method with the name of selected classifiers for SemEval-2017, Task 4A (Ternary) dataset

Classification Scheme	**Candidate Models**	**Acc%**	**Rec_{macro2}%**	**$F_{1\text{-}macro}$%**
Best Single Classifier	LR3	64.27	58.82	60.25
Full Ensemble Classifiers	Combined all classifiers in the pool	*65.4*	*59.67*	*61.2*
Proposed SentiGA	SVM2, NB1, LR3, DT1	**76.4**	**76.6**	**75.7**

[*Acc: Accuracy,* Rec_{macro2}*: Average Recall among Positive, Neutral, and Negative classes,* $F_{1\text{-}macro}$*: Average* F_1*-Score among Positive and Negative classes*.]

Table 6.4 depicts the comparison results of the best-fitting ensemble classifiers formed using the Genetic Algorithm scheme. The results show that the proposed method performs better than both the best single classifier as well as the ensemble of all classifiers in terms of Accuracy, Average Recall, and Average F_1-Score with values 76.4%, 76.6%, and 75.7% respectively. Furthermore, the full ensemble classifier achieves the second-highest performance. This shows that the full ensemble classifier technique is effective in improving the performance of the single classifiers. The proposed SentiGA method improves the second-best performance by 11%, 19.93%, and 14.5% respectively. Moreover, the proposed optimized ensemble classifier selected the chromosome [01000000010001000000000010] which comprises four different classifiers (SVM2, NB1, DT1, and LR3) as an optimal subset of classifiers to classify the test set of SemEval-2017, Task 4A dataset. The

best set of classifiers obtained is combined using the weighted majority voting rule. As seen in Table 6.4, the LR3 classifier is the best classifier in the pool of classifiers which yields the highest performance in terms of Accuracy, Average Recall, and Average F_1-Score with values of 64.27%, 58.82%, and 60.25% respectively. It is evident that the Genetic Algorithm is efficient in selecting the best performing classifiers to classify unseen data.

Table 6.5: Classification results obtained by the single best classifier, full ensemble containing all classifiers, and proposed SentiGA method with the name of selected classifiers for SemEval-2017, Task 4B (Binary) dataset

Classification Scheme	**Candidate Models**	**Acc%**	**Pre%**	**Rec_{macro1}%**	**F_1-Score%**
Best Single Classifier	SVM1	87.2	*91.1*	*79.32*	*91.5*
Full Ensemble	Combined all classifiers in the pool	*88*	88.8	76.63	90.67
Proposed SentiGA	SVM1, SVM2, NB1, LR2, RF1 and SGD1	**94.3**	**96.83**	**94.22**	**95.51**

[*Acc: Accuracy, Pre: Precision, Rec$_{macro1}$: Average Recall among Positive and Negative classes.*]

Based on the experiment results in Table 6.5, the greatest predictive performance for SemEval-2017, Task 4B dataset was obtained with the proposed SentiGA method with the values of 94.3%, 96.83%, 94.22%, and 95.51% respectively. According to the results achieved, SVM1 is a better base classifier in the pool compared with all other base classifiers. Notable, it is evident that the proposed method improves the performance of the second-best algorithm by 6.3%, 5.73%, 14.9%, and 4.01% respectively. As could be seen in Table 6.5, consider that the proposed SentiGA method selected the chromosome [110000000100100100010000 0]. This means the selected classifiers comprise six different classifiers (SVM1, SVM2, NB1, LR2, RF1, and SGD1) from the pool of classifiers presented in Table 6.3. This represents

the best subset of classifiers to classify the testing set of SemEval-2017, Task 4B dataset. This indicates that the Genetic Algorithm is successful in selecting the best performing classifiers in the pool because the SVM1 which is the best single classifier yields the highest classification performance.

Table 6.6: Classification results obtained by the single best classifier, full ensemble containing all classifiers, and proposed SentiGA method with the name of selected classifiers for SemEval-2017, Task 4C (Five-point) dataset

Classification Scheme	**Candidate Models**	**Acc%**	**F_1-Score%**	**MAE^M**	**MAE^μ**
Best Single Classifier	SGD1	77.67	76.04	*0.83*	*1.07*
Full Ensemble	Combined all classifiers in the pool	*80.8*	*80.1*	0.903	1.134
Proposed SentiGA	SVM1, SVM2, LR1, SGD1 and DT2	**85.51**	**85.1**	**0.154**	**0.323**

[*Acc: Accuracy, MAE^M: Macro-average Mean Absolute Error, MAE^μ: Micro-average Mean Absolute Error*.]

Table 6.6 depicts Accuracy, F_1-Score, Average F_1-Score, Macro-average Mean Absolute Error, and Micro-average Mean Absolute Error obtained by the best-performing classifier, the full ensemble of classifiers, and the proposed SentiGA method with the selected classifiers. According to these results in Table 6.6, it could be observed that the proposed SentiGA method surpasses both the best single classifier and the ensemble of all classifiers. It is to be noted that the highest predictive accuracies in terms of Accuracy and F_1-Score, and the lowest predictive performances in terms of Macro-average Mean Absolute Error and Micro-average Mean Absolute Error is achieved with the proposed method, which yielded 85.51%, 85.1%, 0.154, and 0.323, respectively. Additionally, the chromosome [1100000000010000000100001] is selected by SentiGA as an optimized ensemble classifier which contains only five classifiers out of twenty-five classifiers including

SVM1, SVM2, LR1, SGD1, and DT2. This indicates the significance of selecting an optimum subset of the classifiers in the classifier set. The second-best predictive performance was obtained using the full ensemble classifiers method with 80.8% and 80.1% in terms of Accuracy and F_1-Score, respectively. Whereas the second-best performance in classification was obtained with the single best-performing classifier SGD1 with the values of 0.83% and 1.07% in terms of Macro-average Mean Absolute Error and Micro-average Mean Absolute Error respectively. It should be noticed that the full ensemble classifier technique provides higher Macro-average Mean Absolute Error and Micro-average Mean Absolute Error values when compared to the Accuracy and F_1-Score values. This is reasonable since the objective function considered during optimization is the Accuracy or F_1-Score value.

Table 6.7: Classification results obtained by the single best classifier, full ensemble containing all classifiers, and proposed SentiGA method with the name of selected classifiers for SST-2 (Binary) dataset

Classification Scheme	**Candidate Models**	**Acc%**	**Pre%**	**Rec%**	**F_1-Score%**
Best Classifier	NB1	78.97	78.77	*79.82*	79.29
Full Ensemble	Combined all classifiers in the pool	*79.54*	*82.17*	78.29	*80.18*
Proposed SentiGA	SVM2, SVM3, SVM7, SVM8, NB1, LR1, LR3, RF4 and SGD1	**86.6**	**84.2**	**83.3**	**83.74**

[*Acc: Accuracy, Pre: Precision, Rec: Recall.*]

Table 6.7 displays the results achieved by each single best model; full ensemble weighted voting scheme and the proposed optimized ensemble scheme with the SST-2 dataset in terms of Accuracy, Precision, Recall, and F_1-Score. As could be observed from the results, the highest predictive performance was obtained with the proposed SentiGA scheme with the values 86.6%, 84.2%, 83.3%, and 83.74,

respectively. The full ensemble weighted voting scheme was ranked as the second-best performance in terms of Accuracy, Precision, and F_1-Score, which scored 79.54%, 82.17, and 80.18%, respectively. Meanwhile, the second-best performance in terms of Recall was achieved individually by the best performing NB1 classifier was 79.82%. Regarding results presented in Table 6.7, it is clear that the proposed SentiGA scheme performs better than the results obtained by other algorithms. This is an improvement on the second-best performance in terms of Accuracy, Precision, Recall, and F_1-Score by 7.06%, 2.03%, 3.48%, and 3.56%, respectively. The results indicate the significance of the Genetic Algorithm in selecting the well-performing ensemble classifier generally. Additionally, the proposed SentiGA classification scheme selected the chromosome [011000110101010000011000000], which comprises nine classifiers out of twenty-five classifiers, including SVM2, SVM3, SVM7, SVM8, NB1, LR1, LR3, RF4, and SGD1. It can be further observed that classifiers exploit rich feature sets in the classifier set to provide better recognition performances to be selected during the optimization process.

Table 6.8: Classification results obtained by the single best classifier, full ensemble containing all classifiers, and proposed SentiGA method with the name of selected classifiers SST-5 (5-point) dataset

Classification Scheme	**Candidate Models**	**Acc%**	**Pre%**	**Rec%**	**F_1-Score%**
Best Single Classifier	SVM2, SVM3	53.95	*53.62*	53.79	*53.47*
Full Ensemble	Combined all classifiers in the pool	*54.64*	53.17	*54.75*	53.36
Proposed SentiGA	SVM2, SVM3, SVM6, RF3 and SGD1	**62.94**	**62.2**	**62.5**	**62.5**

[*Acc: Accuracy, Pre: Precision, Rec: Recall.*]

Table 6.8 illustrates the performance achieved by the proposed SentiGA method, best single model, and combined all classifiers method in terms of Accuracy, Precision, Recall, and F_1-Score for SST-5 five-point scale dataset. It can be observed that the best classification performance is achieved by the proposed method in all metrics, which are 62.94%, 62.2%, 62.5%, and 62.5%, respectively. The second-best performance yielded an Accuracy of 54.64% and Recall of 54.75% by the full ensemble classifier method. The second-best Precision score of 53.62% and F_1-Score of 53.47% is achieved by both the single classifiers, SVM2 and SVM3. These results show that the proposed optimized ensemble classifier method has been effective with a significant performance improvement. As shown in Table 6.8 the proposed SentiGA method selected the chromosome [01100100000000000001100000] which compromises (SVM2, SVM3, SVM6, RF3, and SGD1) classifiers as optimal subsets of classifier candidates. These are then used in the final ensemble construction to classify the testing set of the SST-5 dataset. This indicates the significance of the SentiGA method in selecting an optimal subset of classifiers. This is because the optimal subset of classifiers contains the best performing single classifiers (SVM2 and SVM3).

In summary, the experimental results clearly show that the highest predictive performances in all terms were generally achieved by the proposed SentiGA classifier method based on the Genetic Algorithm. Concerning the simulation experiments, we found that the proposed method shows a significant improvement compared to individual classifiers as well as the full weighted voting ensemble classifier method across all sentiment datasets. This study has successfully shown the significance of selecting an optimal subset of the classifiers in the classifier set based

on the concept of the Genetic Algorithm. Furthermore, we observed that the SVM classifier significantly performs better than the other single classifiers in the pool. Additionally, the SVM classifier - specifically the SVM2 – with settings (Kernal = "Linear", trained with combined BoW and Bigram features) has more contributions. Conversely, the DT classifier has the least contributions in our proposed scheme as compared to other single classifiers in the pool.

6.3.3.2 Comparison of Results with Related Works

To further evaluate the effectiveness of the proposed SentiGA method, we provide a set of comparative results of the SentiGA method against some other relevant works with the chosen performance metrics Accuracy, Precision, Recall, Average Recall, F_1-Score, Average F_1-Score, Macro-average Mean Absolute Error, and Micro-average Mean Absolute Error. The comparative analysis is based on results obtained using the proposed SentiGA method with those of other methods in the literature using SemEval 2017 Task (4A, 4B and 4C), SST-2 and SST-5 datasets are presented in Table 6.9 and Table 6.10. The results of the proposed methods are tabulated based on the performance of the SentiGA method compared to other related existing methods using the same datasets.

Table 6.9: Comparison results of the proposed optimized method with related work methods on SemEval-2017 Task 4 (A, B, and C) datasets

Studies	**Tasks**	**Acc %**	**Rec_{macro} %**	**F_1-Score%**	**F_{1macro} %**	**MAE^M**	**MAE^μ**	**References**
Cliche, M.	4A	65.8	68.1	-	68.5	-	-	[223]
	4B	89.7	89	88.2	-	-	-	
	4C	-	-	-	-	0.481	0.554	
Baziotis, C. et al	4A	65.1	68.1	-	67.7	-	-	[222]
	4B	86.9	86.1	85.6	-	-	-	
	4C	-	-	-	-	0.555	0.543	
Kolovou, A. et al.	4A	65.9	64.8	-	64.8	-	-	[224]
	4B	86.3	85.6	85.4	-	-	-	
	4C	-	-	-	-	0.623	0.734	
Hama Aziz &	4A	-	-	-	-	-	-	[323]
	4B	90.8	88.6	94	-	-	-	

Dimililer	4C	-	-	-	-	-	-	
Proposed SentiGA	4A	**76.4**	**76.6**	-	**75.7**	-	-	-
	4B	**94.3**	**94.22**	**95.51**	-	-	-	
	4C	-	-	-	-	**0.154**	**0.323**	

[*Acc: Accuracy, Rec_{macro}: Average Recall, F_{1macro}: Average F_1-Score among positive and negative classes, MAE^M: Macro-average Mean Absolute Error, MAE^μ: Micro-average Mean Absolute Error*.]

Table 6.9 presents the comparative results of the proposed SentiGA method with its competitors on the SemEval-2017 Task 4A, 4B, and 4C datasets. We observe that our proposed optimized ensemble classifier method performs convincingly better compared to these existing methods. In the SemEval-2017 shared task, the top three systems employed CNN, LSTM, and Neural Networks. The best performing system for Task 4A, 4B, and 4C was developed by Cliché [223] based on CNN and LSTM. This comparison shows that our proposed method outperforms the best-reported results on all tasks 4A, 4B, and 4C of SemEval-2017 datasets. The results show that our proposed method surpasses the performance of the highest-ranking system by 10.6%, 8.5%, and 7.2% in terms of Accuracy, Average Recall, Average F_1-Score respectively for SemEval-2017 Task 4A 3-point scale classification. Further, our method surpasses the best existing related method by 4.6%, 5.22%, and 7.31% in terms of Accuracy, Average Recall, and F_1-Score respectively for SemEval-2017 Task 4B 2-point scale classification. Similarly, our method outperforms the best existing related method by 0.327and 0.231 respectively in terms of MAE^M and MAE^μ (a lower value is better) for SemEval-2017 Task 4C 5-point scale classification.

Table 6.10: Comparison of the accuracy results of proposed SentiGA method with related work methods on SST-2 and SST-5 datasets

Studies	SST-2	SST-5	References
Tripathi, S. et al.	53.3	-	[308]
Lei, Z. et al.	-	49.7	[236]
Sadr, H. et al.	-	53.42	[239]
Hiyama, Y. et al	73.7	-	[242]
Hassan, A. et al.	-	47.5	[244]
Dong, Y. et al.	-	48.34	[245]
Baktha, K. et al.	81.54	44.61	[241]
Chen, T. et al.	82.3	50.6	[240]
Li, W. et al.	-	50.68	[243]
Lu, Y. et al.	-	47.6	[238]
Giménez, M. et al.	82.45	-	[306]
Sadr, H. et al.	-	51.31	[324]
Kasri, M. et al.	-	48.7	[325]
Xu, Y. et al.	81.8	-	[307]
Park and Ahn	80.9	-	[309]
Socher et al.	82.4	-	[310]
Kim, Y.	82.9	-	[311]
Sochar et al.	82.7	-	[312]
Hama Aziz and Dimililer	85.2	-	[323]
Proposed SentiGA	**86.6**	**62.94**	**-**

In Table 6.10, the accuracy results of the proposed method compared to some other state-of-the-art systems on SST-2 (2-point scale) and SST-5 (5-point scale) datasets are presented. We observe that the proposed SentiGA method yields better accuracies compared to these existing methods on both datasets. In this comparison, it is shown that the accuracy of our proposed method is 1.4% better than the best results on the SST-2 dataset and 9.52% better than the existing method on the SST-5 dataset. The main reasons for this improved performance include using different classification techniques and incorporating the Genetic Algorithm technique in the framework. It should be observed that our proposed method allows the best well-performing classifiers in the system to contribute classifying unseen data as

compared to the existing methods. Hence, the proposed method is less complex compared to the others.

Chapter 7

CONCLUSION AND FUTURE WORK

Sentiment analysis is a sub-field in NLP and has a wide range of applications, including news analysis, marketing, question answering, and knowledge bases. One of the challenges of sentiment analysis is how to develop algorithms that enable the machine to mimic humans in understanding texts. Getting important insights from opinions expressed on the internet, especially from social media blogs, is vital for many companies and institutions. This is because such insights provide an opportunity to get feedback on products, public mood, or investors' opinions. To improve the performance of sentiment classification models, individual models have been combined into single ensemble classifiers for use in different areas such as social media. Therefore, this thesis proposed two novel frameworks for sentiment analysis tasks using machine learning approaches and Genetic Algorithms. To achieve this, we adopted the SentiXGboost ensemble classifier for sentiment classification. Six widely used sentiment datasets were used to test the performance of the proposed method. The datasets include SemEval-2017, Task 4, Subtask B, SLS (Amazon), STS-Gold, SST-2, Yelp Challenge, and Movie Review (Sentiment Polarity Dataset Version 2.0). The results showed that the proposed SentiXGboost ensembling scheme could improve performance when used with traditional approaches such as majority voting, AdaBoost, Gradient Tree Boosting, and Bagging. The reported results confirmed the applicability and effectiveness of the proposed SentiXGboost on all sentiment-labeled datasets.

We introduced a novel and effective optimized ensemble classifier scheme named SentiGA for binary, ternary, and fine-grained sentiment analysis tasks in the second approach. The Genetic Algorithm scheme was applied to identify an optimal subset of classifiers used as base classifiers in the optimized ensemble classifier. The proposed method involves the following as base classifiers: SVM, LR, NB, SGD, RF, and DT. Each classifier was trained with different parameter settings and different combinations of feature sets to produce a pool of classifiers. In the initial step, the weighted majority voting rule was used in the proposed scheme to produce several possible classifier ensembles. This produced a set of solutions; each of which represents a particular classifier combination in the initial population of the Genetic Algorithm. Based on a set of certain criteria, the most promising solution was selected from the final population by the proposed scheme. The experiments were conducted on two binary, one ternary, and two fine-grained sentiment datasets. The results indicate that the proposed SentiXG method yielded better performance for the sentiment analysis task compared to the individual performance of the best base classifier. The results indicate that the proposed method is very effective and has a reasonably high performance in all settings.

7.1 Thesis Contributions

The main contributions of this thesis to knowledge and practice are evident in the two novel ensemble classifier approaches developed for sentiment analysis. These approaches have shown an improved performance in the classification of the polarity of tweets from Twitter. Point-by-point elaborations of the contributions of this research work to knowledge are highlighted below:

- An ML-based system that uses XGBoost as a meta-classifier for stacked ensembling was developed to predict the polarity of sentiments across datasets related to sentiment analysis.
- Further, a novel optimized ensemble classifier scheme that utilizes the Genetic Algorithm to select the best models from a group consisting of many models is proposed. The weighted majority voting rule is then employed to combine the individual models. The pool of classifiers is trained using different feature sets and tuned with different parameters on training data.
- The performance analysis of the results shows that the proposed approach outperforms all the individual classifiers and the traditional ensemble learning approaches.
- To the best of our knowledge, the performance of the proposed methods in sentiment classification has exceeded the performance of similar existing methods of sentiment analysis.

7.2 Limitations

The experiments conducted in this research were limited to the classification of the polarity of tweets on Twitter. The backbone of the proposed approach includes supervised machine learning algorithms and Evolutionary Search algorithms. Even though the accuracy and reliability of the proposed systems have been determined to be good, there is still some room for improvement. Particularly, the predictive accuracy of the classification scheme can be improved in future studies. Other limitations of the research are highlighted as follows:

- The classification process of the SentiXGboost method uses the combination of all feature sets at once to train the classifiers. The

consequence is that the results may not be able to vary for each model since not all possible different combinations of feature sets have been tested. It is important to test and update our models using all the possible combinations of feature sets.

- The number of sentiment terms in this research is limited, and this has the potential to affect the accuracy of sentiment because certain aspects could be neglected due to the limitation of the lexicon. Although a manually built lexicon can achieve better accuracy, the construction of the sentiment lexicon requires human effort and is time-consuming.
- The research focuses mainly on the informal text of online reviews that is close to the way people speak in real life. Text from other sources such as professional critics or product reports from evaluative organizations is not used in this research.
- The SentiGA method can implement and train more robust base classifiers for generating a larger pool of classifiers and the number of generations could be increased, thereby leading to better accuracy. However, the SentiGA method was not considered in this research due to resources and time constraints.
- Both of SentiXGboost and SentiGA methods have been tested on some types of features. Therefore, we could test and update our proposed methods with more NLP-based features but did not due to the limitation of the technical resources.

7.3 Future Work

Although this research has filled a few gaps in the field of sentiment analysis, further work is required to achieve further improvements. In future works, we plan to implement a strong weighted voting ensemble classifier algorithm for decision making. This would be combined with the Genetic Algorithm to optimize the best performing classifiers for sentiment analysis. Furthermore, the time and computational complexities of the proposed schemes will be examined and compared with existing methods. Additionally, the possibility of deploying an optimized ensemble classifier method using other evolutionary algorithms would be explored in future studies. Other future research directions include developing a robust and dynamic ensemble classifier selection method for sentiment analysis based on similarity score using the Genetic Algorithm.

REFERENCES

[1] Alarifi, A., Alsaleh, M., & Al-Salman, A. (2016). Twitter turing test: Identifying social machines. *Information Sciences*, *372*, 332-346.

[2] Öztürk, N., & Ayvaz, S. (2018). Sentiment analysis on Twitter: A text mining approach to the Syrian refugee crisis. *Telematics and Informatics*, *35*(1), 136-147.

[3] Han, Y., Liu, Y., & Jin, Z. (2019). Sentiment analysis via semi-supervised learning: a model based on dynamic threshold and multi-classifiers. *Neural Computing and Applications*, 1-13.

[4] Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1), 1-167.

[5] Pozzi, F. A., Fersini, E., Messina, E., & Liu, B. (2017). Challenges of sentiment analysis in social networks: an overview. In *Sentiment analysis in social networks* (pp. 1-11). Morgan Kaufmann.

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

[7] Turney, P. D. (2002, July). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 417-424). Association for Computational Linguistics.

[8] Stoyanov, V., & Cardie, C. (2008, August). Topic identification for fine-grained opinion analysis. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)* (pp. 817-824).

[9] Kouloumpis, E., Wilson, T., & Moore, J. (2011, July). Twitter sentiment analysis: The good the bad and the omg!. In *Fifth International AAAI conference on weblogs and social media*.

[10] Villena Román, J., Lana Serrano, S., Martínez Cámara, E., & González Cristóbal, J. C. (2013). Tass-workshop on sentiment analysis at sepln.

[11] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, *1*(12), 2009.

[12] Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., & Wilson, T. (2013, June). SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (pp. 312-320).

[13] Rosenthal , S., Ritter , A., Nakov , P., Stoyanov , V. (2014, Aug).SemEval-2014 Task 9: Sentiment Analysis in Twitter. *In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014),* (pp73–80).Dublin, Ireland.

[14] Nakov, P., Rosenthal, S., Kiritchenko, S., Mohammad, S. M., Kozareva, Z., Ritter, A., ... & Zhu, X. (2016). Developing a successful SemEval task in sentiment analysis of Twitter and other social media texts. *Language Resources and Evaluation*, *50*(1), 35-65.

[15] Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., Stoyanov, V. (2015). SemEval-2015 task 10: Sentiment analysis in Twitter. *In: Proceedings of the 9th International Workshopon Semantic Evaluation, SemEval '15,* (pp 450–462). Denver, Colorado, USA.

[16] Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., & Stoyanov, V. (2019). SemEval-2016 task 4: Sentiment analysis in Twitter. *arXiv preprint arXiv:1912.01973*.

[17] Rosenthal, S., Farra, N., & Nakov, P. (2017, August). SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)* (pp. 502-518).

[18] Gamal, D., Alfonse, M., M El-Horbaty, E. S., & M Salem, A. B. (2019). Analysis of Machine Learning Algorithms for Opinion Mining in Different Domains. *Machine Learning and Knowledge Extraction*, *1*(1), 224-234.

[19] Moreno-Seco, F., Inesta, J. M., De León, P. J. P., & Micó, L. (2006, August). Comparison of classifier fusion methods for classification in pattern recognition tasks. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)* (pp. 705-713). Springer, Berlin, Heidelberg.

[20] Talbi, E. G. (2009). *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons.

[21] Mondal, A., Cambria, E., Das, D., Hussain, A., & Bandyopadhyay, S. (2018). Relation extraction of medical concepts using categorization and sentiment analysis. *Cognitive Computation*, *10*(4), 670-685.

[22] Appel, O., Chiclana, F., Carter, J., & Fujita, H. (2018). Successes and challenges in developing a hybrid approach to sentiment analysis. *Applied Intelligence*, *48*(5), 1176-1188.

[23] Arif, F., & Dulhare, U. N. (2017). A Machine Learning Based Approach for Opinion Mining on Social Network Data. In *Computer Communication, Networking and Internet Security* (pp. 135-147). Springer, Singapore.

[24] Gogna, A., & Tayal, A. (2013). Metaheuristics: review and application. *Journal of Experimental & Theoretical Artificial Intelligence*, *25*(4), 503-526.

[25] Dos Santos, E. M. (2012). Evolutionary algorithms applied to classifier ensemble selection. *XLIV SBPO/XVI CLAIO*, 419-430.

[26] Symeonidis, S., Effrosynidis, D., Kordonis, J., & Arampatzis, A. (2017, August). DUTH at SemEval-2017 Task 4: a voting classification approach for Twitter sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 704-708).

[27] Hasan, A., Moin, S., Karim, A., & Shamshirband, S. (2018). Machine learning-based sentiment analysis for twitter accounts. *Mathematical and Computational Applications*, *23*(1), 11.

[28] Othman, M., Hassan, H., Moawad, R., & Idrees, A. M. (2018). A linguistic approach for opinionated documents summary. *Future Computing and Informatics Journal*, *3*(2), 152-158.

[29] Wang, J., & Dong, A. (2010, October). A comparison of two text representations for sentiment analysis. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)* (Vol. 11, pp. V11-35). IEEE.

[30] Kanayama, H., Nasukawa, T., & Watanabe, H. (2004). Deeper sentiment analysis using machine translation technology. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics* (pp. 494-500).

[31] Raychev, V., & Nakov, P. (2019). Language-independent sentiment analysis using subjectivity and positional information. *arXiv preprint arXiv:1911.12544*.

[32] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, *5*(1), 1-167.

[33] Barberis, N., Shleifer, A., & Vishny, R. (1998). A model of investor sentiment. *Journal of financial economics*, *49*(3), 307-343.

[34] Das, S., & Chen, M. (2001, July). Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific finance association annual conference (APFA)* (Vol. 35, p. 43).

[35] Nasukawa, T., & Yi, J. (2003, October). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture* (pp. 70-77).

[36] Yi, J., Nasukawa, T., Bunescu, R., & Niblack, W. (2003, November). Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Third IEEE international conference on data mining* (pp. 427-434). IEEE.

[37] Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 142-150).

[38] Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., & Patwardhan, S. (2005, October). OpinionFinder: A system for subjectivity

analysis. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations* (pp. 34-35).

[39] Alm, C. O., Roth, D., & Sproat, R. (2005, October). Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing* (pp. 579-586).

[40] Kumar, V., & Minz, S. (2013, January). Mood classifiaction of lyrics using SentiWordNet. In *2013 International Conference on Computer Communication and Informatics* (pp. 1-5). IEEE.

[41] Kim, S. M., & Hovy, E. (2004). Determining the sentiment of opinions. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics* (pp. 1367-1373).

[42] Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of natural language processing*, *2*(2010), 627-666.

[43] Kundi, F. M., Ahmad, S., Khan, A., & Asghar, M. Z. (2014). Detection and scoring of internet slangs for sentiment analysis using SentiWordNet. *Life Science Journal*, *11*(9), 66-72.

[44] Majumder, N., Poria, S., Gelbukh, A., & Cambria, E. (2017). Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, *32*(2), 74-79.

[45] Nakagawa, T., Inui, K., & Kurohashi, S. (2010, June). Dependency tree-based sentiment classification using CRFs with hidden variables. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 786-794).

[46] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis (foundations and trends (R) in Information Retrieval).

[47] Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining text data* (pp. 415-463). Springer, Boston, MA.

[48] Ravi, K., & Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-based systems*, *89*, 14-46.

[49] Dave, K., Lawrence, S., & Pennock, D. M. (2003, May). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web* (pp. 519-528).

[50] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*.

[51] Rao, G., Huang, W., Feng, Z., & Cong, Q. (2018). LSTM with sentence representations for document-level sentiment classification. *Neurocomputing*, *308*, 49-57.

[52] Bespalov, D., Qi, Y., Bai, B., & Shokoufandeh, A. (2012, September). Sentiment classification with supervised sequence embedding. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 159-174). Springer, Berlin, Heidelberg.

[53] Li, B., Liu, T., Du, X., Zhang, D., & Zhao, Z. (2015). Learning document embeddings by predicting n-grams for sentiment classification of long movie reviews. *arXiv preprint arXiv:1512.08183*.

[54] Matsumoto, S., Takamura, H., & Okumura, M. (2005, May). Sentiment classification using word sub-sequences and dependency sub-trees. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 301-311). Springer, Berlin, Heidelberg.

[55] Onan, A., Korukoglu, S., & Bulut, H. (2016). LDA-based Topic Modelling in Text Sentiment Classification: An Empirical Analysis. *Int. J. Comput. Linguistics Appl.*, *7*(1), 101-119.

[56] Chen, H., Sun, M., Tu, C., Lin, Y., & Liu, Z. (2016, November). Neural sentiment classification with user and product attention. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 1650-1659).

[57] Tang, D. (2015, February). Sentiment-specific representation learning for document-level sentiment analysis. In *Proceedings of the eighth ACM international conference on web search and data mining* (pp. 447-452).

[58] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016, June). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 1480-1489).

[59] Wiebe, J., Bruce, R., & O'Hara, T. P. (1999, June). Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics* (pp. 246-253).

[60] Wang, S., Li, D., Song, X., Wei, Y., & Li, H. (2011). A feature selection method based on improved fisher's discriminant ratio for text sentiment classification. *Expert Systems with Applications*, *38*(7), 8696-8702.

[61] Benamara, F., Chardon, B., Mathieu, Y., & Popescu, V. (2011, November). Towards context-based subjectivity analysis. In *Proceedings of 5th International Joint Conference on Natural Language Processing* (pp. 1180-1188).

[62] Rustamov, S., Mustafayev, E., & Clements, M. A. (2013, June). Sentence-level subjectivity detection using neuro-fuzzy models. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* (pp. 108-114).

[63] Chenlo, J. M., & Losada, D. E. (2014). An empirical study of sentence features for subjectivity and polarity classification. *Information Sciences*, *280*, 275-288.

[64] Khan, F. H., Qamar, U., & Bashir, S. (2016). SWIMS: Semi-supervised subjective feature weighting and intelligent model selection for sentiment analysis. *Knowledge-Based Systems*, *100*, 97-111.

[65] Hathlian, N. F. B., & Hafez, A. M. (2020). Subjective text mining for arabic social media. In *Cognitive Analytics: Concepts, Methodologies, Tools, and Applications* (pp. 1483-1495). IGI Global.

[66] Wang, D., Zhu, S., & Li, T. (2013). SumView: A Web-based engine for summarizing product reviews and customer opinions. *Expert Systems with Applications*, *40*(1), 27-33.

[67] Mukherjee, A., & Liu, B. (2012, July). Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 339-348).

[68] Yin, Y., Song, Y., & Zhang, M. (2017, September). Document-level multi-aspect sentiment classification as machine comprehension. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2044-2054).

[69] Hu, M., & Liu, B. (2004, August). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 168-177).

[70] Meng, X., & Wang, H. (2009, August). Mining user reviews: from specification to summarization. In *Proceedings of the ACL-IJCNLP 2009 conference short papers* (pp. 177-180).

[71] Lu, Y., Zhai, C., & Sundaresan, N. (2009, April). Rated aspect summarization of short comments. In *Proceedings of the 18th international conference on World wide web* (pp. 131-140).

[72] Nishikawa, H., Hasegawa, T., Matsuo, Y., & Kikui, G. (2010, August). Opinion summarization with integer linear programming formulation for sentence extraction and ordering. In *Coling 2010: Posters* (pp. 910-918).

[73] Lpez Condori, R. E., & Salgueiro Pardo, T. A. (2017). Opinion summarization methods. *Expert Systems with Applications: An International Journal*, *78*(C), 124-134.

[74] Eckman, P. (1972, January). Universal and cultural differences in facial expression of emotion. In *Nebraska symposium on motivation* (Vol. 19, pp. 207-284). University of Nebraska Press.

[75] David, J. P., & Suls, J. (1999). Coping efforts in daily life: Role of Big Five traits and problem appraisals. *Journal of personality*.

[76] Kwon, O. W., Chan, K., Hao, J., & Lee, T. W. (2003). Emotion recognition by speech signals. In *Eighth European Conference on Speech Communication and Technology*.

[77] Tai, C. H., Tan, Z. H., Lin, Y. S., & Chang, Y. S. (2015, October). Mental disorder detection and measurement using latent Dirichlet allocation and SentiWordNet. In *2015 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1215-1220). IEEE.

[78] Avots, E., Sapiński, T., Bachmann, M., & Kamińska, D. (2019). Audiovisual emotion recognition in wild. *Machine Vision and Applications*, *30*(5), 975-985.

[79] Ahmed, F., Bari, A. H., & Gavrilova, M. L. (2019). Emotion recognition from body movement. *IEEE Access*, *8*, 11761-11781.

[80] Lim, J. Z., Mountstephens, J., & Teo, J. (2020). Emotion recognition using eye-tracking: taxonomy, review and current challenges. *Sensors*, *20*(8), 2384.

[81] Wawre, S. V., & Deshmukh, S. N. (2016). Sentiment classification using machine learning techniques. *International Journal of Science and Research (IJSR)*, *5*(4), 819-821.

[82] Weitzel, L., Prati, R. C., & Aguiar, R. F. (2016). The comprehension of figurative language: what is the influence of irony and sarcasm on NLP techniques?. In *Sentiment Analysis and Ontology Engineering* (pp. 49-74). Springer, Cham.

[83] Boldrini, E., Balahur, A., Martínez-Barco, P., & Montoyo, A. (2012). Using EmotiBlog to annotate and analyse subjectivity in the new textual genres. Data Mining and Knowledge Discovery, 25(3), 603-634.165

[84] Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., & Huang, R. (2013, October). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 704-714).

[85] Maynard, D. G., & Greenwood, M. A. (2014, March). Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In Lrec 2014 proceedings. ELRA.

[86] Xu, K., Liao, S. S., Li, J., & Song, Y. (2011). Mining comparative opinions from customer reviews for competitive intelligence. *Decision support systems*, *50*(4), 743-754.

[87] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. Computational linguistics, 37(2), 267-307.166

[88] Das, S. R. (2011). News analytics: Framework, techniques and metrics. *The Handbook of News Analytics in Finance*, *2*.

[89] Kanayama, H., Nasukawa, T., & Watanabe, H. (2004). Deeper sentiment analysis using machine translation technology. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics* (pp. 494-500).

[90] McDonald, R., Hannan, K., Neylon, T., Wells, M., & Reynar, J. (2007, June). Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the*

45th annual meeting of the association of computational linguistics (pp. 432-439).

[91] Abbasi, A., France, S., Zhang, Z., & Chen, H. (2010). Selecting attributes for sentiment classification using feature relation networks. *IEEE Transactions on Knowledge and Data Engineering*, *23*(3), 447-462.

[92] Boiy, E., & Moens, M. F. (2009). A machine learning approach to sentiment analysis in multilingual Web texts. *Information retrieval*, *12*(5), 526-558.

[93] Maks, I., & Vossen, P. (2012). A lexicon model for deep sentiment analysis and opinion mining applications. *Decision Support Systems*, *53*(4), 680-688.

[94] Yu, Y., Duan, W., & Cao, Q. (2013). The impact of social and conventional media on firm equity value: A sentiment analysis approach. *Decision support systems*, *55*(4), 919-926.

[95] Abdul-Mageed, M., Diab, M., & Kübler, S. (2014). SAMAR: Subjectivity and sentiment analysis for Arabic social media. *Computer Speech & Language*, *28*(1), 20-37.

[96] Appel, O., Chiclana, F., Carter, J., & Fujita, H. (2016). A hybrid approach to the sentiment analysis problem at the sentence level. *Knowledge-Based Systems*, *108*, 110-124.

[97] Farooq, U., Mansoor, H., Nongaillard, A., Ouzrout, Y., & Qadir, M. A. (2017). Negation Handling in Sentiment Analysis at Sentence Level. *JCP*, *12*(5), 470-478.

[98] Zhang, Y., Zhang, Z., Miao, D., & Wang, J. (2019). Three-way enhanced convolutional neural networks for sentence-level sentiment classification. *Information Sciences*, *477*, 55-64.

[99] Arulmurugan, R., Sabarmathi, K. R., & Anandakumar, H. J. C. C. (2019). Classification of sentence level sentiment analysis using cloud machine learning techniques. *Cluster Computing*, *22*(1), 1199-1209.

[100] Wang, H., Lu, Y., & Zhai, C. (2010, July). Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 783-792).

[101] Poria, S., Chaturvedi, I., Cambria, E., & Bisio, F. (2016, July). Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis. In *2016 international joint conference on neural networks (IJCNN)* (pp. 4465-4473). IEEE.

[102] Ruder, S., Ghaffari, P., & Breslin, J. G. (2016). A hierarchical model of reviews for aspect-based sentiment analysis. *arXiv preprint arXiv:1609.02745*.

[103] Cambria, E., Schuller, B., Xia, Y., & Havasi, C. (2013). New avenues in opinion mining and sentiment analysis. *IEEE Intelligent systems*, *28*(2), 15-21.

[104] Poria, S., Cambria, E., Winterstein, G., & Huang, G. B. (2014). Sentic patterns: Dependency-based rules for concept-level sentiment analysis. *Knowledge-Based Systems*, *69*, 45-63.

[105] Tsai, A. C. R., Wu, C. E., Tsai, R. T. H., & Hsu, J. Y. J. (2013). Building a concept-level sentiment dictionary based on commonsense knowledge. *IEEE Intelligent Systems*, *28*(2), 22-30.

[106] Balahur, A., Hermida, J. M., & Montoyo, A. (2011). Building and exploiting emotinet, a knowledge base for emotion detection based on the appraisal theory model. *IEEE transactions on affective computing*, *3*(1), 88-101.

[107] Weichselbraun, A., Gindl, S., & Scharl, A. (2013). Extracting and grounding contextualized sentiment lexicons. *IEEE Intelligent Systems*, *28*(2), 39-46.

[108] Poria, S., Gelbukh, A., Hussain, A., Howard, N., Das, D., & Bandyopadhyay, S. (2013). Enhanced SenticNet with affective labels for concept-based opinion mining. *IEEE Intelligent Systems*, *28*(2), 31-38.

[109] Cambria, E., Gastaldo, P., Bisio, F., & Zunino, R. (2015). An ELM-based model for affective analogical reasoning. *Neurocomputing*, *149*, 443-455.

[110] Shah, R. R., Yu, Y., Verma, A., Tang, S., Shaikh, A. D., & Zimmermann, R. (2016). Leveraging multimodal information for event summarization and concept-level sentiment analysis. *Knowledge-Based Systems, 108*, 102-109.

[111] Peng, H., & Cambria, E. (2017, April). CSenticNet: a concept-level resource for sentiment analysis in chinese language. In *International Conference on Computational Linguistics and Intelligent Text Processing* (pp. 90-104). Springer, Cham.

[112] Satapathy, R., Singh, A., & Cambria, E. (2019, November). PhonSenticNet: a cognitive approach to microtext normalization for concept-level sentiment analysis. In *International Conference on Computational Data and Social Networks* (pp. 177-188). Springer, Cham.

[113] Benamara, F., Cesarano, C., Picariello, A., Recupero, D. R., & Subrahmanian, V. S. (2007). Sentiment analysis: Adjectives and adverbs are better than adjectives alone. *ICWSM*, *7*, 203-206.

[114] Serrano-Guerrero, J., Olivas, J. A., Romero, F. P., & Herrera-Viedma, E. (2015). Sentiment analysis: A review and comparative analysis of web services. *Information Sciences*, *311*, 18-38.

[115] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, *5*(4), 1093-1113.

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

[117] Saleh, M. R., Martín-Valdivia, M. T., Montejo-Ráez, A., & Ureña-López, L. A. (2011). Experiments with SVM to classify opinions in different domains. *Expert Systems with Applications*, *38*(12), 14799-14804.

[118] Moraes, R., Valiati, J. F., & Neto, W. P. G. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*, *40*(2), 621-633.

[119] Bilal, M., Israr, H., Shahid, M., & Khan, A. (2016). Sentiment classification of Roman-Urdu opinions using Naïve Bayesian, Decision Tree and KNN classification techniques. *Journal of King Saud University-Computer and Information Sciences*, *28*(3), 330-344.

[120] Bhavitha, B. K., Rodrigues, A. P., & Chiplunkar, N. N. (2017, March). Comparative study of machine learning techniques in sentimental analysis. In *2017 International conference on inventive communication and computational technologies (ICICCT)* (pp. 216-221). IEEE.

[121] Poornima, A., & Priya, K. S. (2020, March). A comparative sentiment analysis of sentence embedding using machine learning techniques. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 493-496). IEEE.

[122] Yasen, M., & Tedmori, S. (2019, April). Movies Reviews sentiment analysis and classification. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)* (pp. 860-865). IEEE.

[123] Liu, B. (2015). Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press.

[124] Ding, X., Liu, B., & Yu, P. S. (2008, February). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining* (pp. 231-240).

[125] Kim, H., & Jeong, Y. S. (2019). Sentiment classification using convolutional neural networks. *Applied Sciences*, *9*(11), 2347.

[126] Ranzato, M. A., Huang, F. J., Boureau, Y. L., & LeCun, Y. (2007, June). Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *2007 IEEE conference on computer vision and pattern recognition* (pp. 1-8). IEEE.

[127] Kosmopoulos, D. I., Doulamis, N. D., & Voulodimos, A. S. (2012). Bayesian filter based behavior recognition in workflows allowing for user feedback. *Computer Vision and Image Understanding*, *116*(3), 422-434.

[128] Seeger, M. (2000). Learning with labeled and unlabeled data.

[129] Zhu, X. J. (2005). Semi-supervised learning literature survey.

[130] Cheng, H., Hua, K. A., Vu, K., & Liu, D. (2008, March). Semi-supervised dimensionality reduction in image feature space. In *Proceedings of the 2008 ACM symposium on Applied computing* (pp. 1207-1211).

[131] Wang, J., Kumar, S., & Chang, S. F. (2012). Semi-supervised hashing for large-scale search. *IEEE transactions on pattern analysis and machine intelligence*, *34*(12), 2393-2406.

[132] Cortes, C., & Mohri, M. (2007). On transductive regression. *Advances in neural information processing systems*, *19*, 305.

[133] Grira, N., Crucianu, M., & Boujemaa, N. (2004). Unsupervised and semi-supervised clustering: a brief survey. *A review of machine learning techniques for processing multimedia content*, *1*, 9-16.

[134] Anand, S., Mittal, S., Tuzel, O., & Meer, P. (2013). Semi-supervised kernel mean shift clustering. *IEEE transactions on pattern analysis and machine intelligence*, *36*(6), 1201-1215.

[135] Wang, Q., Yuen, P. C., & Feng, G. (2013). Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions. *Pattern Recognition*, *46*(9), 2576-2587.

[136] Hoi, S. C., Liu, W., & Chang, S. F. (2010). Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *6*(3), 1-26.

[137] Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, *3*(2), 143-157.

[138] De Albornoz, J. C., Plaza, L., & Gervás, P. (2010, July). A hybrid approach to emotional sentence polarity and intensity classification. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning* (pp. 153-161).

[139] Ghiassi, M., Skinner, J., & Zimbra, D. (2013). Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with applications*, *40*(16), 6266-6282.

[140] Al Amrani, Y., Lazaar, M., & El Kadiri, K. E. (2018). Random forest and support vector machine based hybrid approach to sentiment analysis. *Procedia Computer Science*, *127*, 511-520.

[141] Asghar, M. Z., Kundi, F. M., Ahmad, S., Khan, A., & Khan, F. (2018). T-SAF: Twitter sentiment analysis framework using a hybrid classification scheme. *Expert Systems*, *35*(1), e12233.

[142] Rajeswari, A. M., Mahalakshmi, M., Nithyashree, R., & Nalini, G. (2020, July). Sentiment Analysis for Predicting Customer Reviews using a Hybrid

Approach. In *2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)* (pp. 200-205). IEEE.

[143] Takamura, H., Inui, T., and Okumura, M. (2007). Extracting semantic orientations of phrases from dictionary. *In HLT-NAACL*,2007, 292–299.

[144] Mohammad, S. M. and Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, *29*(3):436–465.

[145] Lin, Z., Tan, S., Liu, Y., Cheng, X., and Xu, X. (2013). Cross-language opinion lexicon extraction using mutual-reinforcement label propagation. *PloS one*, *8*(11):e79294.

[146] Zhang, Z., & Singh, M. P. (2014, June). Renew: A semi-supervised framework for generating domain-specific lexicons and sentiment analysis. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 542-551).

[147] Hatzivassiloglou, V., & McKeown, K. (1997, July). Predicting the semantic orientation of adjectives. In *35th annual meeting of the association for computational linguistics and 8th conference of the european chapter of the association for computational linguistics* (pp. 174-181).

[148] Taboada, M., Anthony, C., & Voll, K. D. (2006, May). Methods for Creating Semantic Orientation Dictionaries. In *LREC* (pp. 427-432).

[149] Nasim, Z., Rajput, Q., & Haider, S. (2017, July). Sentiment analysis of student feedback using machine learning and lexicon based approaches. In *2017 international conference on research and innovation in information systems (ICRIIS)* (pp. 1-6). IEEE.

[150] Han, H., Zhang, J., Yang, J., Shen, Y., & Zhang, Y. (2018). Generate domain-specific sentiment lexicon for review sentiment analysis. *Multimedia Tools and Applications*, *77*(16), 21265-21280.

[151] Rezaeinia, S. M., Rahmani, R., Ghodsi, A., & Veisi, H. (2019). Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications*, *117*, 139-147.

[152] Machová, K., Mikula, M., Gao, X., & Mach, M. (2020). Lexicon-Based Sentiment Analysis Using Particle Swarm Optimization. *Electronics*, *9*(8), 1317.

[153] Kuncheva, L. I. (2014). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

[154] Rokach, L. (2010). Ensemble-based classifiers. *Artificial intelligence review*, *33*(1), 1-39.

[155] Tang, B., Chen, Q., Wang, X., & Wang, X. (2010, November). Reranking for stacking ensemble learning. In *International Conference on Neural Information Processing* (pp. 575-584). Springer, Berlin, Heidelberg.

[156] Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.

[157] Kearns, M. (1988). Thoughts on hypothesis boosting. *Unpublished manuscript*, *45*, 105.

[158] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

[159] Sewell, M. (2008). Ensemble learning. *RN*, *11*(02), 1-34.

[160] Qadir, A., & Riloff, E. (2013, June). Bootstrapped learning of emotion hashtags# hashtags4you. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis* (pp. 2-11).

[161] Prusa, J., Khoshgoftaar, T. M., & Napolitano, A. (2015, December). Utilizing ensemble, data sampling and feature selection techniques for improving classification performance on tweet sentiment data. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (pp. 535-542). IEEE.

[162] Celikyilmaz, A., Hakkani-Tür, D., & Feng, J. (2010, December). Probabilistic model-based sentiment analysis of twitter messages. In *2010 IEEE Spoken Language Technology Workshop* (pp. 79-84). IEEE.

[163] Ho, T. K. (1995, August). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (Vol. 1, pp. 278-282). IEEE.

[164] Friedman, J. H. (1999). Greedy function approximation: A gradient boosting machine 1 function estimation 2 numerical optimization in function space. *North*, *1*(3), 1-10.

[165] Ho, T. K. (2002). A data complexity analysis of comparative advantages of decision forest constructors. *Pattern Analysis & Applications*, *5*(2), 102-112.

[166] Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, *20*(8), 832-844.

[167] Polikar, R. (2012). Ensemble learning. In *Ensemble machine learning* (pp. 1-34). Springer, Boston, MA.

[168] Gryc, W., & Moilanen, K. (2014). Leveraging textual sentiment analysis with social network modelling. *From Text to Political Positions: Text analysis across disciplines*, *55*, 47.

[169] Wan, Y., & Gao, Q. (2015, November). An ensemble sentiment classification system of twitter data for airline services analysis. In *2015 IEEE international conference on data mining workshop (ICDMW)* (pp. 1318-1325). IEEE.

[170] Chalothom, T., & Ellman, J. (2015). Simple approaches of sentiment analysis via ensemble learning. In *information science and applications* (pp. 631-639). Springer, Berlin, Heidelberg.

[171] Aziz, R. H. H., & Dimililer, N. (2020, December). Twitter Sentiment Analysis using an Ensemble Weighted Majority Vote Classifier. In *2020 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 103-109). IEEE.

[172] Rokach, L. (2010). *Pattern classification using ensemble methods* (Vol. 75). World Scientific.

[173] Kolyal, A. K., Ekbal, A., & Bandyopadhyay, S. (2013). USING VOTING APPROACH FOR EVENT EXTRACTION AND EVENT-DCT, EVENT-TIME RELATION IDENTIFICATION. *International Journal of Artificial Intelligence & Applications*, *4*(1), 65.

[174] Nazeer, I., Rashid, M., Gupta, S. K., & Kumar, A. (2021). Use of Novel Ensemble Machine Learning Approach for Social Media Sentiment Analysis. In *Analyzing Global Social Media Consumption* (pp. 16-28). IGI Global.

[175] Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, *5*(2), 241-259.

[176] Džeroski, S., & Ženko, B. (2004). Is combining classifiers with stacking better than selecting the best one?. *Machine learning*, *54*(3), 255-273.

[177] Chan, P. K., & Stolfo, S. J. (1993, July). Toward parallel and distributed learning by meta-learning. In *AAAI workshop in Knowledge Discovery in Databases* (pp. 227-240).

[178] Chan, P. K., & Stolfo, S. J. (1997). On the accuracy of meta-learning for scalable data mining. *Journal of Intelligent Information Systems*, *8*(1), 5-28.

[179] Chan, P. K., & Stolfo, S. J. (1995). A comparative evaluation of voting and meta-learning on partitioned data. In *Machine Learning Proceedings 1995* (pp. 90-98). Morgan Kaufmann.

[180] Seewald, A. K., & Fürnkranz, J. (2001, September). An evaluation of grading classifiers. In *International symposium on intelligent data analysis* (pp. 115-124). Springer, Berlin, Heidelberg.

[181] Schaffer, C., Selecting a classification method by cross-validation. Machine Learning 13(1):135-143, 1993.

[182] Ruta, D., & Gabrys, B. (2001, July). Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting. In *International Workshop on Multiple Classifier Systems* (pp. 399-408). Springer, Berlin, Heidelberg.

[183] Ruta, D., & Gabrys, B. (2005). Classifier selection for majority voting. *Information fusion*, *6*(1), 63-81.

[184] Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.

[185] Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Boston, MA: Addison-Wesley.

[186] Mitchell, M. (1995). Genetic algorithms: An overview. *Complexity*, *1*(1), 31-39.

[187] Albadr, M. A., Tiun, S., Ayob, M., & AL-Dhief, F. (2020). Genetic Algorithm Based on Natural Selection Theory for Optimization Problems. *Symmetry*, *12*(11), 1758.

[188] Dave, K., Lawrence, S., & Pennock, D. M. (2003, May). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web* (pp. 519-528).

[189] Yu, H., & Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing* (pp. 129-136).

[190] Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*.

[191] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1631-1642).

[192] Taj, S., Shaikh, B. B., & Meghji, A. F. (2019, January). Sentiment analysis of news articles: A lexicon based approach. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-5). IEEE.

[193] Cambria, E. (2013, November). An introduction to concept-level sentiment analysis. In *Mexican international conference on artificial intelligence* (pp. 478-483). Springer, Berlin, Heidelberg.

[194] Jabreel, M., & Moreno, A. (2016, October). SentiRich: Sentiment Analysis of Tweets Based on a Rich Set of Features. In *CCIA* (pp. 137-146).

[195] Paltoglou, G., & Thelwall, M. (2012). Twitter, MySpace, Digg: Unsupervised sentiment analysis in social media. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *3*(4), 1-19.

[196] Joshi, M., & Rosé, C. (2009, August). Generalizing dependency features for opinion mining. In *Proceedings of the ACL-IJCNLP 2009 conference short papers* (pp. 313-316).

[197] Hemmatian, F., & Sohrabi, M. K. (2019). A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review*, 1-51.

[198] Madasu, A., & Elango, S. (2020). Efficient feature selection techniques for sentiment analysis. *Multimedia Tools and Applications*, *79*(9), 6313-6335.

[199] Kumar, H. M., Harish, B. S., & Darshan, H. K. (2019). Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method. *International Journal of Interactive Multimedia & Artificial Intelligence*, *5*(5).

[200] Hassonah, M. A., Al-Sayyed, R., Rodan, A., Ala'M, A. Z., Aljarah, I., & Faris, H. (2020). An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter. *Knowledge-Based Systems*, *192*, 105353.

[201] Afzaal, M., Usman, M., & Fong, A. (2019). Predictive aspect-based sentiment classification of online tourist reviews. *Journal of Information Science*, *45*(3), 341-363.

[202] Naresh, A., Venkata Krishna, P. (2020). An efficient approach for sentiment analysis using machine learning algorithm. *Evolution Intelligence*. doi: 10.1007/s12065-020-00429-1

[203] Wang, G., Sun, J., Ma, J., Xu, K., & Gu, J. (2014). Sentiment classification: The contribution of ensemble learning. *Decision support systems*, *57*, 77-93.

[204] Xia, R., Zong, C., & Li, S. (2011). Ensemble of feature sets and classification algorithms for sentiment classification. *Information sciences*, *181*(6), 1138-1152.

[205] Kilimci, Z. H., & Omurca, S. I. (2018). Extended feature spaces based classifier ensembles for sentiment analysis of short texts. *Information Technology and Control*, *47*(3), 457-470.

[206] Akhtar, M. S., Ekbal, A., & Cambria, E. (2020). How intense are you? predicting intensities of emotions and sentiments using stacked ensemble. *IEEE Computational Intelligence Magazine*, *15*(1), 64-75.

[207] Khan, J., Alam, A., Hussain, J., & Lee, Y. K. (2019). EnSWF: effective features extraction and selection in conjunction with ensemble learning methods for document sentiment classification. *Applied Intelligence*, *49*(8), 3123-3145.

[208] Pong-Inwong, C., & Kaewmak, K. (2016, October). Improved sentiment analysis for teaching evaluation using feature selection and voting ensemble learning integration. In *2016 2nd IEEE international conference on computer and communications (ICCC)* (pp. 1222-1225). IEEE.

[209] Khalid, M., Ashraf, I., Mehmood, A., Ullah, S., Ahmad, M., & Choi, G. S. (2020). GBSVM: Sentiment Classification from Unstructured Reviews Using Ensemble Classifier. *Applied Sciences*, *10*(8), 2788.

[210] Saleena, N. (2018). An ensemble classification system for twitter sentiment analysis. *Procedia Computer Science*, *132*, 937-946.

[211] Yueyang, L., & Wang, Y. Z. (2019, May). Detecting Opinion Polarities Using Ensemble of Classification Algorithms. In *Journal of Physics: Conference Series* (Vol. 1229, No. 1, p. 012065). IOP Publishing.

[212] Araque, O., Corcuera-Platas, I., Sánchez-Rada, J. F., & Iglesias, C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, *77*, 236-246.

[213] Alrehili, A., & Albalawi, K. (2019, April). Sentiment Analysis of Customer Reviews Using Ensemble Method. In *2019 International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-6). IEEE.

[214] Kraft, D. H., Petry, F. E., Buckles, B. P., & Sadasivan, T. (1994, June). The use of genetic programming to build queries for information retrieval. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence* (pp. 468-473). IEEE.

[215] Martin-Bautista, M. J., Larsen, H. L., Nicolaisen, J., & Svendsen, T. (1997, July). An approach to an adaptive information retrieval agent using genetic algorithms with fuzzy set genes. In *Proceedings of 6th International Fuzzy Systems Conference* (Vol. 3, pp. 1227-1232). IEEE.

[216] Ishaq, A., Asghar, S., & Gillani, S. A. (2020). Aspect-Based Sentiment Analysis Using a Hybridized Approach Based on CNN and GA. *IEEE Access*, *8*, 135499-135512.

[217] Cahya, R. A., Adimanggala, D., & Supianto, A. A. (2019, September). Deep Feature Weighting Based on Genetic Algorithm and Naïve Bayes for Twitter Sentiment Analysis. In *2019 International Conference on Sustainable Information Engineering and Technology (SIET)* (pp. 326-331). IEEE.

[218] Iqbal, F., Hashmi, J. M., Fung, B. C., Batool, R., Khattak, A. M., Aleem, S., & Hung, P. C. (2019). A hybrid framework for sentiment analysis using genetic algorithm based feature reduction. *IEEE Access*, *7*, 14637-14652

[219] Fatyanosa, T. N., Bachtiar, F. A., & Data, M. (2018, November). Feature Selection using Variable Length Chromosome Genetic Algorithm for Sentiment Analysis. In *2018 International Conference on Sustainable Information Engineering and Technology (SIET)* (pp. 27-32). IEEE.

[220] Keshavarz, H., Abadeh, M. S., & Almasi, M. (2017, September). A new lexicon learning algorithm for sentiment analysis of big data. In *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)* (pp. 000249-000254). IEEE.

[221] Saidani, F. R., & Rassoul, I. (2017). A weighted genetic approach for feature selection in sentiment analysis. *International Journal of Computational Intelligence and Applications*, *16*(02), 1750013.

[222] Baziotis, C., Pelekis, N., & Doulkeridis, C. (2017, August). Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)* (pp. 747-754).

[223] Cliche, M. (2017). Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms. *arXiv preprint arXiv:1704.06125*.

[224] Kolovou, A., Kokkinos, F., Fergadis, A., Papalampidi, P., Iosif, E., Malandrakis, N., ... & Potamianos, A. (2017, August). Tweester at SemEval-2017 Task 4: Fusion of Semantic-Affective and pairwise classification models for sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 675-682).

[225] Onyibe, C., & Habash, N. (2017, August). OMAM at SemEval-2017 Task 4: English sentiment analysis with conditional random fields. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)* (pp. 670-674).

[226] Zhang, H., Wang, J., Zhang, J., & Zhang, X. (2017, August). Ynu-hpcc at semeval 2017 task 4: using a multi-channel cnn-lstm model for sentiment classification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 796-801).

[227] González, J. A., Pla, F., & Hurtado, L. F. (2017, August). ELiRF-UPV at SemEval-2017 task 4: sentiment analysis using deep learning. In *Proceedings of the 11th international workshop on semantic evaluation (SEMEVAL-2017)* (pp. 723-727).

[228] Lozić, D., Šarić, D., Tokić, I., Medić, Z., & Šnajder, J. (2017, August). TakeLab at SemEval-2017 Task 4: Recent deaths and the power of nostalgia in sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 784-789).

[229] Gupta, R. K., & Yang, Y. (2017, August). Crystalnest at semeval-2017 task 4: Using sarcasm detection for enhancing sentiment classification and quantification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 626-633).

[230] Rozental, A., & Fleischer, D. (2017). Amobee at SemEval-2017 Task 4: Deep learning system for sentiment detection on Twitter. *arXiv preprint arXiv:1705.01306*.

[231] Wang, M., Chen, S., Xie, Y., & Zhao, L. (2017, August). EICA at SemEval-2017 Task 4: A Simple Convolutional Neural Network for Topic-based Sentiment Classification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 737-740).

[232] Rajendram, S. M., & Mirnalinee, T. T. (2017, August). SSN_MLRG1 at SemEval-2017 Task 4: sentiment analysis in twitter using multi-kernel gaussian process classifier. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 709-712).

[233] Li, Q., Nourbakhsh, A., Liu, X., Fang, R., & Shah, S. (2017, August). funSentiment at SemEval-2017 Task 4: Topic-based message sentiment classification by exploiting word embeddings, text features and target contexts. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 741-746).

[234] Dovdon, E., & Saias, J. (2017, August). ej-sa-2017 at semeval-2017 task 4: Experiments for target oriented sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 644-647).

[235] Laskari, N. K., & Sanampudi, S. K. (2017, August). TWINA at SemEval-2017 Task 4: Twitter sentiment analysis with ensemble gradient boost tree classifier. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 659-663).

[236] Lei, Z., Yang, Y., & Yang, M. (2018, June). SAAN: A sentiment-aware attention network for sentiment analysis. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (pp. 1197-1200).

[237] Yu, L. C., Wang, J., Lai, K. R., & Zhang, X. (2017). Refining word embeddings using intensity scores for sentiment analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(3), 671-681.

[238] Lu, Y., Rao, Y., Yang, J., & Yin, J. (2018, July). Incorporating Lexicons into LSTM for sentiment classification. In *2018 International joint conference on neural networks (IJCNN)* (pp. 1-7). IEEE.

[239] Sadr, H., Pedram, M. M., & Teshnehlab, M. (2019). A robust sentiment analysis method based on sequential combination of convolutional and recursive neural networks. *Neural Processing Letters*, *50*(3), 2745-2761.

[240] Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications*, *72*, 221-230.

[241] Baktha, K., & Tripathy, B. K. (2017, April). Investigation of recurrent neural networks in the field of sentiment analysis. In *2017 International Conference on Communication and Signal Processing (ICCSP)* (pp. 2047-2050). IEEE.

[242] Hiyama, Y., & Yanagimoto, H. (2018). Word polarity attention in sentiment analysis. *Artificial Life and Robotics*, *23*(3), 311-315.

[243] Li, W., Zhu, L., Shi, Y., Guo, K., & Zheng, Y. (2020). User reviews: Sentiment analysis using lexicon integrated two-channel CNN-LSTM family models. *Applied Soft Computing*, 106435.

[244] Hassan, A., & Mahmood, A. (2017, April). Deep learning approach for sentiment analysis of short texts. In *2017 3rd international conference on control, automation and robotics (ICCAR)* (pp. 705-710). IEEE.

[245] Dong, Y., Fu, Y., Wang, L., Chen, Y., Dong, Y., & Li, J. (2020). A sentiment analysis method of capsule network based on BiLSTM. *IEEE Access*, *8*, 37014-37020.

[246] Gams, M., Bohanec, M., & Cestnik, B. (1994, July). A schema for using multiple knowledge. In *Proceedings of the workshop on Computational learning*

theory and natural learning systems (vol. 2): intersections between theory and experiment: intersections between theory and experiment (pp. 157-170).

[247] Dietterich, T. G. (1997). Machine-learning research. *AI magazine*, *18*(4), 97-97.

[248] Asghar, N. (2016). Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*.

[249] Saif, H., Fernandez, M., He, Y., & Alani, H. (2013). Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the STS-Gold.

[250] Kotzias, D., Denil, M., De Freitas, N., & Smyth, P. (2015, August). From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 597-606).

[251] Jianqiang, Z., & Xiaolin, G. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, *5*, 2870-2879.

[252] Hidayatullah, A. F., Ratnasari, C. I., & Wisnugroho, S. (2015). The influence of stemming on Indonesian tweet sentiment analysis. In *Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2015)* (pp. 127-132).

[254] Wahbeh, A., Al-Kabi, M., Al-Radaideh, Q., Al-Shawakfa, E., & Alsmadi, I. (2011). The effect of stemming on arabic text classification: an empirical study. *International Journal of Information Retrieval Research (IJIRR)*, *1*(3), 54-70.

[255] Jain, A. K., & Chandrasekaran, B. (1982). 39 Dimensionality and sample size considerations in pattern recognition practice. *Handbook of statistics*, *2*, 835-855.

[256] Pechenizkiy, M. (2005). Feature extraction for supervised learning in knowledge discovery systems. *Jyväskylä studies in computing*, (56).

[257] Aladjem, M. E. (1994). Multiclass discriminant mappings. *Signal Processing*, *35*(1), 1-18.

[258] Pham, H. N. A., & Triantaphyllou, E. (2008). The impact of overfitting and overgeneralization on the classification accuracy in data mining. In *Soft computing for knowledge discovery and data mining* (pp. 391-431). Springer, Boston, MA.

[259] Bramer, M. (2007). *Principles of data mining* (Vol. 180). London: Springer.

[260] Paltoglou, G., & Thelwall, M. (2010, July). A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 1386-1395).

[261] Tandon, N., & De Melo, G. (2010, July). Information extraction from web-scale n-gram data. In *Web N-gram Workshop* (Vol. 7).

[262] Wilks, Y., & Stevenson, M. (1998). The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Natural Language Engineering*, *4*(2), 135-143.

[263] Riloff, E., & Wiebe, J. (2003). Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing* (pp. 105-112).

[264] Na, J. C., Sui, H., Khoo, C., Chan, S., & Zhou, Y. (2004). Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. *Advances in Knowledge Organization*, *9*, 49-54.

[265] Benamara, F., Cesarano, C., Picariello, A., Recupero, D. R., & Subrahmanian, V. S. (2007). Sentiment analysis: Adjectives and adverbs are better than adjectives alone. *ICWSM*, *7*, 203-206.

[266] Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank.

[267] Ahire, S. (2014). A survey of sentiment lexicons. *Computer Science and Engineering IIT Bombay, Bombay*.

[268] Sun, S., Luo, C., & Chen, J. (2017). A review of natural language processing techniques for opinion mining systems. *Information fusion*, *36*, 10-25.

[269] Giachanou, A., & Crestani, F. (2016). Like it or not: A survey of twitter sentiment analysis methods. *ACM Computing Surveys (CSUR)*, *49*(2), 1-41.

[270] Lo, S. L., Cambria, E., Chiong, R., & Cornforth, D. (2017). Multilingual sentiment analysis: from formal to informal and scarce resource languages. *Artificial Intelligence Review*, *48*(4), 499-527.

[271] Le, H. Q., Tran, M. V., Dang, T. H., & Collier, N. (2015). The UET-CAM system in the BioCreAtIvE V CDR task. In *Fifth BioCreative challenge evaluation workshop* (pp. 208-213).

[272] Lewis, D. D. (1998, April). Naive (Bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning* (pp. 4-15). Springer, Berlin, Heidelberg.

[273] Jiang, S., Pang, G., Wu, M., & Kuang, L. (2012). An improved K-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications*, *39*(1), 1503-1509.

[274] Tan, S. (2005). Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, *28*(4), 667-671.

[275] Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (Vol. 398). John Wiley & Sons.

[276] Yasen, M., & Tedmori, S. (2019, April). Movies Reviews sentiment analysis and classification. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)* (pp. 860-865). IEEE.

[277] QUINLAU, R. (1986). Induction of decision trees. *Machine learning*, *1*(1), S1-S106.

[278] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

[279] Kirasich, K., Smith, T., & Sadler, B. (2018). Random Forest vs logistic regression: binary classification for heterogeneous datasets. *SMU Data Science Review*, *1*(3), 9.

[280] Xu, X., & Frank, E. (2004, May). Logistic regression and boosting for labeled bags of instances. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 272-281). Springer, Berlin, Heidelberg.

[281] Sharma, A., & Dey, S. (2013). A boosted SVM based sentiment analysis approach for online opinionated text. In *Proceedings of the 2013 research in adaptive and convergent systems* (pp. 28-34).

[282] Breiman, L. (1997). *Arcing the edge* (Vol. 7). Technical Report 486, Statistics Department, University of California at Berkeley.

[283] Budur, E., Lee, S., & Kong, V. S. (2015). Structural analysis of criminal network and predicting hidden links using machine learning. *arXiv preprint arXiv:1507.05739*.

[284] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

[285] Ren, X., Guo, H., Li, S., Wang, S., & Li, J. (2017, August). A novel image classification method with CNN-XGBoost model. In *International Workshop on Digital Watermarking* (pp. 378-390). Springer, Cham.

[286] Kuncheva, L. I. (2014). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

[287] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1-47.

[288] Nassirtoussi, A. K., Aghabozorgi, S., Wah, T. Y., & Ngo, D. C. L. (2014). Text mining for market prediction: A systematic review. *Expert Systems with Applications*, *41*(16), 7653-7670.

[289] Zunic, A., Corcoran, P., & Spasic, I. (2020). Sentiment analysis in health and well-being: systematic review. *JMIR medical informatics*, *8*(1), e16023.

[290] Ghosh, M., & Sanyal, G. (2018). Performance assessment of multiple classifiers based on ensemble feature selection scheme for sentiment analysis. *Applied Computational Intelligence and Soft Computing*, *2018*.

[291] Zhang, S. T., Wang, F. F., Duo, F., & Zhang, J. L. (2018). Research on the Majority Decision Algorithm based on WeChat sentiment classification. *Journal of Intelligent & Fuzzy Systems*, *35*(3), 2975-2984.

[292] Dhaliwal, S. S., Nahid, A. A., & Abbas, R. (2018). Effective intrusion detection system using XGBoost. *Information*, *9*(7), 149.

[293] Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing*, *5*(2), 221.

[294] Çığşar, B., & Ünal, D. (2019). Comparison of data mining classification algorithms determining the default risk. *Scientific Programming*, *2019*.

[295] Korovesis, K. (2018). Sentiment analysis for tweets.

[296] Chen, X., Rao, Y., Xie, H., Wang, F. L., Zhao, Y., & Yin, J. (2019). Sentiment classification using negative and intensive sentiment supplement information. *Data Science and Engineering*, *4*(2), 109-118.

[297] Huang, M., Xie, H., Rao, Y., Liu, Y., Poon, L. K., & Wang, F. L. (2020). Lexicon-based sentiment convolutional neural networks for online review analysis. *IEEE Transactions on Affective Computing*.

[298] Xu, Z., Fu, Y., Chen, X., Rao, Y., Xie, H., Wang, F. L., & Peng, Y. (2018, July). Sentiment classification via supplementary information modeling. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data* (pp. 54-62). Springer, Cham.

[299] Saif, H., He, Y., Fernandez, M., & Alani, H. (2016). Contextual semantics for sentiment analysis of Twitter. *Information Processing & Management*, *52*(1), 5-19.

[300] Kermani, F. Z., Sadeghi, F., & Eslami, E. (2020). Solving the twitter sentiment analysis problem based on a machine learning-based approach. *Evolutionary Intelligence*, *13*(3), 381-398.

[301] Troussas, C., Krouska, A., & Virvou, M. (2016, July). Evaluation of ensemble-based sentiment classifiers for Twitter data. In *2016 7th international conference on information, intelligence, systems & applications (IISA)* (pp. 1-6). IEEE.

[302] Yan, Y., Yang, H., & Wang, H. M. (2017, July). Two simple and effective ensemble classifiers for Twitter sentiment analysis. In *2017 Computing Conference* (pp. 1386-1393). IEEE.

[303] Kauer, A. U., & Moreira, V. P. (2016). Using information retrieval for sentiment polarity prediction. *Expert Systems with Applications*, *61*, 282-289.

[304] Keshavarz, H., & Abadeh, M. S. (2017). ALGA: Adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs. *Knowledge-Based Systems*, *122*, 1-16.

[305] Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications*, *72*, 221-230.

[306] Giménez, M., Palanca, J., & Botti, V. (2020). Semantic-based padding in convolutional neural networks for improving the performance in natural language processing. A case of study in sentiment analysis. *Neurocomputing*, *378*, 315-323.

[307] Xu, Y., Li, L., Gao, H., Hei, L., Li, R., & Wang, Y. (2021). Sentiment classification with adversarial learning and attention mechanism. *Computational Intelligence*, *37*(2), 774-798.

[308] Tripathi, S., Singh, C., Kumar, A., Pandey, C., & Jain, N. (2019, June). Bidirectional transformer based multi-task learning for natural language understanding. In *International Conference on Applications of Natural Language to Information Systems* (pp. 54-65). Springer, Cham.

[309] Park, D., & Ahn, C. W. (2019). Self-Supervised Contextual Data Augmentation for Natural Language Processing. *Symmetry*, *11*(11), 1393.

[310] Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011, July). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 151-161). Association for Computational Linguistics.

[311] Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[312] Socher, R., Huval, B., Manning, C. D., & Ng, A. Y. (2012, July). Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 1201-1211). Association for Computational Linguistics.

[313] Guerreiro, J., & Rita, P. (2020). How to predict explicit recommendations in online reviews using text mining and sentiment analysis. *Journal of Hospitality and Tourism Management*, *43*, 269-272.

[314] Potts, C., Wu, Z., Geiger, A., & Kiela, D. (2020). Dynasent: A dynamic benchmark for sentiment analysis. *arXiv preprint arXiv:2012.15349*.

[315] Chen, R. C. (2019). User rating classification via deep belief network learning and sentiment analysis. *IEEE Transactions on Computational Social Systems*, *6*(3), 535-546.

[316] Hemalatha, S., & Ramathmika, R. (2019, May). Sentiment Analysis of Yelp Reviews by Machine Learning. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)* (pp. 700-704). IEEE.

[317] Rathee, N., Joshi, N., & Kaur, J. (2018, June). Sentiment Analysis Using Machine Learning Techniques on Python. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 779-785). IEEE.

[318] Ahmed, B. H., & Ghabayen, A. S. (2020). Review rating prediction framework using deep learning. *Journal of Ambient Intelligence and Humanized Computing*, 1-10.

[319] Zhu, Y., Moh, M., & Moh, T. S. (2016, December). Multi-layer text classification with voting for consumer reviews. In *2016 IEEE International Conference on Big Data (Big Data)* (pp. 1991-1999). IEEE.

[320] Singh, S. K., & Sachan, M. K. (2019). SentiVerb system: classification of social media text using sentiment analysis. *Multimedia Tools and Applications*, *78*(22), 32109-32136.

[321] Carvalho, F., Rodrigues, R. G., & Guedes, G. P. (2018, October). LIWBC: a bigram algorithm to enhance results in polarity classification. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web* (pp. 419-422).

[322] Korovkinas, K., Danėnas, P., & Garšva, G. (2017). SVM and Na¨ ýve Bayes Classification Ensemble Method for Sentiment Analysis. *Baltic Journal of Modern Computing*, *5*(4), 398-409.

[323] Hama Aziz, R. H., & Dimililer, N. (2021). SentiXGboost: enhanced sentiment analysis in social media posts with ensemble XGBoost classifier. *Journal of the Chinese Institute of Engineers*, 1-11.

[324] Sadr, H., Solimandarabi, M. N., Pedram, M. M., & Teshnehlab, M. (2021). A Novel Deep Learning Method for Textual Sentiment Analysis. *arXiv preprint arXiv:2102.11651*.

[325] Kasri, M., Birjali, M., & Beni-Hssane, A. (2021). Word2Sent: A new learning sentiment-embedding model with low dimension for sentence level sentiment classification. *Concurrency and Computation: Practice and Experience*, *33*(9), e6149.

APPENDICES

Appendix A: Penn Treebank PoS Tagset

Table A.1: List of P Penn Treebank PoS Tagset used

Tag	Description	Example	Tag	Description	Example
CC	coord. conjunction	*and, or*	RB	adverb	*extremely*
CD	cardinal number	*one, two*	RBR	adverb, comparative	*never*
DT	determiner	*a, the*	RBS	adverb, superlative	*fastest*
EX	existential there	*there*	RP	particle	*up, off*
FW	foreign word	*noire*	SYM	symbol	+, %
IN	preposition or sub-conjunction	*of, in*	TO	"to"	*to*
JJ	adjective	*small*	UH	interjection	*oops, oh*
JJR	adject., comparative	*smaller*	VB	verb, base form	*fly*
JJS	adject., superlative	*smallest*	VBD	verb, past tense	*flew*
LS	list item marker	*1, one*	VBG	verb, gerund	*flying*
MD	modal	*can, could*	VBN	verb, past participle	*flown*
NN	noun, singular or mass	*dog*	VBP	verb, non-3sg pres	*fly*
NNS	noun, plural	*dogs*	VBZ	verb, 3sg pres	*flies*
NNP	proper noun, sing.	*London*	WDT	wh-determiner	*which, that*
NNPS	proper noun, plural	*Azores*	WP	wh-pronoun	*who, what*
PDT	predeterminer	*both, lot of*	WP$	possessive wh-	*whose*
POS	possessive ending	*'s*	WRB	wh-adverb	*where, how*
PRP	personal pronoun	*he, she*			

Appendix B: List of Stop Words

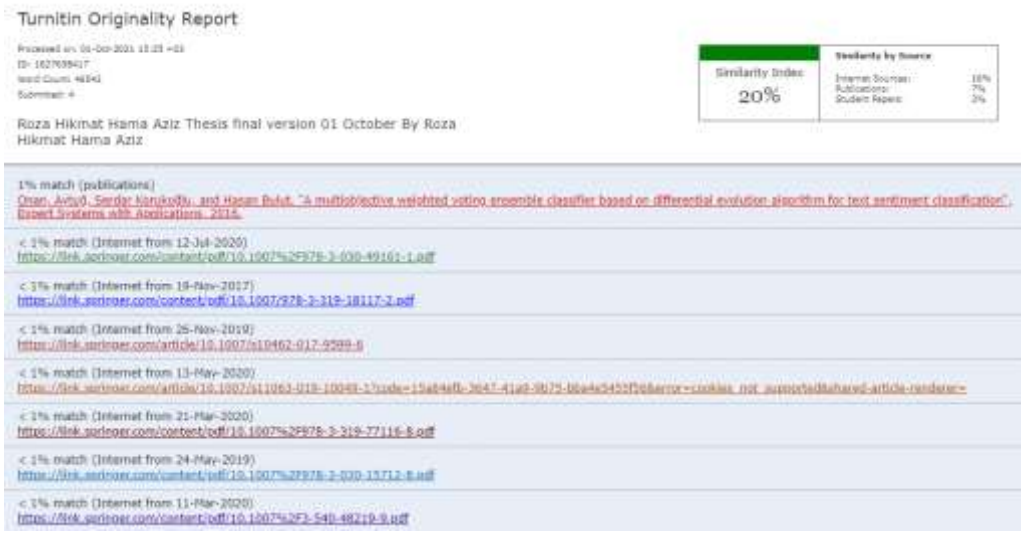
Table B.1: List of stop words used

a	available	containing	fifth	him
able	away	contains	first	himself
about	awfully	correspondin	five	his
above	b	g	followed	hither
according	be	could	following	hopefully
accordingly	became	course	follows	how
across	because	currently	for	howbeit
actually	become	d	former	however
after	becomes	definitely	formerly	i
afterward	becoming	described	forth	ie
again	been	despite	four	if
all	before	did	from	ignored
allow	beforehand	different	further	immediate
allows	behind	do	furthermore	in
almost	being	does	g	inasmuch
alone	believe	doing	get	inc
along	below	done	gets	indeed
already	beside	down	getting	indicate
also	besides	downwards	given	indicated
although	best	during	gives	indicates
always	better	e	go	inner
am	between	each	goes	insofar
among	beyond	edu	going	instead
amongst	both	eg	gone	into
an	brief	eight	got	inward
and	but	either	gotten	is
another	by	else	greetings	it
any	c	elsewhere	h	its
anybody	came	enough	had	itself
anyhow	can	entirely	happens	j
anyone	cannot	especially	hardly	just
anything	cant	et	has	k
anyway	cause	etc	have	keep
anyways	causes	even	having	keeps
anywhere	certain	ever	he	kept
apart	certainly	every	hello	know
appear	changes	everybody	help	knows
appreciate	clearly	everyone	hence	known
appropriate	co	everything	her	l
are	com	everywhere	here	last
around	come	ex	hereafter	lately
as	comes	exactly	hereby	later
aside	concerning	example	herein	latter
ask	consequently	except	hereupon	latterly
asking	consider	f	hers	least
associated	considering	far	herself	less
at	contain	few	hi	lest

let
little
you'll
look
looking
looks
ltd
m
mainly
many
may
maybe
me
mean
meanwhile
merely
might
more
moreover
most
mostly
much
must
my
myself
n
oh
ok
okay
old
on
once
one
ones
only
onto
or
other
others
otherwise
ought
our
ours
ourselves
out
outside
over
overall
own
p
particular
particularly
right
s
said
same
saw
say
saying
says
second
secondly
see
seeing
seem
seemed
seeming
seems
seen
self
selves
sensible
sent
serious
seriously
seven
several
shall
t
take
taken
tell
tends
th
than
thank
thanks
thanx
that
thats
the
their
theirs
them
themselves
then
thence
there
thereafter
thereby
therefore
therein
theres
thereupon
trying
twice
two
u
un
under
unfortunately
unless
unlikely
until
unto
up
upon
us
use
used
useful
uses
using
usually
uucp
v
value
various
you've
very
name
namely
nd
near
nearly
necessary
need
needs
neither
never
nevertheless
new
next
nine
nobody
normally
novel
now
nowhere
o
obviously
of
off
often
per
perhaps
placed
please
plus
possible
presumably
probably
provides
q
que
quite
qv
r
rather
rd
re
really
reasonably
regarding
regardless
regards
relatively
respectively
she
should
since
six
so
some
somebody
somehow
someone
something
sometime
sometimes
somewhat
somewhere
soon
sorry
specified
specify
specifying
still
sub
such
sup
sure
these
they
think
third
this
thorough
thoroughly
those
though
three
through
throughout
thru
thus
to
together
too
took
toward
towards
tried
tries
truly
try
via
viz
vs
w
want
wants
was
way
we
welcome
well
went
were
what
whatever
when
whence
whenever
where
whereafter
whereas
whereby
wherein
whereupon

wherever	whole	with	y	yourselves
whether	whom	within	yes	z
which	whose	without	yet	zero
while	why	wonder	you	
whither	will	would	your	
who	willing	would	yours	
whoever	wish	x	yourself	

Appendix C: Similarity Report



Turnitin Originality Report

Processed on: 01-Oct-2021 13:25 +03
ID: 1627638417
Word Count: 46843
Submitted: 4

Roza Hikmat Hama Aziz Thesis final version 01 October By Roza Hikmat Hama Aziz

Similarity Index	Similarity by Source	
20%	Internet Sources:	16%
	Publications:	7%
	Student Papers:	3%

1% match (publications)
Onan, Aytuğ, Serdar Korukoğlu, and Hasan Bulut. "A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification", Expert Systems with Applications, 2016.

< 1% match (Internet from 12-Jul-2020)
https://link.springer.com/content/pdf/10.1007%2F978-3-030-49161-1.pdf

< 1% match (Internet from 19-Nov-2017)
https://link.springer.com/content/pdf/10.1007/978-3-319-18117-2.pdf

< 1% match (Internet from 26-Nov-2019)
https://link.springer.com/article/10.1007/s10462-017-9599-6

< 1% match (Internet from 13-May-2020)
https://link.springer.com/article/10.1007/s11063-019-10049-1?code=15a84efb-3647-41a9-9b75-0ba4e5451f5b&error=cookies_not_supported&shared-article-renderer

< 1% match (Internet from 21-Mar-2020)
https://link.springer.com/content/pdf/10.1007%2F978-3-319-77116-8.pdf

< 1% match (Internet from 24-May-2019)
https://link.springer.com/content/pdf/10.1007%2F978-3-030-15712-8.pdf

< 1% match (Internet from 11-Mar-2020)
https://link.springer.com/content/pdf/10.1007%2F3-540-48219-9.pdf