CONTEXT- AND SENTIMENT-AWARE MACHINE LEARNING MODELS FOR SENTIMENT ANALYSIS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

FİRDEVSİ AYÇA DENİZ-KIZILÖZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

JANUARY 2023

Approval of the thesis:

CONTEXT- AND SENTIMENT-AWARE MACHINE LEARNING MODELS FOR SENTIMENT ANALYSIS

submitted by **FİRDEVSİ AYÇA DENİZ-KIZILÖZ** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Pelin Angın
Supervisor, **Computer Engineering, METU**

Assist. Prof. Dr. Merih Angın
Co-supervisor, **International Relations, Koç University**

Examining Committee Members:

Prof. Dr. Pınar Karagöz
Computer Engineering, METU

Assoc. Prof. Dr. Pelin Angın
Computer Engineering, METU

Prof. Dr. İlyas Çiçekli
Computer Engineering, Hacettepe University

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering, METU

Assoc. Prof. Dr. Tansel Dökeroğlu
Software Engineering, Çankaya University

Date:24.01.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Firdevsi Ayça Deniz-Kızılöz

Signature :

ABSTRACT

CONTEXT- AND SENTIMENT-AWARE MACHINE LEARNING MODELS FOR SENTIMENT ANALYSIS

Deniz-Kızılöz, Firdevsi Ayça
Ph.D., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Pelin Angın
Co-Supervisor: Assist. Prof. Dr. Merih Angın

January 2023, 124 pages

With the advances in information technologies, the amount of available data on web sources where people express their opinions increases continually. Sentiment analysis supports decision-makers in gaining insights from massive heaps of data. It has gained much attraction recently as it has proven to be a practical tool in a wide range of areas, including monitoring public opinion. Nevertheless, sentiment analysis research is still facing some challenges. One of the main challenges is the irrelevant and redundant features in the data. Such features not only increase the search space enormously but also disrupt the context awareness of the model. Another main challenge is the lack of domain-agnostic models for the sentiment analysis tasks as an existing model may not be the best fit for another domain in terms of context. Although deep learning models provide high-performance results, they require a massive amount of labeled data. However, obtaining a sufficient amount of labeled data is often impractical.

In this thesis, we propose four models to remedy the aforementioned drawbacks. Our first model extracts the most informative features in the data for sentiment analysis.

The second one constructs a context-refined word embedding model. The third model transfers the knowledge in pre-trained models to a new domain without the necessity of labeled data. The last one is a feature ensemble model that builds a pool of varying features for sentiment analysis. To verify the effectiveness of our models, we held extensive experiments on three benchmark datasets. Moreover, we introduced two novel datasets consisting of thousands of sentence and sentiment class pairs. Experiment results demonstrated that the proposed models yield performance improvements.

Keywords: natural language processing, sentiment analysis, machine learning

ÖZ

DUYGU ANALİZİ İÇİN BAĞLAM VE DUYGUYA DUYARLI MAKİNE ÖĞRENMESİ MODELLERİ

Deniz-Kızılöz, Firdevsi Ayça
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Pelin Angın
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi. Merih Angın

Ocak 2023 , 124 sayfa

Bilişim teknolojilerindeki gelişmelerle birlikte insanların görüşlerini ifade ettikleri web kaynaklarındaki mevcut veri miktarı sürekli olarak artmaktadır. Duygu analizi, karar merciilerin büyük veri yığınlarından içgörüler elde etmelerini destekler. Kamuoyunun takip edilmesi de dahil olmak üzere çok çeşitli alanlarda pratik bir araç olduğunu kanıtladığı için son zamanlarda çok ilgi görmüştür. Buna rağmen, duygu analizi araştırması hala bazı zorluklarla karşı karşıyadır. Ana zorluklardan biri, verilerdeki ilgisiz ve gereksiz özelliklerdir. Bu tür özellikler, yalnızca arama alanını büyük ölçüde artırmakla kalmaz, aynı zamanda modelin bağlam farkındalığını da bozar. Diğer bir temel zorluk, duygu analizi görevleri için etki alanından bağımsız modellerin bulunmamasıdır, çünkü mevcut bir model bağlam açısından başka bir alan için uygun olmayabilir. Derin öğrenme modelleri yüksek performanslı sonuçlar sağlasa da çok büyük miktarda etiketlenmiş veri gerektirirler. Bununla birlikte, yeterli miktarda etiketlenmiş veri elde etmek genellikle kolay bir süreç değildir.

Bu tezde, yukarıda belirtilen dezavantajları gidermek için dört model öneriyoruz. İlk modelimiz, duygu analizi için verilerdeki en bilgilendirici özellikleri çıkarır. İkincisi, bağlamla rafine edilmiş bir kelime gömme modeli oluşturur. Üçüncü model, etiketlenmiş verilere ihtiyaç duymadan önceden eğitilmiş modellerdeki bilgiyi yeni bir alana aktarır. Sonuncusu, duygu analizi için çeşitli özelliklerden oluşan bir havuz oluşturan bir özellik topluluğu modelidir. Modellerimizin etkinliğini doğrulamak için üç adet tanınmış veri seti üzerinde kapsamlı deneyler yaptık. Ayrıca, binlerce cümle ve duygu sınıfı çiftinden oluşan iki yeni veri seti oluşturduk. Deney sonuçları, önerilen modellerin performans iyileştirmeleri sağladığını göstermiştir.

Anahtar Kelimeler: doğal dil işleme, duygu analizi, makine öğrenmesi

ACKNOWLEDGMENTS

Sincere thanks to my supervisors, Assoc. Prof. Dr. Pelin Angın and Assist. Prof. Dr. Merih Angın, for their guidance.

Many thanks to my thesis committee members, Prof. Dr. Pınar Karagöz and Prof. Dr. İlyas Çiçekli, and the jury members, Prof. Dr. İsmail Hakkı Toroslu and Assoc. Prof. Dr. Tansel Dökeroğlu, for their reviews.

Huge thanks to all researchers in the MA-Computational Social Science Lab for their efforts in labeling the datasets.

Thanks to TÜBİTAK for providing me with financial support as part of the Project 118C309 titled "IMF Decides, Machine Learns: An AI Approach to International Organizations".

Thanks to Google for giving me the chance to work as a software engineer while my Ph.D. study was ongoing.

Thanks to TÜBİTAK ULAKBİM for providing high-performance computing resources through TRUBA.

Special thanks to my mother/home, Emine Deniz, for providing me with continuous encouragement, my father/mentor, Hamdi Deniz, for being my source of inspiration, and my love/best friend, Hakan Ezgi Kızılöz, for the endless emotional support and technical discussions.

This accomplishment would not have been possible without you all.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

BERT	Bidirectional Encoder Representations from Transformers
BoW	Bag-of-Words
GloVe	Global Vectors for Word Representation
IG	Information Gain
IGF	Information Gain Filtering
IMF	International Monetary Fund dataset
LR	Logistic Regression
MR	Movie Reviews dataset
NLP	Natural Language Processing
NSGA-II	Non-dominated Sorting Genetic Algorithm II
PCA	Principal Component Analysis
S140	Sentiment 140 dataset
SST	Stanford Sentiment Treebank dataset
SVM	Support Vector Machines
TF-IDF	Term Frequency–Inverse Document Frequency
WHO	World Health Organization dataset

CHAPTER 1

INTRODUCTION

1.1 Overview

The significant advances in data storage, communication, and processing technologies in recent years have given rise to the big data era, with a plethora of information flowing in from various data sources at high speeds. The high volume of generated data is useful for providing insightful information to decision-makers in various domains. However, it needs to be appropriately exploited with respect to the requirements.

With the evolution of Internet-based applications, people increasingly express their opinions on social networks and other web sources. Sentiment analysis, a subfield of Natural Language Processing (NLP), is one of the powerful tools that play an essential role in analyzing public opinion, which guides businesses and researchers in their decision-making processes [1, 2, 3]. It is also known as opinion mining since it aims to identify the sentimental polarity of given content by providing automated extraction of subjective opinions [4, 5]. It may be used to understand the opinions or sentiments of people towards a specific entity, such as a product, a service, or an organization [6]. Accordingly, it provides low-cost solutions with respect to monitoring public opinion in terms of preferences [7]. As a result, it has attracted a wide range of communities, such as researchers, organizations, governments, and businesses.

Sentiment analysis has been gaining more attention in the past two decades, as it is a significant element of many real-world applications, including but not limited to recommendation systems [8, 6, 9], analysis of product reviews [10, 11, 12], quality assessment [13, 14], terrorist organization tracking [15], detection and analysis of critical events [16, 17, 18, 19], real-time observation of public opinion [20], election prediction [21], social media monitoring [22, 23], finance [24, 25, 26], tourism [27], and healthcare systems [28, 29, 30, 31, 32]. Therefore, it remains an effective tool to convey information between policymakers and public opinion [7, 33].

Sentiment analysis can be defined as a polarity classification problem [34, 35]. In this problem, there might be a different number of classes (binary or multi-class) that represents varying degrees of sentiment scores (positive, negative, or neutral) depending on the domain.

According to Liu [4], a sentiment or an opinion comprises five elements: entity, aspect, sentiment, author, and time. Sentiment analysis can be considered the art of gathering insights from unstructured data collected from several sources. It can be applied in four levels [36]: document-, sentence-, aspect-, or comparative-based. Recently, aspect-based sentiment analysis has gained attention as a text may contain multiple aspects having different sentiments [37, 38].

At a high level, there exist three approaches to address the sentiment analysis task [39]: lexicon-based, machine learning-based, and hybrid.

Lexicon-based approaches rely on dictionaries that contain token and sentiment pairs [40]. This way, the sentiment of a sentence can be calculated using the sentiments of each word, combined with different techniques such as aggregation (e.g. majority voting). Although lexicon-based methods are easy to apply, and hence offer a simple solution in a timely manner, they suffer from the lack of domain-specific dictionaries [41].

Machine learning-based approaches examine the historical data to make predictions on the new data. They are generally grouped under two main categories: supervised and unsupervised. Supervised ones predict the unknown label by training on the labeled data. Fundamentally, they utilize a function, $y = f(x)$, that maps the data features (x) to a sentiment value (y). In contrast, the unsupervised ones discover hidden patterns in the data by analyzing the association within the features. In general, the supervised methods perform better than the unsupervised ones. However, obtaining labeled data regarding the target domain may be a challenging task.

Even though machine learning-based approaches do not require external dictionaries like lexicon-based approaches, they require feature engineering for NLP tasks [42], regardless of the selection of supervised or unsupervised methods. More specifically, free-form textual data must be translated into a standard representation (vectorization) that the machine learning techniques can interpret. It is crucial to extract features that represent the data properly as it affects the prediction quality [43]. On the other hand, feature engineering is a labor-intensive task, and it should be obviated where possible by discovering valuable information automatically from the data itself [44]. Once the feature engineering step is complete, machine learning-based approaches generally perform better than lexicon-based ones [45].

Hybrid approaches combine lexicon-based and machine learning-based methods to take the best advantage of both methods for sentiment analysis [46].

Recent research on sentiment analysis has mainly focused on deep learning architectures [47, 48, 49, 50, 51, 52]. The advances in hardware technologies and the increase in easily accessible data amount pave the way for deep learning studies upon the outstanding success of neural networks. These architectures provide better performance than legacy methods, as they provide semantic information intrinsically through their hierarchical learning process [53]. Moreover, deep learning models are generally reusable in multi tasks such as question answering or sentiment classification. They generally contain hundreds of layers, e.g. one of the well-known language models, Bidirectional Encoder Representations from Transformers (BERT), had 340 million parameters [54]. Additionally, they require a huge amount of training data to create accurate models, e.g. BERT was trained on 3.3 billion tokens. Accordingly, this amount of data and models need a huge computational power. Nevertheless, they have limitations. First of all, they are very data-hungry. To learn from data, they require very high computation resources. Other than that, they can be easily fooled by adversarial examples. In general, they are considered uninterpretable black boxes, which means they lack transparency. And finally, it is not easy to incorporate prior knowledge into the deep learning models.

Besides, expert knowledge may be required to determine the ideal parameters of a deep learning model. However, in some domains, such as hate speech detection, the acquisition of expert knowledge may be lacking. For example, Madukwe et al. [55] proposed utilizing a genetic algorithm as a remedy to decide the optimal architecture and select the best set of hyperparameters for fine-tuning BERT.

1.2 Challenges and Opportunities

There exist different application strategies for sentiment analysis tasks. These strategies have their own challenges, hence, their own opportunities for improvement. In this section, we spotlight various challenges of sentiment analysis.

Challenge 1: Sentiment analysis faces challenges due to the existence of slang words, spelling mistakes, and ironic remarks in documents [56]. One of the main challenges in sentiment classification is the high amount of data that contains irrelevant or redundant features [57], which adversely affect the performance of machine learning models [58].
Opportunity 1: Feature selection is one of the effective preprocessing techniques to eliminate features that have low or no contribution to the classification task [59]. It plays a major role in minimizing the data with the most informative representatives, especially for the domains having a huge amount of features such as NLP tasks.

Challenge 2: When applying a machine learning-based sentiment analysis model, it is possible to vectorize the free-form textual data using word embeddings. There exists a list of word embedding models pre-trained on large datasets such as Wikipedia[1], which are commonly used in NLP tasks. However, these pre-trained word embeddings may not be the best fit for the data in terms of the context when a domain-specific sentiment analysis task such as movie reviews is in question [60]. Moreover, especially for the sentiment analysis task, word embedding models such as Global Vectors for Word Representation (GloVe) lack the sentiment information of the words and the context they appear. Words having dissimilar sentiments, e.g. happy-sad or good-bad, may have similar semantics and consequently have similar vector representations [61].
Opportunity 2: Sentiment analysis may benefit from infusing different word embed-

[1] Available at http://dumps.wikimedia.org/enwiki/latest/

ding models with sentiment information. It has also been shown that such hybrid methods applied for word embeddings perform better than single ones [62].

Challenge 3: There exist pre-trained language models[2] that have been fine-tuned for the sentiment analysis task using well-known public datasets. Therefore, they can determine the sentiments of the sentences in these datasets well. However, the classification performance drops due to the rule of generalization on less popular domains or datasets. Yet, it may still provide an acceptable accuracy depending on the dataset. *Opportunity 3:* In general, pre-trained models can correctly identify the sentiment of the content with high accuracy if its confidence score is high. Diffusing the information inherited from high-confidence predictions could be valuable for the model.

To sum up, sentiment classification can be considered a domain-dependent task since the content's sentiment may contradict in different contexts [63]. Therefore, a model trained for a domain may not be suitable to be applied to another domain. However, obtaining a sufficient amount of labeled data for every domain is often impractical. Therefore, the research on domain-independent methods becomes worthwhile to handle the limited resources.

1.3 Objectives

Although there exist different word representation methods, language models, and sentiment dictionaries, which are separately reported to be effective in certain domains, there is no domain-independent method that integrates the context and sentiment information efficiently for the sentiment analysis task, handles unknown words, and does not require hyperparameter tuning.

In light of the information provided in the previous section, we set our first objective as systematically investigating the mentioned opportunities to alleviate their associated challenges. Then, we aim to leverage the information gained from these investigations to build domain-independent models for the sentiment analysis task that incorporate the context and sentiment information in various ways.

[2] Available at `https://huggingface.co/`

1.4 Contributions

Our contributions are as follows:

- We propose a new hybrid multiobjective feature selection model for the sentiment analysis task, which harnesses the power of an entropy-based metric, i.e., Information Gain, and an evolutionary algorithm, i.e., Non-dominated Sorting Genetic Algorithm II (NSGA-II). Experiments with different machine learning and feature extraction techniques on varying types of datasets demonstrate that our proposed model improves the learning performance of the sentiment analysis task considerably.
- In an effort to address the shortcomings of existing word embeddings for sentiment analysis tasks, we propose a model that effectively combines context and sentiment information when building a text representation. The proposed model refines word vectors with immediate context information without relying on any specific domain and can be employed with any pre-trained word embedding models. It further integrates sentiment scores obtained using a lexicon-based method when building the final word vectors of sentences to be classified. The model achieves significantly higher sentiment classification accuracy than baseline word embedding models commonly used in sentiment analysis.
- To alleviate the drawbacks of available pre-trained models on less popular datasets, we build a sentiment classification method that does not depend on feature engineering and works domain-independent. The method leverages existing pre-trained BERT models without relying on its training domain. We find out the sentences that the pre-trained model is the most confident with and propagate its classification result to other sentences with respect to how similar those sentences are to the confident ones. Our model does not use the actual label during the classification phase; hence, it is also applicable to datasets with no labeled instances. The ablation study results confirm that our method enhances the performance of unsupervised sentiment analysis.
- We propose a novel feature ensemble model for sentiment analysis that generates a feature pool of distinct types. Our model exploits the intrinsic properties

of the data with the help of different techniques. Fundamentally, we first extract baseline features and apply a feature selection method to remove the irrelevant and redundant data. By using these fine-grained features, we generate a graph representation of the word relationships to capture the context of the words. Then, we project the graph vertexes onto a low-dimensional space to feed them into our machine learning algorithm. In addition to these features, we include classification results of a pre-trained language model as another feature to take advantage of common existing knowledge on NLP. Moreover, we include sentiment polarity scores of the content into our feature pool to include the valence information. The final model becomes highly portable to other domains and languages as it does not require any linguistic or domain knowledge. Comprehensive experiments on datasets with different domains present the efficiency of our proposed model.

Implementations of the models are publicly available on GitHub[3].

1.5 Paper Organization

The rest of this thesis is organized as follows.

In Chapter 2, we provide background information about the methodologies utilized in the proposed models in two sections: lexicon- and machine learning-based sentiment analysis. We further explain the second part in three subsections: text representation, pre-trained language models, and feature selection.

In Chapter 3, we describe the experimental environment. First, we introduce the datasets utilized in the experiments. Then, we share the preprocessing steps employed on the datasets. After that, we provide the applied machine learning techniques and the evaluation metric. Finally, we give the experimental settings.

We present our proposed models in Chapters 4 to 7 as follows:

Chapter 4: Feature selection for sentiment analysis

[3] https://github.com/faycadnz

Chapter 5: Enhancing word embeddings for sentiment analysis
Chapter 6: Unsupervised learning for sentiment analysis
Chapter 7: Feature ensemble model for sentiment analysis

Each chapter mainly consists of three sections. First, we provide related studies that focus on the main subject of the model. Then, we describe the proposed model in detail. In the final section, we provide parameter settings specific to that study, followed by the experiment results along with discussions.

In the final chapter, Chapter 8, we share concluding remarks and potential directions for future work.

CHAPTER 2

BACKGROUND

In this chapter, we provide background information on the technologies utilized in various steps of this thesis study.

2.1 Lexicon-Based Sentiment Analysis

The lexicon-based approaches calculate the sentiment from the semantic orientation or polarity of the phrases. Therefore, they depend on dictionaries (token-sentiment pairs) that are created manually or automatically using seed phrases. The sentiment of a sentence can be calculated using the sentiments of each word, combined with different techniques such as aggregation (e.g. majority voting). Some of the well-known lexicon-based sentiment analysis tools are SentiStrength [64], SentiWordNet [65], and VADER (Valence Aware Dictionary and sEntiment Reasoner) [66].

In two of the proposed models, we utilized sentiment scores provided by VADER[1]. It calculates four ratio values (positive, compound, neutral, and negative) for a given text. A sample output is as follows: 'positive': 0.706, 'compound': 0.9469, 'neutral': 0.294, 'negative': 0.0. The calculation of the compound score consists of three steps. First, the valence scores of all words in the text are summed. Then, the result is adjusted using different rules, e.g., the existence of a booster word increases or decreases the intensity of the score. Finally, the score is normalized between -1 and 1, which indicates strongly negative and strongly positive, respectively. As suggested by the original paper, we set the label of the content as positive when the compound score is greater than 0.05, we set it as negative when it is less than -0.05, and the rest

[1] `https://github.com/cjhutto/vaderSentiment`

becomes neutral.

Lexicon-based approaches generally provide an initial result in a short time; however, machine learning-based approaches perform better in terms of classification accuracy [45].

2.2 Machine Learning-Based Sentiment Analysis

For efficient sentiment classification, it is important to obtain a low-dimensional and non-sparse vector representation to feed the machine learning models.

2.2.1 Text Representation

Supervised learning algorithms require feature engineering for the training part of their nature. Therefore, arbitrary data (or content) must be translated into a vector representation so that the machine learning techniques can interpret it. There exist various techniques for representing the text to be classified in sentiment analysis. One-hot encoding is one of the primitive text representation techniques. It converts arbitrary texts into fixed-length vectors. It assigns a number to each unique word in the text corpora, which makes the length of each word vector equal to the number of unique words. To represent a piece of text, 1 is assigned to the corresponding word of the vector if a word is present in the text, and 0 otherwise. Bag-of-Words (BoW) is another basic and well-known text representation technique [67]. It builds on one-hot encoding by storing the count of the words in the text rather than only containing the existence information. These methods do not consider word ordering when generating the features. Hence, the syntactic and semantic relationships are lost [68]. However, word order can be effective when deciding the sentiment class of a sentence. Term Frequency–Inverse Document Frequency (TF-IDF) is another technique that is based on the BoW model [69]. It is a statistical method that assesses the importance of a word for a document in a set of documents. TF-IDF vectorization resembles the BoW structure, but it uses the TF-IDF value of words instead of their counts. It calculates the relevance of a word for a document considering the set of all documents. However, similar to its predecessors, context information is not retained as each word is handled individually. To overcome these problems, n-grams are utilized [42]. Briefly, the concatenation of n-neighbor words constitutes the n-grams of a document. However, n-grams can cause another problem, i.e., data sparsity. Sparsity increases as the *n* increases.

We elaborate on BoW, as it is used as a baseline text representation technique in our experiments. In BoW, each sentence is represented as a vector s = $<x_1, x_2, \ldots, x_n>$ where x_i denotes the number of occurrences of the i-th token and n is the total number of unique tokens in all sentences. The following example shows that syntactic and semantic relationships are lost in this method as it does not consider word orders. Assuming there are two sentences in the dataset: (i) 'I love tea, but I hate coffee', and (ii) 'I love coffee, but I hate tea'. The unique tokens (features) for this dataset will be $\{$'I', 'love', 'tea', 'but', 'hate', 'coffee'$\}$. Although the two sentences have different meanings, their vector representations with BoW will be identical: $<$2, 1, 1, 1, 1, 1$>$. Despite its weaknesses, BoW is one of the simple yet powerful text representation techniques.

In our studies, every unique word in the dataset represents a feature, and each sentence has a vector representation where the values are constructed by the number of occurrence information.

2.2.2 Pre-trained Language Models

As a remedy to the drawbacks described in the previous section, pre-trained models have become prevalent recently as they have shown great success in various NLP tasks, including sentiment analysis. Word embeddings have paved the way to demonstrate the effectiveness of pre-trained models [70]. A word embedding can simply be described as representing each word of a document with a vector of latent features of a specific length, where words with similar meanings have a similar representation. The real-valued feature vectors are calculated via training a neural network using a massive number of documents. Each dimension of this vector represents an underlying feature of the word. This training process utilizes word positions in the documents. As a result, it is possible to capture semantic relations with word embeddings [52]. A famous example that demonstrates the existence of semantic relations is as follows:

Having the feature vectors of the words *King*, *Queen*, *man*, and *woman*; if we subtract *man* from *King* and add *woman* to it, the result becomes the feature vector of the word *Queen*, demonstrating the semantic closeness of *King* and *Queen* in the vector space. This example shows that the model automatically learns the male/female relationship. In the last decade, Word2Vec [71] and GloVe [72] were the pioneers in terms of the pre-training methodology. One problem with these embeddings is that they may not consider the context [51]. For example, the words *beetle* as a car and *beetle* as an animal are represented with the same vector [51]. While some recent word embeddings, including BERT [54] and ELMo (Embeddings from Language Model) [73], operate in context-sensitive mode [74], pre-trained word embeddings with these models may not be the best fit for the studied domain or the language (dialect) [75].

GloVe was trained on the Wikipedia dataset by using word co-occurrence statistics [72]. It consists of vectors for four hundred thousand tokens with an option of varying dimensions (50, 100, 200, and 300)[2]. Each dimension of these vectors represents a feature. There exist different approaches to aggregate the word vectors to construct the vector representation of a sentence, such as concatenation or averaging. For example, the words ’I’, ’love’, and ’tea’ have the following vectors in 50-dimensional GloVe: <0.118, 0.152, ..., 0.921>, <-0.138, 1.140, ..., 0.289>, and <-0.449, -0.002, ..., -0.902>, respectively. Consequently, the vector representation of ’I love tea’ will be as follows when concatenated: <0.118, 0.152, ..., 0.921, -0.138, 1.140, ..., 0.289, -0.449, -0.002, ..., -0.902>.

Similarly, Word2Vec was trained on the Google News dataset by using continuous BoW architecture [71]. It consists of vectors for three million tokens[3].

Other than these pre-trained word embeddings, pre-trained language models are used to determine the initialization parameters for specific tasks or domains. The language models trained on vast corpora enable fine-tuning with a smaller amount of task-specific data as they have already acquired a significant amount of knowledge necessary for language processing. Therefore, these models help perform new tasks using the gained knowledge instead of learning from scratch. Moreover, pre-trained models come in handy, especially when there is a lack of labeled data. Many pre-

[2] `https://nlp.stanford.edu/projects/glove/`

[3] `https://code.google.com/archive/p/word2vec/`

trained language models, such as BERT [54], ELMo [73], or GPT [76], are shared publicly through artificial intelligence communities, such as Kaggle[4] and Hugging Face[5]. NLP studies have gained acceleration, especially with the effects of easily sharing code, models, and datasets through these artificial intelligence communities. Researchers worldwide can access an existing model, improve it in a certain way, and publish their findings over the same channel to allow others to use them directly or to improve them even further. For example, BERT pre-training has been successfully applied in various domains, some of which are provided below:

- FinBERT [77] (financial services)
- SciBERT [78] (biomedical and computer science literature)
- ClinicalBERT [79] (clinical notes)

Moreover, language models can be adapted to different languages. Farahani et al. [80] presented ParsBERT, a monolingual BERT for the Persian language. Similarly, Chouikhi et al. [81] proposed a BERT tokenizer specialized for Arabic texts.

In our study, we utilized pre-trained BERT models to take advantage of available language knowledge. There exist many pre-trained BERT models that are fine-tuned for the text classification task [82]. For example, the *distilbert-base-uncased-finetuned-sst-2-english* model is a well-known sentiment classification model that classifies sentences as positive or negative with a confidence score. The confidence score is a value between 0 and 1 for the predicted class.

2.2.3 Feature Selection

The classification task is one of the fundamental problems in knowledge discovery. The accuracy of classification highly depends on the quality of the data. Therefore, it is vital to preprocess the data to extract valuable information. Especially in real-world applications, the data amount is generally high, and there exist many redundant or irrelevant features that have no contribution to the classification task.

Feature selection is an important preprocessing step for classification. It aims to find the most informative features that can represent the data. Through feature selection,

[4] https://www.kaggle.com/
[5] https://huggingface.co/

the training time of the model is also reduced. Moreover, the learning performance of the model improves as unnecessary features will not clutter the model. However, the feature selection task can be challenging, as it is a combinatorial optimization problem.

Feature selection requires optimizing two objectives, minimizing the number of features and maximizing the classification performance. This optimization task can be formally defined as follows:

$$\begin{aligned}
&min\ obj_1 \\
&max\ obj_2 \\
&subject\ to \\
&\qquad obj_1 = |d| \\
&\qquad obj_2 = performance(d) \\
&where\ d \subseteq D
\end{aligned} \tag{2.1}$$

where D is the data with all features, and d is the selected feature subset of D. In this equation, obj_1 and obj_2 indicate the first and second objectives, respectively. Regarding these objectives, we aim to reduce the number of features, i.e., obj_1, while we try to improve the classification performance, i.e., obj_2. The classification performance could be measured with any performance metric, such as Accuracy, F1 score, or Area Under the Curve score. For instance, when the performance metric is accuracy, an ideal solution would have a 100% classification accuracy using only one feature according to the feature selection definition.

In a multiobjective optimization problem, there might be a solution set instead of only one solution. The reason is that one solution might be good at achieving one objective, while another solution is good at achieving another. To illustrate, in Figure 2.1, we provide sample solutions for a feature selection task in which the two objectives defined above are optimized. In this figure, the solutions in green fit on a Pareto curve. These solutions are called non-dominated solutions, as they are not dominated by any other solution in both objectives. On the other hand, the red-colored solutions are dominated in both objectives by at least one other solution. For example, solution

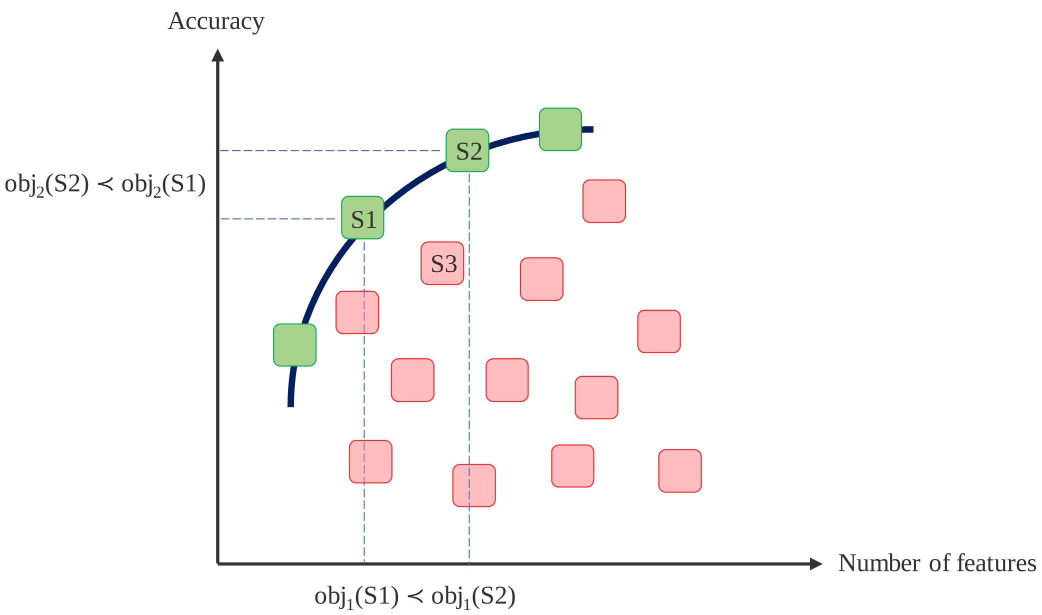


Figure 2.1: Sample solutions fitting to a Pareto curve for the two objectives of the multiobjective feature selection problem.

S1 is better than solution S3 in both objectives as it has fewer features and higher accuracy, as given by the inequalities below:

$$\begin{aligned} obj_1(S1) < obj_1(S3) \\ obj_2(S1) > obj_2(S3) \end{aligned} \tag{2.2}$$

As a result, S1 dominates S3, as represented below:

$$S1 \prec S3 \tag{2.3}$$

With a similar comparison, it can be seen that solution S1 cannot dominate solution S2. The number of features in S1 is less than the number of features in S2, but the accuracy of S2 is higher than the accuracy of S1. Hence, they are non-dominated solutions as they have better results in different objectives. As a result, these non-dominated solutions are presented as the final solution set for the problem.

There are three approaches for feature selection: filter-based, wrapper-based, and embedded [83]. Filter-based feature selection relies on statistical metrics (e.g. Chi-square, Information Gain, Gini index) that calculate the significance of features, and they are known for their fast executions [84]. Wrapper-based methods employ heuristic search algorithms (e.g. Genetic Algorithm). Such search algorithms tend to achieve better results than the filter-based approaches, yet, they require an excessive amount of time [85]. Finally, embedded methods perform feature selection while training the model, as they combine feature selection with the construction of the machine learning models.

CHAPTER 3

EXPERIMENTAL ENVIRONMENT

In this chapter, we describe the experimental environment by introducing the datasets, preprocessing and machine learning techniques, evaluation metric, and experimental settings.

3.1 Datasets

We evaluated the performance of our models on five datasets: three benchmark and two new datasets. The specifications of all datasets are provided in detail in the following subsections. We note that the language of the examined texts in all datasets is English.

3.1.1 Benchmark Datasets

3.1.1.1 Stanford Sentiment Treebank

Stanford Sentiment Treebank (SST) was presented in 2013 by Socher et al. [86], and since then, it has been one of the most commonly used datasets for sentiment analysis tasks in the literature [49, 51, 52]. In the dataset, there exist more than 10,000 movie reviews, and they are split into the train, test, and validation sets. The dataset contains sentiment scores between 0 and 1 for each review (sentence). Sample instances from the SST dataset are provided in Table 3.1. We converted the sentiment scores into classes using threshold values. We labeled the sentences as positive when the sentiment score is greater than or equal to 0.7, and we labeled them as negative

Table 3.1: Sample instances from the SST dataset.

Sentence	Train (1), test (2) or validation (3)	Sentiment score
"If you enjoy more thoughtful comedies with interesting conflicted characters; this one is for you."	3	0.91667
"So original in its base concept that you can not help but get caught up."	1	0.88889
"I have two words to say about Reign of Fire."	1	0.5
"Scene-by-scene, things happen, but you'd be hard-pressed to say what or why."	2	0.31944
"The most offensive thing about the movie is that Hollywood expects people to pay to see it."	1	0.18056
"Plodding, poorly written, murky and weakly acted, the picture feels as if everyone making it lost their movie mojo."	1	0

Table 3.2: Number of instances for each sentiment class in the SST dataset.

Sentiment	Number of instances	
	Train set	Test set
positive	2469	996
neutral	2526	892
negative	3122	1281

Table 3.3: Statistics of the sentences in the SST dataset.

Description	Before preprocessing	After preprocessing
total count of sentences	7868	7868
total count of unique tokens in all sentences	14,420	15,334
average number of tokens in all sentences	16.1	9.3
standard deviation of the number of tokens in all sentences	8.2	4.7
minimum number of tokens in all sentences	1	0
25% percentile (lower quartile) of the number of tokens in all sentences	10	6
50% percentile (median) of the number of tokens in all sentences	15	9
75% percentile (upper quartile) of the number of tokens in all sentences	22	12
maximum number of tokens in all sentences	50	28

Table 3.4: Sample instances from the MR dataset.

Sentence	Sentiment
"a comedy that swings and jostles to the rhythms of life ."	positive
"two badly interlocked stories drowned by all too clever complexity ."	negative

Table 3.5: Number of instances for each sentiment class in the MR dataset.

Sentiment	Number of instances
positive	5331
negative	5331

when the sentiment score is less than or equal to 0.4. Furthermore, in Table 3.2, we share the total number of instances for each sentiment in training and test sets separately. As seen from the table, we obtained 7868 positive and negative labeled sentences. Moreover, we report statistics of the sentences in the dataset in Table 3.3.

3.1.1.2 Movie Reviews

Movie Reviews (MR) is another well-known source for sentiment analysis tasks[87, 88]. It was presented by Pang and Lee [89]. The dataset consists of 10,662 sentences retrieved from movie reviews. Some samples are presented in Table 3.4. The sentences are labeled as positive and negative. In Table 3.5, we present the number of instances for each class. Finally, the statistics of these sentences are provided in Table 3.6.

3.1.1.3 Sentiment 140

Sentiment 140 (S140) is another well-known dataset for sentiment analysis studies [90, 91]. It contains 1,600,000 tweets retrieved from Twitter, along with their sentiment labels [92]. We present sample instances from this dataset in Table 3.7. The dataset consists of separate training and test instances. We present the total sentence

Table 3.6: Statistics of the sentences in the MR dataset.

Description	Before preprocessing	After preprocessing
total count of sentences	10,662	10,662
total count of unique tokens in all sentences	18,330	20,320
average number of tokens in all sentences	18.1	10.6
standard deviation of the number of tokens in all sentences	8.5	4.8
minimum number of tokens in all sentences	1	1
25% percentile (lower quartile) of the number of tokens in all sentences	12	7
50% percentile (median) of the number of tokens in all sentences	18	10
75% percentile (upper quartile) of the number of tokens in all sentences	24	14
maximum number of tokens in all sentences	50	39

Table 3.7: Sample instances from the S140 dataset.

Sentence	Train (1) or test (2)	Sentiment
"ow, i can't move my neck or my back hurts too much and i don't know why D:"	1	negative
"making the best cupcakes EVER.. and eating them as I go"	1	positive
"blah, blah, blah same old same old. No plans today, going back to sleep I guess."	2	negative
"Malcolm Gladwell might be my new man crush"	2	positive

Table 3.8: Number of instances for each sentiment class in the S140 dataset.

(a) All dataset.

Sentiment	Number of instances	
	Train set	Test set
positive	800,000	182
negative	800,000	177

(b) Used dataset.

Sentiment	Number of instances	
	Train set	Test set
positive	5000	182
negative	5000	177

Table 3.9: Statistics of the sentences in the S140 dataset.

Description	Before preprocessing	After preprocessing
total count of sentences	10,359	10,359
total count of unique tokens in all sentences	19,181	20,456
average number of tokens in all sentences	12.5	7.8
standard deviation of the number of tokens in all sentences	6.6	3.9
minimum number of tokens in all sentences	1	0
25% percentile (lower quartile) of the number of tokens in all sentences	7	5
50% percentile (median) of the number of tokens in all sentences	12	7
75% percentile (upper quartile) of the number of tokens in all sentences	18	11
maximum number of tokens in all sentences	30	25

count for each label in Table 3.8 with two subtables. The first subtable contains information about all instances of the dataset. Since the dataset is very large, we downsampled it when using it in the experiments. So, the second subtable contains information about the utilized dataset. Moreover, the statistics of these utilized sentences are given in Table 3.9.

3.1.2 New Datasets

We introduce two new datasets[1], details of which are provided in the following subsections.

3.1.2.1 International Monetary Fund Executive Board Meeting Minutes

The International Monetary Fund (IMF) is an international organization that is collectively managed by 190 countries[2]. Its main purposes are providing financial stability and cooperation, reducing unemployment and poverty, and facilitating trade between countries. IMF monitors the financial policies of the member countries and suggests modifications where necessary. Moreover, it provides loans to member countries to help them recover from financial problems.

[1] Datasets are available at MA-Computational Social Science Lab: https://ma-cssl.com/

[2] https://www.imf.org/en/home

Table 3.10: Sample instances from the IMF dataset.

Country	Year	Sentence	Sentiment	Author
Bulgaria	2004	"Mr. Mozhin and Mr. Lissovolik submitted the following statement:"	0	Mr Lissovolik;Mr Mozhin
Bulgaria	2004	"We thank the staff for a comprehensive and lucid paper."	0	Mr Lissovolik;Mr Mozhin
Bulgaria	2004	"Bulgaria is making important progress in consolidating the achievements of its economic reforms, which in recent periods have been duly rewarded by upgrades of credit rating agencies."	1	Mr Lissovolik;Mr Mozhin
Bulgaria	2004	"At the same time the staff report rightly notes that on some of the fronts vulnerabilities have increased, most notably this concerns the high level of the external gap."	-1	Mr Lissovolik;Mr Mozhin
Bulgaria	2004	"On the whole, however, we were pleased to learn from the statement by the staff representative on Bulgaria that the prior actions for the stand-by request have been met."	1	Mr Lissovolik;Mr Mozhin
El Salvador	1990	"Ms. Powell made the following statement:"	0	Ms Powell
El Salvador	1990	"The staff notes that in the last five years, economic growth in El Salvador averaged around 1.5 percent a year, and the external position deteriorated–a result of armed conflict, adverse external developments, and political uncertainty."	-1	Ms Powell
El Salvador	1990	"The new authorities in El Salvador should, therefore, be congratulated for the initiative they are taking to reduce imbalances and secure stronger economic growth in that country."	1	Ms Powell
El Salvador	1990	"By their recent actions, which included a tightening of monetary policy and the unification of the exchange rate, the authorities have shown a strong commitment to this process, and the program they have outlined for 1990 and 1991 deserves our support."	1	Ms Powell
El Salvador	1990	"It is clear that the ability of the authorities to successfully implement this program will depend critically on a resolution of the military conflict."	0	Ms Powell

Research studies have been leveraging IMF reports to analyze the effects of decisions made. Couharde et al. [93] argued that the Regional Economic Outlook reports published by IMF have a significant reflection on the stock market. Similarly, Breen et al. [94] revealed the incoherence in regime complexes by analyzing the sentiments of IMF and European Union surveillance documents.

We collected 611 files that contain IMF Executive Board meeting minutes between 1983 and 2015 for 127 countries. These files consist of 35,911 pages. The sentences in these files are labeled as positive (1), neutral (0), or negative (-1) regarding the performance of the borrowing country (the design and implementation of IMF programs). Sample annotations are provided in Table 3.10. Every document has been labeled by two different annotators. If two annotators agree on the label of the sentence, then the final label is directly set. Otherwise, if one of the annotator's label is neutral, the final label of the sentence is set as the other annotator's label. In case there is a conflict between the annotators, i.e., one label is negative, and the other one is positive, then the conflict is resolved when processing the documents using a weighted randomized selection with the reliability score of each annotator. In order to calculate the reliability scores, we initially asked the annotators to label the same document, which was labeled by two experts beforehand. We set every annotator's reliability score with respect to the accurate labeling percentage of this document. When a conflict occurs between two annotators, we resolve it by setting the final label using a roulette wheel selection in order to give a chance to each annotator's labeling. For example, if the reliability scores of the two annotators are 96% and 92%, then their labels will be selected with the probabilities of 96/188 and 92/188, respectively.

We present the total sentence count for each label in Table 3.11 with two subtables. The first subtable consists of information about the whole dataset. Around 50% of the sentences are labeled as neutral, 30% of them are labeled as positive, and the remaining 20% are labeled as negative. The second subtable presents the number of used instances in the experiments. Moreover, the statistics of these instances are given in Table 3.12.

We anticipate that modeling the sentiment information within this dataset is more

Table 3.11: Number of instances for each sentiment class in the IMF dataset.

(a) All dataset.

Sentiment	Number of instances
positive	86,934
neutral	144,625
negative	57,106

(b) Used dataset.

Sentiment	Number of instances
positive	6174
neutral	0
negative	3826

challenging than other datasets, such as Twitter or reviews data, for two reasons. First of all, many technical terms are used during these meetings, as the target audience is not the public but the representatives of other countries. Besides, speakers may aim to disguise their true feelings for bureaucratic purposes.

Initial Analysis

In order to gain a good understanding of the data, we held initial experiments using different feature extraction methods, machine learning techniques, and neural network language models [95].

Accuracy results achieved by three feature extraction methods, BoW, N-gram, and TF-IDF, along with two machine learning techniques Logistic Regression (LR) and Support Vector Machines (SVM), are given in Table 3.13. The first subtable contains results for the whole dataset. We encountered an out-of-memory problem as the number of features in the whole dataset is 61,953. Therefore, we removed the tokens that appear only once in the documents to shrink the number of features. We were able to achieve maximum accuracy of 86.1% with 17,187 features by the BoW method using LR as the classifier. Although we applied the same method to N-gram, we could not get a result as its number of features reached almost 200,000. The second subtable presents results for a portion of the dataset used in the experiments. The maximum accuracy, 81.3%, is achieved by the BoW method using LR as the classifier. The BoW is followed by TF-IDF and N-gram, in respective order, in terms of accuracy. N-gram extracts more features than BoW and TF-IDF, which affects the classification performance adversely and increases the execution time. When we compare

Table 3.12: Statistics of the sentences in the IMF dataset.

(a) All dataset.

Description	Before preprocessing	After preprocessing
total count of sentences	288,665	288,665
total count of unique tokens in all sentences	43,415	61,953
average number of tokens in all sentences	25.4	14.5
standard deviation of the number of tokens in all sentences	12.3	7.1
minimum number of tokens in all sentences	1	0
25% percentile (lower quartile) of the number of tokens in all sentences	17	10
50% percentile (median) of the number of tokens in all sentences	24	13
75% percentile (upper quartile) of the number of tokens in all sentences	32	18
maximum number of tokens in all sentences	259	171

(b) Used dataset.

Description	Before preprocessing	After preprocessing
total count of sentences	10,000	10,000
total count of unique tokens in all sentences	8758	9640
average number of tokens in all sentences	24.9	14.4
standard deviation of the number of tokens in all sentences	11.3	6.6
minimum number of tokens in all sentences	1	0
25% percentile (lower quartile) of the number of tokens in all sentences	17	10
50% percentile (median) of the number of tokens in all sentences	23	14
75% percentile (upper quartile) of the number of tokens in all sentences	31	18
maximum number of tokens in all sentences	164	94

machine learning techniques, LR stands out with the achieved maximum accuracy, and it achieves better accuracy values in two out of three methods.

In Table 3.14, we present the accuracy results achieved by two pre-trained BERT models, namely RoBERTa-large (*siebert/sentiment-roberta-large-english*) and FinBERT (*ProsusAI/finbert*). Moreover, we fine-tuned the RoBERTa-large model on our data. We share the prediction results in the last row. Similar to the previous table, we present the results in two subtables for the whole (a) and used (b) datasets. RoBERTa-large achieves the maximum accuracy of 82.4% among pre-trained mod-

Table 3.13: Accuracy results of all feature extraction methods on the IMF dataset.

(a) All dataset.

Method	Number of features	Accuracy	
		LR	SVM
BoW	17,187	0.8612	0.8609
N-gram	194,859	out of memory	
TF-IDF	17,187	0.6035	0.6035

(b) Used dataset.

Method	Number of features	Accuracy	
		LR	SVM
BoW	9640	0.8128	0.8116
N-gram	86,286	0.7176	0.7212
TF-IDF	9640	0.8090	0.8089

Table 3.14: Accuracy results of BERT models on the IMF dataset.

(a) All dataset.

Model	Accuracy	Execution time (min.)
RoBERTa-large	0.8243	253
FinBERT	0.6082	72
Fine-tuned BERT	0.9672	5483

(b) Used dataset.

Model	Accuracy	Execution time (min.)
RoBERTa-large	0.8216	17
FinBERT	0.6085	5
Fine-tuned BERT	0.9700	217

els for the original data. When we fine-tune the BERT model with randomly selected 80% of our data, we achieve 96.7% accuracy in the remaining test data. We observe similar results for the used dataset. RoBERTa-large achieves the maximum accuracy of 82.2% among pre-trained models. Fine-tuned BERT model outperforms others by obtaining 97.0% accuracy. For both versions of the dataset, fine-tuning increases the performance significantly; however, the execution time of it is much more than applying the pre-trained models. The fine-tuned model is available on Hugging Face in binary[3] and multi-class[4] forms.

Experiment results show that highly accurate results are only achieved when the training data includes data annotated specifically for the domain in question in addition to the data of more generic pre-trained models. This is most likely due to the usage

[3] https://huggingface.co/faycadnz/IMFBERT_binary

[4] https://huggingface.co/faycadnz/IMFBERT_multi

Table 3.15: Sample instances from the WHO dataset.

Sentence	Sentiment
"This marked one of the greatest public health achievements of all time."	positive
"That is when you can clearly see what works, what doesn't and what you need to improve."	neutral
"However, the COVID-19 pandemic hurt momentum as polio and immunization efforts were suspended."	negative

Table 3.16: Number of instances for each sentiment class in the WHO dataset.

Sentiment	Number of instances
positive	5355
neutral	2718
negative	2002

of technical terminology specific to the domain (IMF in this case) and the different usage of language to maintain political correctness.

3.1.2.2 World Health Organization Director-General's Speeches

This dataset consists of the speeches of the World Health Organization (WHO)[5] Director-General during the emergence of the COVID pandemic (between February 2020 and November 2020) [96]. After collecting the data, we labeled the sentences in the same manner as the IMF dataset. Sample instances from the WHO dataset are presented in Table 3.15. The dataset contains a total of 10,075 sentences. In Table 3.16, we share the total number of instances for each sentiment category. Moreover, we report statistics of the sentences in the dataset in Table 3.17.

Initial Analysis

In order to gain a good understanding of the data, we held the same initial experiments as the IMF dataset.

[5] `https://www.who.int/`

Table 3.17: Statistics of the sentences in the WHO dataset.

Description	Before preprocessing	After preprocessing
total count of sentences	7357	7357
total count of unique tokens in all sentences	6801	7028
average number of tokens in all sentences	18.7	10.1
standard deviation of the number of tokens in all sentences	9.3	5.4
minimum number of tokens in all sentences	1	0
25% percentile (lower quartile) of the number of tokens in all sentences	12	6
50% percentile (median) of the number of tokens in all sentences	18	10
75% percentile (upper quartile) of the number of tokens in all sentences	24	13
maximum number of tokens in all sentences	70	57

Table 3.18: Accuracy results of all feature extraction methods on the WHO dataset.

Method	Number of features	Accuracy	
		LR	SVM
BoW	7028	0.8603	0.8612
N-gram	42,069	0.8073	0.8105
TF-IDF	7028	0.8441	0.8448

In Table 3.18, we present the number of features and accuracy values for various feature extraction methods applied to the WHO dataset. We observe a maximum accuracy of 86.1% with 7028 features by the BoW method using SVM as the classifier. Other feature extraction methods achieve similar results with BoW. However, N-gram requires more features (42,069), whereas TF-IDF uses the same amount of features as BoW.

In Table 3.19, we present the performance of the pre-trained language models and the fine-tuned model as we did for the IMF dataset. RoBERTa-large achieves the maximum accuracy of 82.9% among pre-trained models for the original data. It is clear that FinBERT is not a good fit for the WHO dataset as it underperforms. When

Table 3.19: Accuracy results of BERT models on the WHO dataset.

Model	Accuracy	Execution time (min.)
RoBERTa-large	0.8293	12
FinBERT	0.3476	3
Fine-tuned BERT	0.9769	137

we fine-tune the BERT model with randomly selected 80% of our data, we achieve 97.7% accuracy in the remaining test data. As expected, fine-tuning increases the performance significantly; however, the execution time of it is much more than applying the pre-trained models.

3.2 Preprocessing

Preprocessing is a crucial phase that affects the performance of classifiers [97]. With this step, the redundant data in the raw dataset are filtered out, as they do not have a meaningful contribution to the classification task. Moreover, reducing the dimensionality of the data speeds up the training process. We utilized the NLTK[6] library for preprocessing operations. In our proposed models, the preprocessing phase is three-fold:

Conversion to lowercase

In this step, all the words in all sentences are converted to lowercase. Without this operation, the model treats a word with a capital letter differently from the same word without any capital letters, which could increase data sparsity and decrease the prediction accuracy of the model.

[6] https://www.nltk.org

Punctuation removal

In this step, we removed all non-alphanumeric characters and punctuation marks except parentheses, commas, exclamation marks, question marks, and apostrophes. Similar to the previous step, the aim is to lower data sparsity, as the model cannot discriminate between punctuation and other characters.

Stop words removal

Stop words are the words that occur in texts with high frequencies but do not add a specific meaning to the text, such as *a*, *an*, *the*, *of*, etc. Therefore, in this step, stop words are removed so that only significant words are left for the training part.

The effects of preprocessing can be seen in Tables 3.3, 3.6, 3.9, 3.12, and 3.17, where we share the descriptive statistics of all datasets before and after preprocessing. The tables show that the total count of unique tokens increases after the preprocessing steps are applied. With the punctuation removal phase, two-word tokens separate, which increases the token count, e.g., *machine-learning* becomes *machine* and *learning*. However, the average number of tokens in sentences decreases after the preprocessing. The main reason for that is the removal of the stop words. Converting all words to lowercase may also have an effect as tokenization is case-sensitive, e.g., *Machine* and *machine* are two different tokens before preprocessing.

3.3 Machine Learning Techniques

There exist many effective machine learning techniques for classification tasks. We evaluated the performance of our models using two machine learning techniques which are briefly described below. We utilized the scikit-learn[7] implementation of these techniques.

[7] https://www.scikit-learn.org

3.3.1 Logistic Regression

Logistic Regression (LR) builds a probabilistic classification model [98]. It is known as an easy-to-use and efficient classifier [98]. It uses the Sigmoid function to generate a probability value for each instance belonging to a class as follows:

$$P(Y = 1 \mid X, \theta) = \frac{1}{1 + e^{-\theta X}} \tag{3.1}$$

where X is the input data, θ is the coefficient values for the input, and Y is the probability of an item belonging to class 1.

3.3.2 Support Vector Machines

Support Vector Machines (SVM) builds a linear classification model [99]. It generates a hyperplane between two classes by positioning it as far as possible from the closest data points of each class, called support vectors, with regard to the equation below:

$$\begin{aligned} & minimize\ ||w||\ in\ (w, b) \\ & subject\ to \\ & \qquad y_i(w^T x_i + b) \geq 1 \quad for\ i = 1...N \end{aligned} \tag{3.2}$$

where w, b, x, and y are the weight, bias, input, and output vectors, respectively, and N is the number of instances.

3.4 Evaluation

We used accuracy as the performance metric to evaluate the performance of the models. Accuracy is the ratio of correctly predicted instances over all instances. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.3}$$

where TP, TN, FP, and FN indicate the number of true positive, true negative, false positive, and false negative instances, respectively.

3.5 Experimental Settings

In this section, we share the common parameter settings of all used algorithms and techniques. The model-specific settings are provided in the model's chapter.

Implementation

We carried out the experiments on a computer with Intel Core i7-9700K Eight-Core Processor with a 3.6 GHz clock rate and 16 GB of main memory. For the neural network models, we utilized the high-performance computers of the Turkish National e-Science e-Infrastructure (TRUBA)[8] with a 28-core of 2.70 GHz clock rate and 192 GB of memory.

We used Python for implementation.

We used bigrams when the text representation technique was n-grams. We set the *n* as 2 to capture the two-word sequence of words.

Datasets

We focused on sentence-level polarity classification.

For IMF and S140 datasets, we randomly selected 10,000 sentences to be used in the experiments as they had a huge amount of instances.

We filtered out the neutral-labeled instances from all datasets as we applied binary classification.

For the SST and S140 datasets, the instances were split into the train and test sets in the original data. Therefore, we utilized the predefined test data in the testing part of our study. For the other datasets, we performed *k*-fold cross-validation to prevent bias, as there were no predefined splits.

[8] `https://docs.truba.gov.tr/`

Machine learning techniques

For LR, we set the *solver* parameter as *lbfgs* and the *multi_class* parameter as *ovr* since we perform binary classification. Other than that, we set the maximum number of iterations used to converge, *max_iter*, as *1000*.

For SVM, the regularization parameter, i.e., *C*, is an important parameter for performance. When it increases, training error decreases, whereas computation time massively increases as it tries to find a smaller-margin hyperplane that separates the classes. Therefore, we set *C* as *0.1* in our implementation. We kept it small to avoid redundant computation to find smaller-margin hyperplanes.

We retrieved the pre-trained neural network language models from Hugging Face. We used *Trainer API* to fine-tune BERT. We set the *learning rate* parameter as *1e-5*, the *number of epochs* parameter as *3*, and the *batch size* parameter as *8*.

CHAPTER 4

FEATURE SELECTION FOR SENTIMENT ANALYSIS

The big data era, with a high volume of data generated by a variety of sources, has provided enhanced opportunities for utilizing sentiment analysis in various domains. In order to take the best advantage of the high volume of data for accurate sentiment analysis, it is essential to clean the data before the analysis, as irrelevant or redundant data will hinder the extraction of valuable information. In this chapter, we present a hybrid feature selection algorithm to improve the performance of sentiment analysis tasks. Our proposed approach builds a binary classification model based on two feature selection techniques: an entropy-based metric and an evolutionary algorithm. The proposed feature selection model is shown to achieve significant performance improvements in all datasets, increasing classification accuracy for all utilized machine learning and text representation technique combinations. Moreover, it achieves over 60% reduction in feature size for all datasets, which provides efficiency in computation time and space.

4.1 Related Work

Although sentiment analysis has been extensively studied in the literature, new studies continue to emerge as available data continually grow and become more complex. It is crucial to select the optimal feature subset for sentiment analysis to achieve high performance [100]. Therefore, feature selection is an indispensable preprocessing step, alleviating the burden caused by the high-dimensional data.

Feature selection has been widely used for sentiment analysis in various domains and has proven to enhance the performance of sentiment classification [101]. Recently, Madasu and Elango [102] presented a detailed evaluation of different feature selection methods for sentiment analysis. They reported that feature selection methods, especially the ones that utilize ensemble techniques, obtain superior results by boosting the sentiment analysis performance. Ahmad et al. [103] reviewed feature selection methods used for sentiment analysis. They identified and presented the advantages and disadvantages of these methods. The authors suggested that metaheuristic algorithms perform well when selecting the optimal features for sentiment analysis. Shang et al. [104] presented a binary-based Particle Swarm Optimization (PSO) for feature selection in the sentiment analysis domain. Their algorithm was built to overcome the shortcomings of the traditional PSO algorithm, such as the update formula of velocity. Similarly, Kumar et al. [105] proposed a Firefly Algorithm for optimizing the feature sets to be used in sentiment analysis. They applied their algorithm to Hindi and English texts using SVM as the classifier. Gokalp et al. [106] proposed another wrapper-based feature selection method for sentiment analysis. The proposed model is based on a Greedy Algorithm that utilizes six different filter-based metrics, including Chi-square and ReliefF, in the construction of the model. Experiments on many public datasets showed that the model is more effective than conventional filter-based feature selection methods.

Recent literature on feature selection has mainly focused on wrapper-based feature selection methods as they generally perform better than filter-based methods [107]. However, these methods are expensive in terms of computation time and space, as wrapper-based feature selection is an NP-hard problem [108]. Metaheuristic algorithms are known to be very efficient for NP-hard problems [109]. They have been utilized by many researchers for feature selection in recent years. Al-Tashi et al. [110] presented a detailed review of multiobjective feature selection techniques and challenges. Kiziloz et al. [111] proposed three variants of multiobjective Teaching Learning Based Optimization algorithm for the feature selection task. Similarly, Sihwail et al. [112] proposed an improved version of Harris Hawk Optimization for the feature selection task. They presented three new search strategies to enhance the exploration capability of the hawks. Hu et al. [113] proposed a fuzzy cost-based Particle Swarm Optimization algorithm for multiobjective feature selection. Similarly, Zhang et al. [114] presented novel operators for the Artificial Bee Colony algorithm to tackle cost-sensitive multiobjective feature selection problems. Zhang et al. [115] employed differential evolution to improve the search operation of multiobjective feature selection tasks.

There exist studies that combine multiple feature selection methods to enhance the efficiency of the sentiment analysis task. Rasool et al. [41] proposed a hybrid feature selection method for sentiment classification. They selected promising features using different wrapper approaches and transferred them to the population of their Genetic Algorithm. Similarly, Ansari et al. [116] proposed another hybrid method for sentiment classification. They first applied two filter-based methods and extracted the most valuable features obtained by both methods. Then, they fed these features to two wrapper-based methods separately, namely, PSO and Recursive Feature Elimination, and reported that feature selection improves the classification performance tremendously. Pandey et al. [117] introduced another metaheuristic method, namely Cuckoo Search Algorithm, for sentiment analysis tasks. They utilized K-means to enhance the initialization process of their algorithm for faster convergence and better solution sets. Recently, Tubishat et al. [118] proposed an improved version of the Whale Optimization Algorithm (WOA) for sentiment analysis in Arabic texts. They combined Differential Evolution with Elite Opposition-Based Learning to boost the performance of WOA. Moreover, they utilized a filter-based feature selection method to feed valuable features to their algorithm. Hassonah et al. [119] introduced a hybrid feature selection method for sentiment analysis. Their method consists of a filter- and wrapper-based approach. They analyzed the extracted features to find out which type of features (subjective, objective, or emoticons) are more valuable in the sentiment analysis task.

As a result, previous studies mainly focused on filter [120] and wrapper [121] based feature selection methods. Although there exist feature selection methods that combine filter- and wrapper-based approaches for sentiment analysis [118, 119], all of them approach the problem from a single objective perspective. To the best of our knowledge, applying a multiobjective hybridized feature selection method to the sentiment analysis task has not been investigated yet.

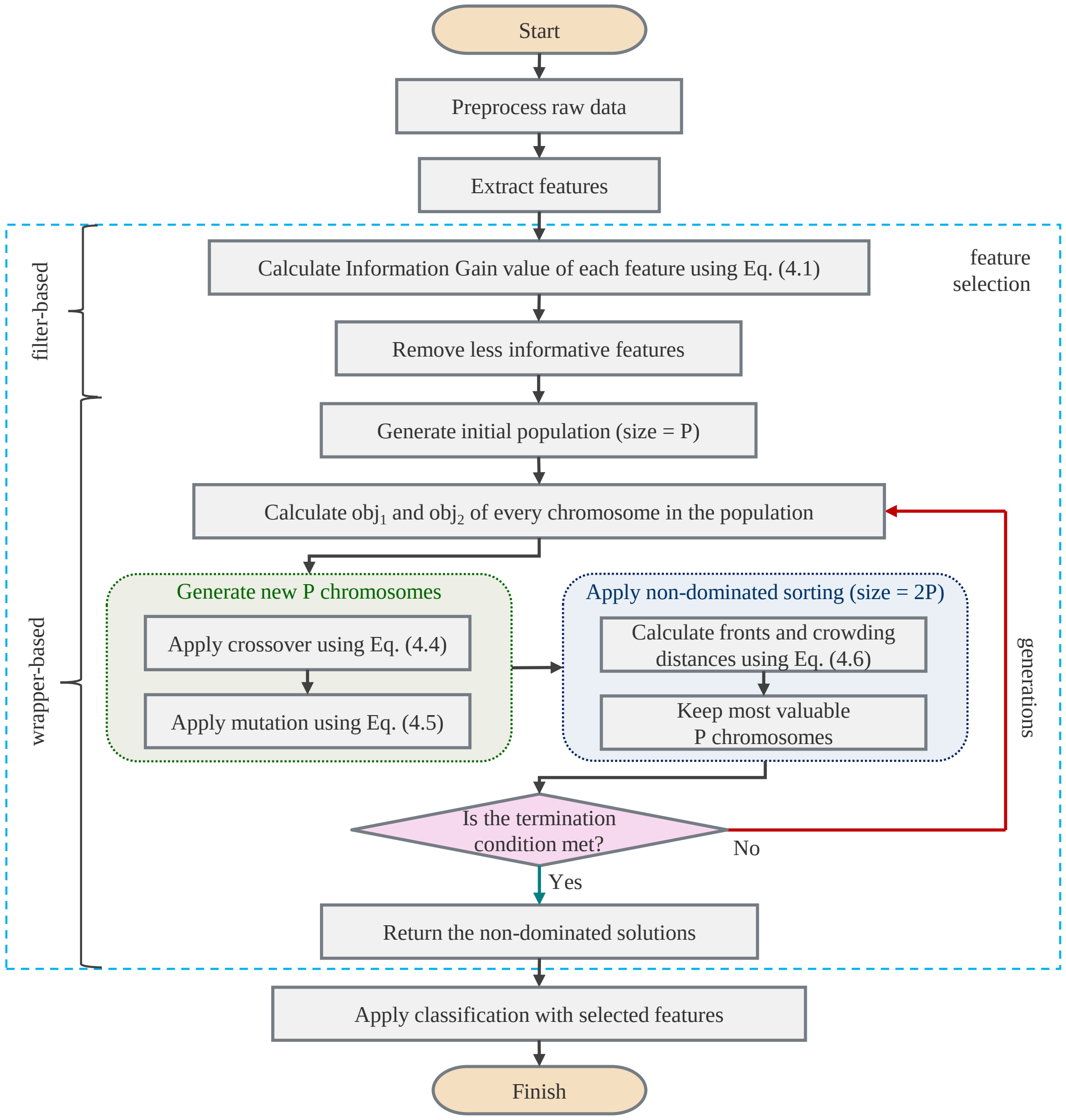

Start
Preprocess raw data
Extract features
filter-based
Calculate Information Gain value of each feature using Eq. (4.1)
feature selection
Remove less informative features
Generate initial population (size = P)
wrapper-based
Calculate obj_1 and obj_2 of every chromosome in the population
Generate new P chromosomes
Apply crossover using Eq. (4.4)
Apply mutation using Eq. (4.5)
Apply non-dominated sorting (size = 2P)
Calculate fronts and crowding distances using Eq. (4.6)
Keep most valuable P chromosomes
generations
Is the termination condition met?
No
Yes
Return the non-dominated solutions
Apply classification with selected features
Finish

Figure 4.1: Proposed feature selection model.

4.2 Model

The flowchart of the proposed feature selection model is depicted in Figure 4.1. The algorithm begins by applying preprocessing to the raw data (see Algorithm 1). After preprocessing is completed, features are extracted. There exist many feature extraction techniques to translate free-form textual data into a standard representation that machine learning techniques can interpret. In order to show that our model is viable regardless of the feature extraction technique, we tested it with different techniques separately. In this work, we utilized two feature representation techniques, *BoW* and

Algorithm 1: Algorithm of the data cleaning process.

```
Input: the sentences as separate instances: instances
Output: preprocessed sentences: instances
Function CleanData(instances):
    instances $\leftarrow$ RemovePunctuation(instances);
    instances $\leftarrow$ RemoveExtraWhitespaces(instances);
    instances $\leftarrow$ ConvertToLowercase(instances);
    instances $\leftarrow$ RemoveStopWords(instances);
    return instances
```

GloVe, which have different strengths and weaknesses, as mentioned in Chapter 2. As soon as the features are ready, the feature selection process begins. Feature selection in our model comprises two parts: filter- and wrapper-based. With this process, the most promising features for the sentiment classification task are extracted. The steps of our feature selection algorithm are explained in detail in the subsections below.

Filter-based feature selection

Fundamentally, the value of features can be measured with different filter-based methods. In the filter-based feature selection part of our model, we opted for Information Gain [122], a widely recognized metric with a straightforward implementation. It measures the information amount that a single feature carries in a set of features. Information Gain of a feature F is calculated with the following formula:

$$IG(D,F) = Entropy(D) - \sum_{u \in U} \frac{|D_u|}{|D|} Entropy(D_u) \tag{4.1}$$

where D is the data with all features and instances, F is the particular feature, U is the set of all the unique values for the related feature, and D_u is a subset of D, having the instances in which the value of F is u. $|D|$ and $|D_u|$ are the number of instances in D and D_u, respectively. The entropy of a subset S of the data is calculated as follows:

$$Entropy(S) = -\sum_{c \in C} p_c \log_2 p_c \tag{4.2}$$

where C is the set of all classes in the dataset and p_c is the ratio of the number of instances in the c-th class over the number of all instances in S.

In the literature, it is common to filter out the words that occur only once as they do not provide any predictive power [123]. By building on this idea, we filter out the words whose Information Gain value is below a certain threshold. However, it is not easy to choose a generic threshold value that would work well for all datasets. For this reason, we leverage information conveyed by the dataset itself to determine the threshold value. Consequently, in our model, we first calculate the Information Gain value of each feature in the dataset. Then, we compute the median value and set it as the threshold. Finally, we filter out the features whose values are less than the threshold as their predictive power is low. We call this procedure Information Gain Filtering (IGF) (see Algorithm 2). Choosing a smaller threshold value (e.g. first quartile value) would lead to the elimination of discriminative features for sentiment analysis. On the other hand, selecting this value larger (e.g. third quartile value) would prevent most features with low predictive power from being filtered out, which

Algorithm 2: Algorithm of the information gain filtering.

```
Input: information gain values of the features: ig_values,
threshold value to filter features: threshold
Output: indexes of selected features: selected_features

Function InformationGainFiltering(ig_values, threshold):
    $D \leftarrow length(ig\_values)$;
    $selected\_features \leftarrow \{\}$;
    for i=1,...,$D$ do
        if $ig\_values[i] \geq threshold$ then
            $selected\_features \leftarrow selected\_features \cup i$;
    return $selected\_features$
```

would worsen the learning performance.

Wrapper-based feature selection

In the wrapper-based feature selection part of our model, we apply the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [124]. NSGA-II is a well-known and efficient multiobjective optimization algorithm. With regard to the evolutionary nature of this algorithm, every possible solution is represented with a chromosome/individual, I, which equals to $[f_1, f_2, ..., f_N]$ where N is the total number of features in the dataset and f_i is the i-th feature in the dataset. A sample chromosome is also depicted in Figure 4.2. Each chromosome's length is the total number of features in the dataset. The value of each segment can be either 1 or 0, indicating that a feature is selected or not, respectively, as given below.

$$f_i = \{0, 1\} \qquad \text{for } i = 1...N \tag{4.3}$$

In the figure, features two, three, five, and eight are selected. Accordingly, the first objective (number of features) for this chromosome becomes four. In order to calculate the second objective (accuracy), the remaining features (one, four, six, and seven) are filtered out, and only the selected features are used to train a classifier.

The NSGA-II algorithm in our study executes as follows. First, an initial population that consists of randomly generated chromosomes is generated. Then, the values of both objectives are calculated for every individual in the population. With the determination of the population, the first generation begins. Similar to a standard genetic algorithm, crossover and mutation operators are applied to randomly selected

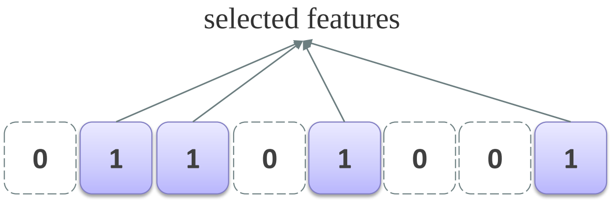


Figure 4.2: Sample chromosome in the proposed feature selection model.

individuals (parents) to create new individuals (children) as many as the population size. With crossover and mutation operators, we aim to increase the diversity in the population.

We utilized the half-uniform crossover operator in our study. Let C_1 and C_2 be two chromosomes in the population. Two new chromosomes, C_3 and C_4, are generated using the crossover operation between C_1 and C_2, respectively. The equation below depicts the generation of C_3:

$$C_{3i} = \begin{cases} C_{1i}, & \text{if } C_{1i} = C_{2i} \\ rand(0,1), & \text{otherwise} \end{cases} \qquad \forall i \in C_1 \tag{4.4}$$

where C_3 is the new chromosome and C_{1i}, C_{2i}, and C_{3i} are the i-th features in the chromosomes C_1, C_2, and C_3, respectively. C_4 is generated over C_2 in a similar fashion.

For mutating the newly generated chromosomes, we utilize the bit-flip mutation operator. The bit-flip mutation alters the chromosome as given in the equation below:

$$C_i' = \{1 - C_i : P(i) \geq MP\} \qquad \forall i \in C \tag{4.5}$$

where C' is the mutated chromosome, C_i' and C_i are the i-th features in the chromosomes C' and C, $P(i)$ is the randomly generated probability that the feature i is mutated, and MP is the predefined mutation probability which is shared in Section 4.3.1.

After crossover and mutation operations are applied in the population, all new individuals are evaluated in terms of both objectives. Particularly, NSGA-II is an elitist algorithm. Therefore, the new individuals do not necessarily replace the existing individuals, but rather all individuals are combined in a pool, doubling the population size. To continue its execution, NSGA-II selects the better half of the pool as the next generation. However, due to having two objective values, selecting the better half is not a straightforward process. For this purpose, we use the non-dominated sorting algorithm, a methodology to compare the individuals in a multiobjective environment.

Algorithm 3: Algorithm of the non-dominated sorting.

```
Input: population: P
Output: fronts: F

Function NonDominatedSort($P$):
    $i$ = 1;
    while $P \neq \varnothing$ do
        $F_i$ = $\varnothing$;
        foreach $p \in P$ do
            $n$ = 0;
            foreach $q \in P$ do
                if $q \prec p$ then
                    $n$ = $n$ + 1;
            if $n$ = 0 then
                $F_i$ = $F_i \cup$ {p};
        $P$ = $P \setminus F_i$;
        $i$ = $i$ + 1;
    return $F$; // $F$ consisting of all fronts {$F_1$, $F_2$, ...}
```

The non-dominated sorting algorithm (see Algorithm 3) divides the individuals into multiple fronts, as many fronts as required according to the dominance relationship. All the individuals that are not dominated by any other individual constitute the first front. Similarly, all the individuals that are dominated only by the individuals in the first front but not dominated by any other individuals constitute the second front. This operation is repeated until all the individuals are assigned into a front. In comparison, any individual assigned to a front with a smaller front number is better than any individual that is assigned to a front with a larger front number.

Crowding distance is used to compare the individuals within the same front. The crowding distance values of the individuals are determined considering their neighbors. The half perimeter of the rectangle including the nearest left and right neighbor individuals in the same front denotes the crowding distance of the related individual. The crowding distance value of an individual (solution), S, is calculated as follows:

$$CD(S) = \sum_{o \in O} \frac{|S_{o+1} - S_{o-1}|}{|f_o^{max} - f_o^{min}|} \tag{4.6}$$

where O is the set of all objectives, S_{o+1} and S_{o-1} are the o-th objective values of the immediate neighbors of S, and f_o^{max} and f_o^{min} are the maximum and minimum values obtained for the o-th objective. The two extreme individuals, one individual having the maximum accuracy value and the one individual having the minimum number of features, are provided with the maximum crowding distance values for the specific front. Once all the individuals are assigned a crowding distance value, the individual having a higher crowding distance is considered better. Application of the non-dominated sorting algorithm for the determination of the better half as the next population concludes the generation. The algorithm iterates for a predetermined number of generations and finally reports the non-dominated solutions of the final population as the result.

For clarity, we also provide the algorithm of our proposed model in Algorithm 4. As can be seen from the algorithm, the number of generations and population size are two main components contributing to the time complexity. The dataset size and complexity of the selected machine learning technique are other factors in the equation.

Our model naturally supports multi-class datasets. The only difference between binary and multi-class classification would be the selected machine learning technique used for the fitness value calculation.

4.3 Experiments

4.3.1 Settings

Deniz et al. [125] report that the NSGA-II algorithm achieves better results as the population size and the number of generations grow larger. Furthermore, they suggest that an increase in population size negatively affects the computation time more than an increase in the number of generations. Therefore, considering the sparsity of NLP datasets, in this study, we selected the *population size* as *100* and the *number of generations* as *200*. As the NSGA-II algorithm is elitist in its nature, it keeps a copy

Algorithm 4: Algorithm of the proposed feature selection model.

```
instances: input data
FE: feature extraction technique
ML: machine learning technique

// apply preprocessing
instances ← CleanData(instances); // Alg. 1

// extract features
features ← ExtractFeatures(instances, FE);

// apply filter-based feature selection
ig_values ← CalculateInformationGain(features, labels); // Eq. 4.1
threshold ← Median(ig_values);
feature_indexes ← InformationGainFiltering(ig_values, threshold); // Alg. 2

// apply wrapper-based feature selection
population ← GeneratePopulation(feature_indexes);
population ← CalculateFitnessValues(population, ML);

for (g ← 1 to number_of_generations) do
    for (p ← 1 to population_size) do
        $parent_1$, $parent_2$ ← SelectParents(population);
        child ← Crossover($parent_1$, $parent_2$); // Eq. 4.4
        child ← Mutation(child); // Eq. 4.5
        child ← CalculateFitnessValues(child, ML);
        population ← population $\cup$ child;
    // population size is doubled, keep better half
    fronts ← NonDominatedSort(population); // Alg. 3
    fronts ← CalculateCrowdingDistance(fronts); // Eq. 4.6
    population ← KeepBetterHalf(fronts);

print ($fronts_1$); // most valuable feature subsets
```

of the parents in the pool of individuals for the next generation. Therefore, we set the *crossover ratio* as *100%* to increase the diversity inside the population. Moreover, we set the *mutation ratio* as *2%* to increase the exploration space of the algorithm.

For IGF, we set the *threshold* value as the *median* of information gain values of the features. All features having an information gain value less than the median were filtered out, as they have less predictive power.

We employed *50-dimensional* GloVe vectors. When using GloVe as the feature extraction technique, we represented each sentence with the same vector size. Therefore, the sentences having fewer tokens than the threshold value were padded with empty vectors, and the sentences having more tokens were cut off from the threshold value. We set the threshold, i.e., the *maximum token count for each sentence*, as the *upper quartile value* of the number of tokens in all sentences.

For the IMF, MR, and WHO datasets, we applied a 5-fold cross-validation technique in our experiments as there was no specification for the train and test sets in the original data. For the other ones, we used the train and test sets provided by the original data in our experiments.

4.3.2 Results and Discussion

Table 4.1 presents the accuracy and number of features achieved by various algorithms combined with feature extraction and machine learning techniques in all datasets. Baseline results (preprocessed data) are given in the first row. In the second row, the results when only IGF is applied (preprocessed data + IGF) are shared. In the next row, the results when only NSGA-II is applied (preprocessed data + NSGA-II) are given. The results for the combined model (preprocessed data + IGF + NSGA-II) are presented in the last row of the table. It can be clearly seen that the proposed model achieves a significant increase in accuracy with much fewer features as compared to the baseline.

When we compare feature extraction techniques, BoW achieves higher accuracy values than GloVe. In terms of decreasing the number of features, both techniques manage to achieve a reduction of around 70%. We note that the results of GloVe might

Table 4.1: Experiment results of the proposed feature selection model with other sub-methods in terms of accuracy and number of features for all datasets.

(a) The SST dataset.

Model	BoW				GloVe			
	LR		SVM		LR		SVM	
	# of features	Accuracy	# of features	Accuracy	# of features	Accuracy	# of features	Accuracy
baseline	15334	0.8100	15334	0.8120	600	0.7418	600	0.7463
IGF	7669	0.8474	7669	0.8487	300	0.7302	300	0.7296
NSGA-II	7013	0.8455	7018	0.8603	177	0.7791	168	0.7830
IGF + NSGA-II	3344	0.8686	3314	0.8796	96	0.7740	94	0.7830

(b) The MR dataset.

Model	BoW				GloVe			
	LR		SVM		LR		SVM	
	# of features	Accuracy	# of features	Accuracy	# of features	Accuracy	# of features	Accuracy
baseline	20320	0.7563	20320	0.7534	700	0.6781	700	0.6796
IGF	15458	0.8017	15458	0.8034	350	0.6625	350	0.6632
NSGA-II	9154	0.7641	9253	0.7624	240	0.6956	174	0.6981
IGF + NSGA-II	7358	0.7989	7132	0.7955	96	0.6750	84	0.6750

(c) The S140 dataset.

Model	BoW				GloVe			
	LR		SVM		LR		SVM	
	# of features	Accuracy	# of features	Accuracy	# of features	Accuracy	# of features	Accuracy
baseline	20456	0.7716	20456	0.7744	550	0.7131	550	0.7159
IGF	17769	0.7939	17769	0.7967	275	0.7047	275	0.7019
NSGA-II	9258	0.8969	9356	0.9164	189	0.8662	159	0.8691
IGF + NSGA-II	8191	0.9192	7648	0.9415	85	0.8495	85	0.8523

(d) The IMF dataset.

Model	BoW				GloVe			
	LR		SVM		LR		SVM	
	# of features	Accuracy	# of features	Accuracy	# of features	Accuracy	# of features	Accuracy
baseline	9640	0.8128	9640	0.8116	900	0.7227	900	0.7233
IGF	4867	0.8287	4867	0.8303	450	0.7309	450	0.7291
NSGA-II	4347	0.8279	4420	0.8218	262	0.7512	297	0.7541
IGF + NSGA-II	2043	0.8337	2151	0.8368	155	0.7472	162	0.7476

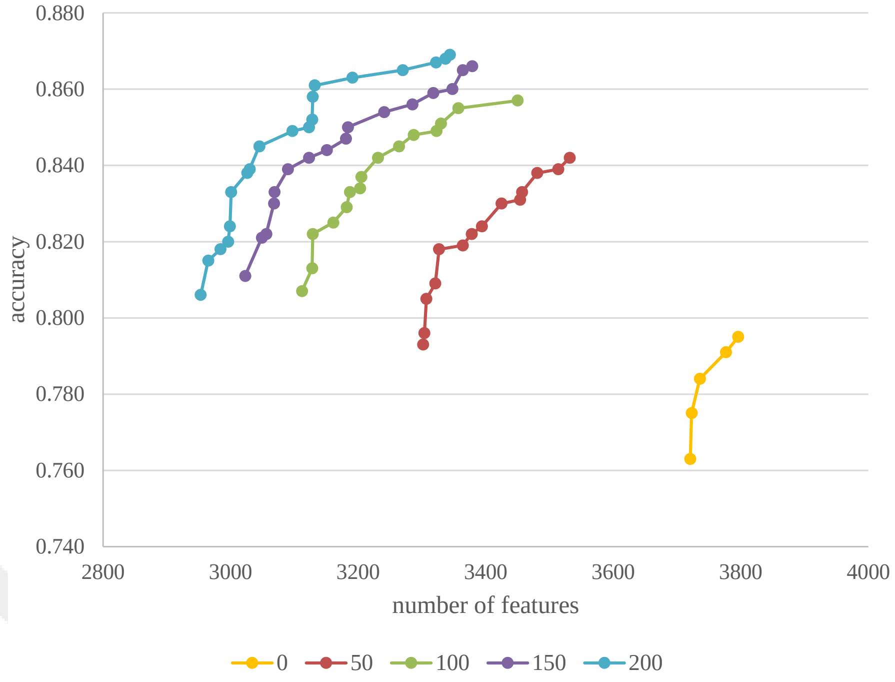
(e) The WHO dataset.

Model	BoW				GloVe			
	LR		SVM		LR		SVM	
	# of features	Accuracy	# of features	Accuracy	# of features	Accuracy	# of features	Accuracy
baseline	7028	0.8603	7028	0.8612	650	0.8117	650	0.8116
IGF	4038	0.8694	4038	0.8707	325	0.8168	325	0.8161
NSGA-II	3072	0.8697	2868	0.8722	191	0.8342	157	0.8370
IGF + NSGA-II	1683	0.8744	1615	0.8790	100	0.8287	88	0.8306

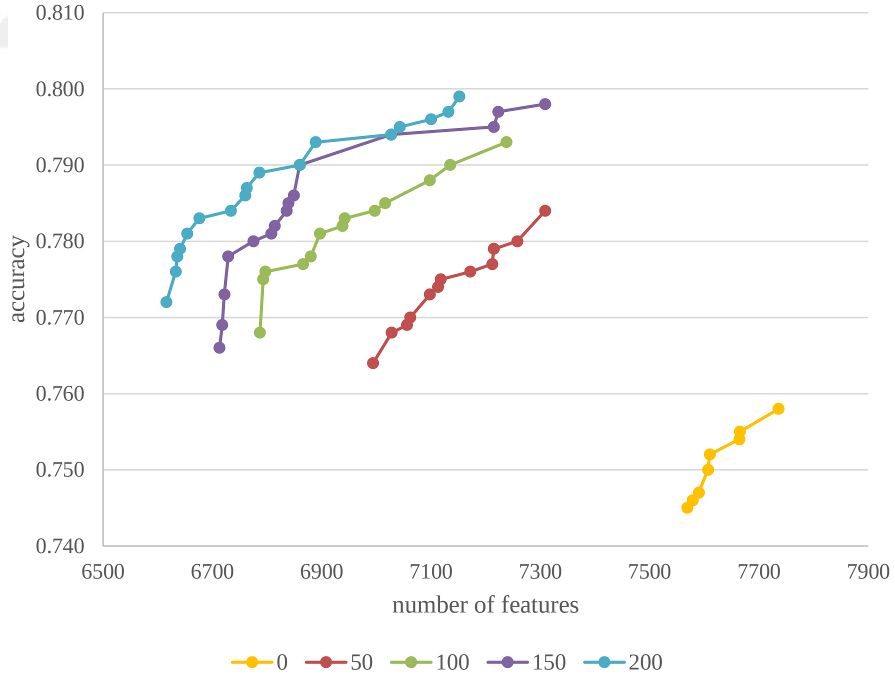
improve if a longer representation is chosen rather than the 50-dimensional GloVe vectors. Nevertheless, we can clearly see an improvement in accuracy over the baseline with our proposed model, even for this version of GloVe.

In Figure 4.3, we present the non-dominated solutions obtained through the generations on a two-dimensional plot. In the subfigures, the number of features and accuracy values are given on the x- and y-axis, respectively. We report the results up to 200 generations, in intervals of 50. Significant improvements in terms of both the number of features and accuracy are observed as the number of generations increases. For example, initially, the number of features is about 2000, and the accuracy is about 83% for the WHO dataset. With the proposed model, the number of features goes down to about 1450, and accuracy goes up to about 87%.

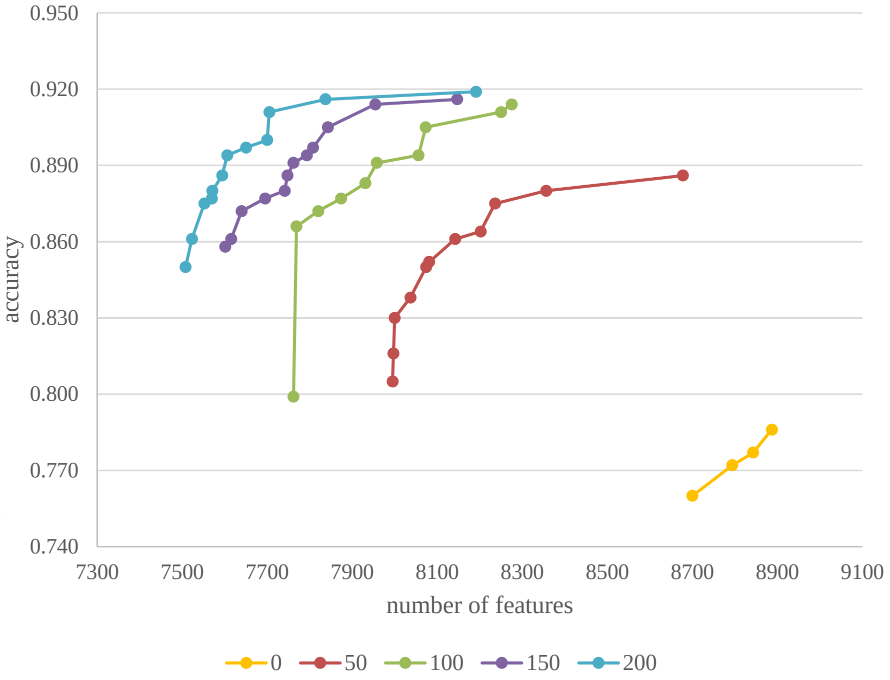
We provide the initial and final populations in Figure 4.4 to show that the proposed model evolves to approximate the optimal solution. The figures show that the initial population improves throughout the generations and gets closer to the ideal point, i.e.,



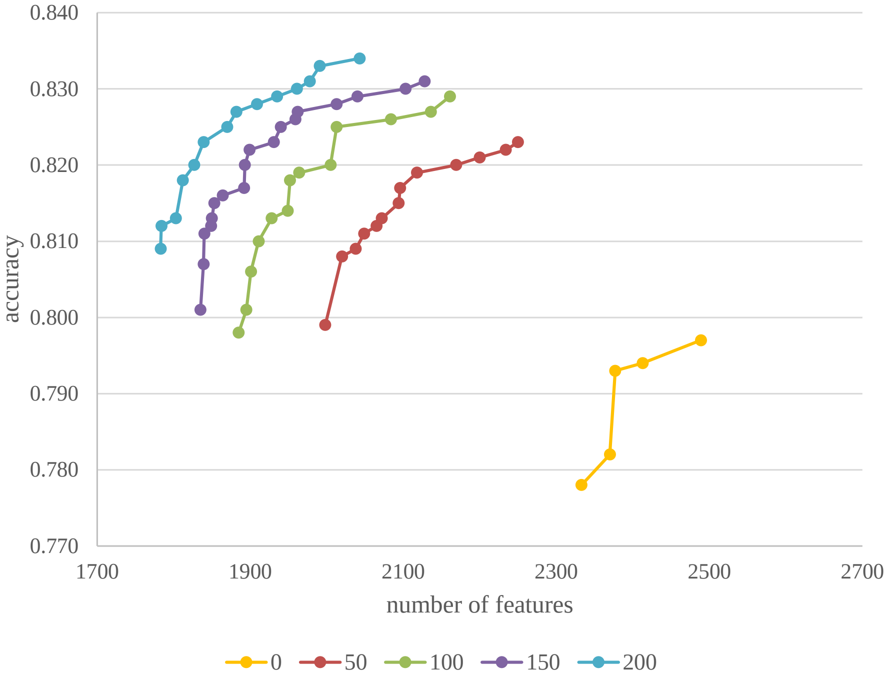
(a) The SST dataset.



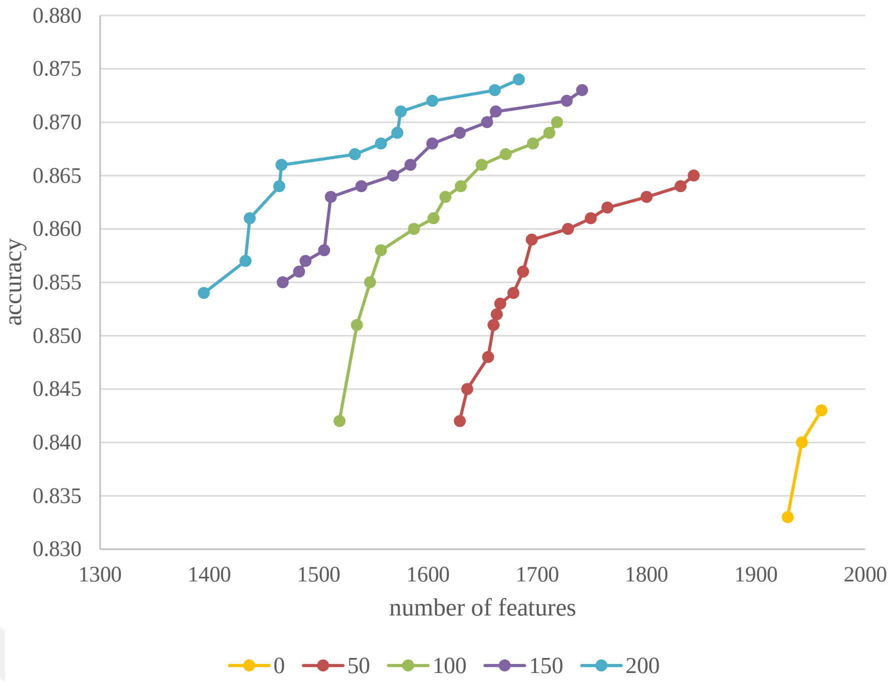
(b) The MR dataset.



(c) The S140 dataset.



(d) The IMF dataset.

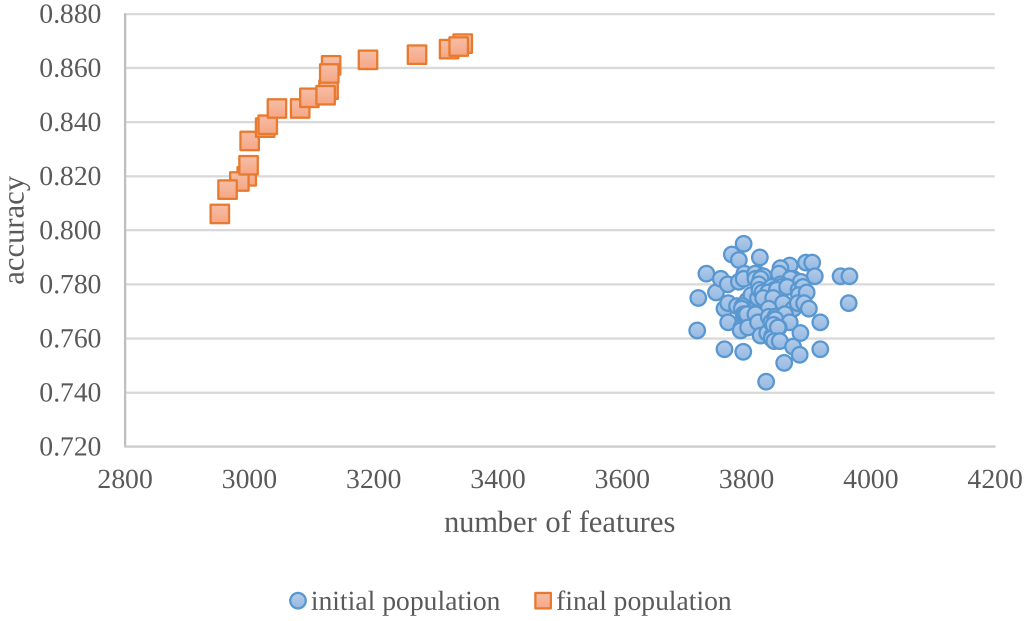


(e) The WHO dataset.

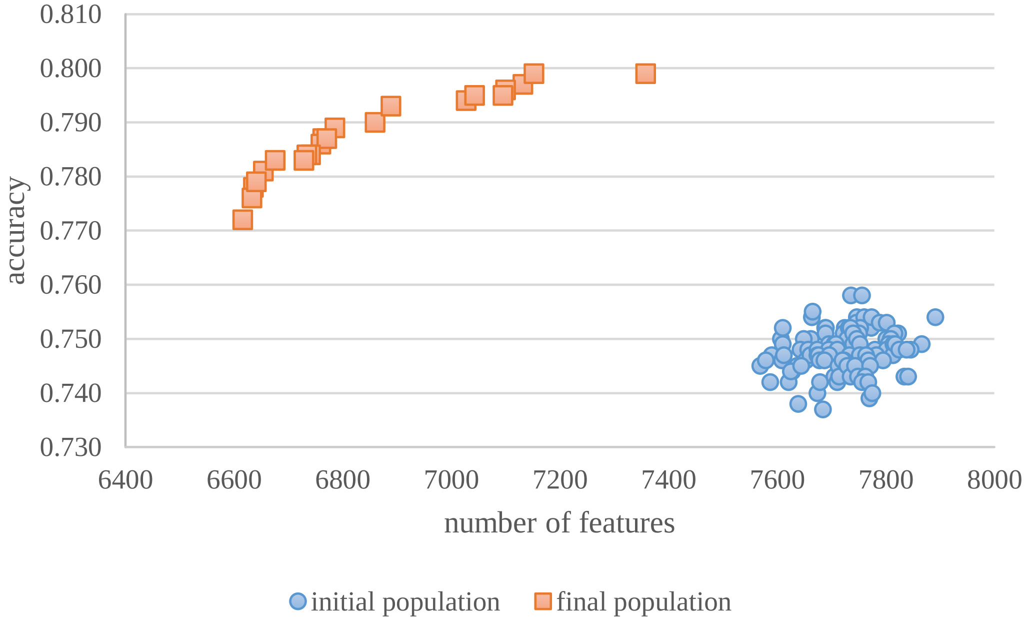
Figure 4.3: Evolution of the non-dominated solutions through generations with the proposed feature selection model.

the point where the number of features is one and the accuracy is 1.00. The individuals in the initial population are more scattered. In contrast, the non-dominated solutions in the final population fit a Pareto-like curve as suggested in the problem definition of Feature Selection (see Section 2.2.3).

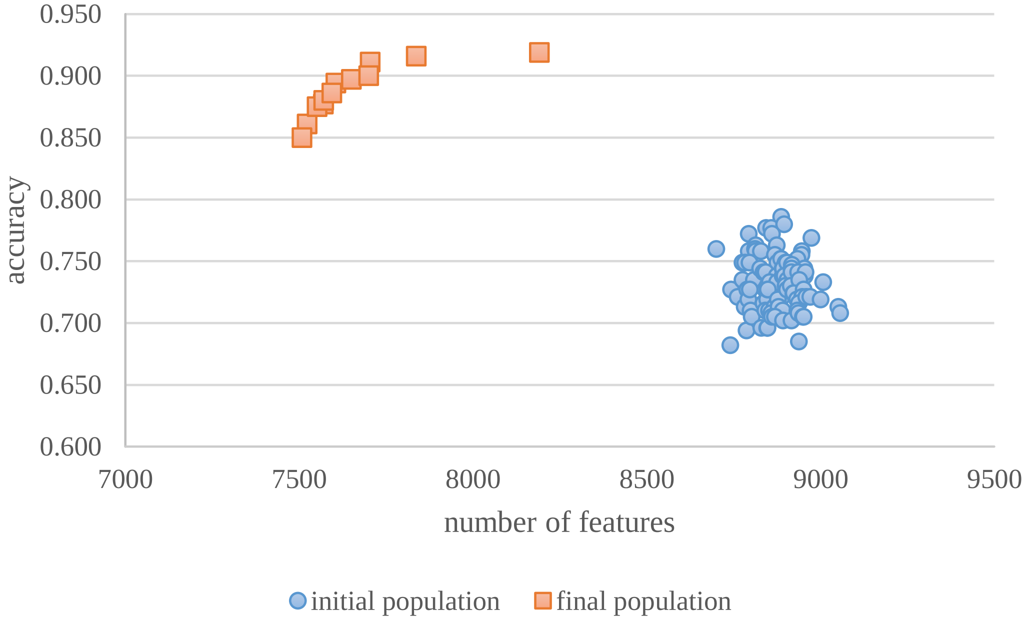
In Figure 4.5, we share the improvements in terms of the number of features, accuracy, and execution time after the proposed algorithm is applied with the LR classifier on BoW representation. The percentages above the bars in the subfigures present the amount of improvement in the related category and dataset. The figures show that the proposed algorithm decreases the number of features in the SST dataset by 78%. As the amount of data decreases, computation time reduces as well. We observe an 81% gain in the execution time of the classifier. Moreover, the proposed algorithm boosts accuracy by around 6%. Similar improvements are observed for the other datasets in the figure.



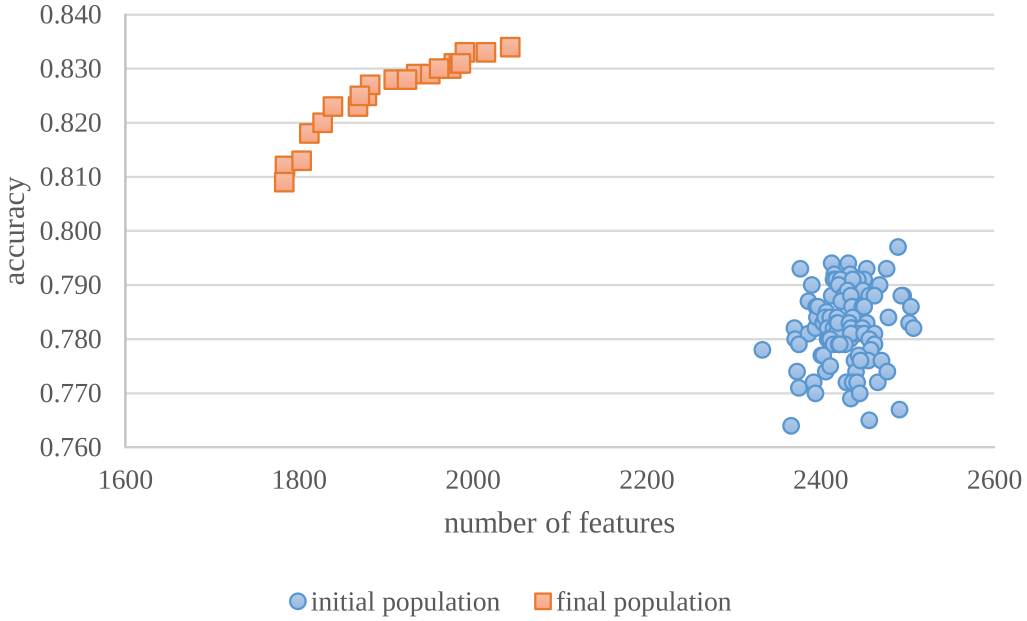
(a) The SST dataset.



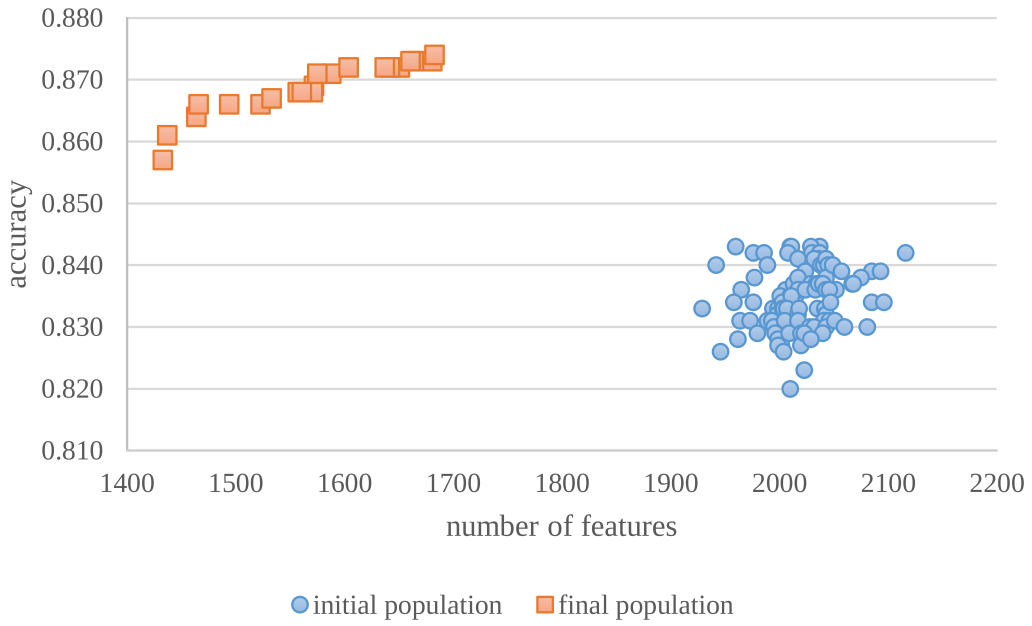
(b) The MR dataset.



(c) The S140 dataset.



(d) The IMF dataset.

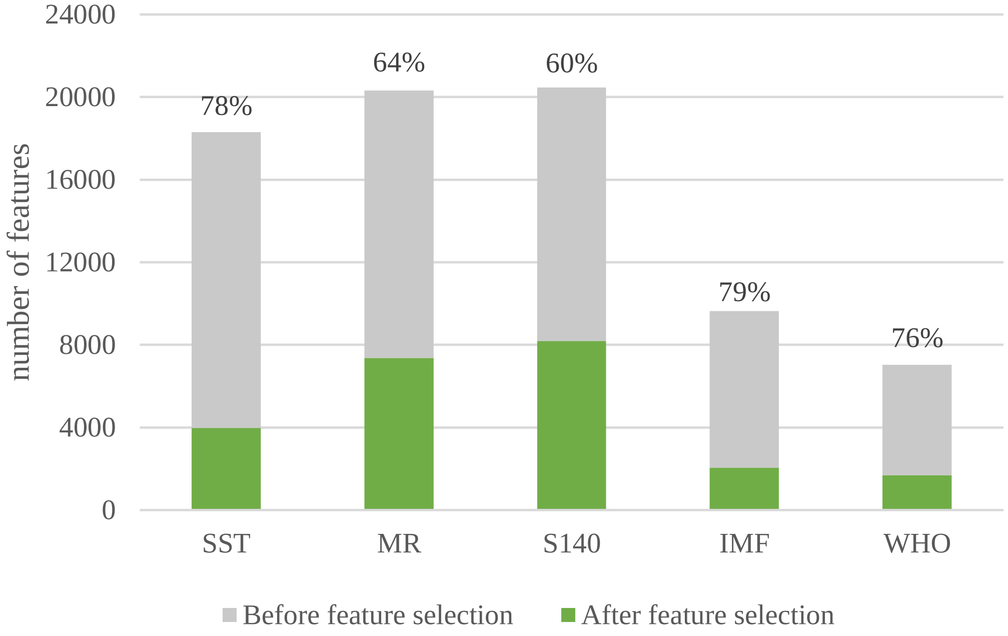


(e) The WHO dataset.

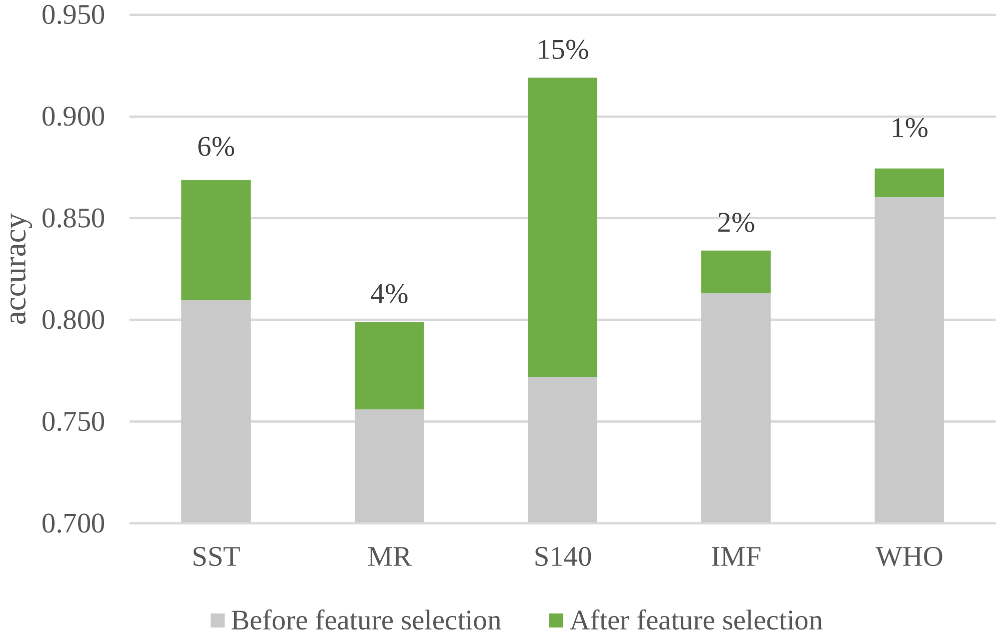
Figure 4.4: Initial population and the non-dominated solutions in the final population of the datasets with the proposed feature selection model.

In order to verify the effectiveness of the proposed model, we compared our results with off-the-shelf feature selection methods [84]. Table 4.2 presents the accuracy results for seven well-known feature selection methods along with the proposed model's accuracy with BoW. The feature size parameter of these methods is set the same as our proposed model (e.g., 3344 for LR in the SST dataset) to obtain a fair comparison. The results show that the proposed model outperforms all feature selection methods in all datasets regardless of the machine learning technique. As stated in the Introduction section, wrapper-based methods generally perform better than filter-based methods, with an additional computation cost in return. Our model achieves up to 20% more accuracy than the other techniques, as it exploits the power of wrapper-based methods for high prediction accuracy.

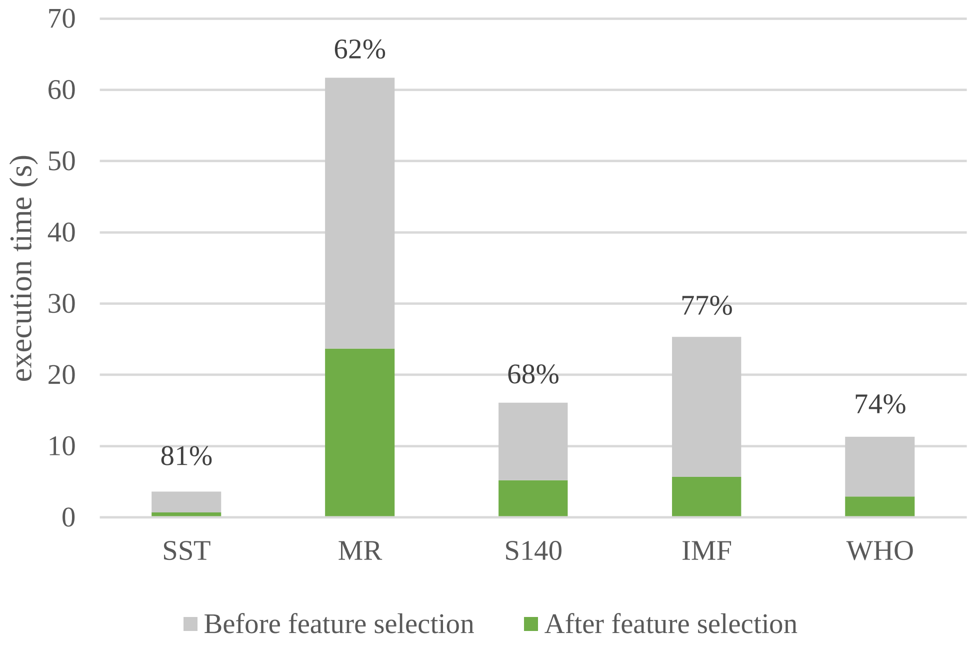
There exist many optimization algorithms for feature selection; however, the skills of these algorithms may change based on the problem they are applied to. According to the No Free Lunch theorem [126], there is no superior algorithm that prevails over every other algorithm in every domain. In this study, we developed a new multiobjective



(a) Number of features.



(b) Accuracy.



(c) Execution time.

Figure 4.5: Improvements in the number of features, accuracy, and execution time after the proposed feature selection model is applied.

feature selection algorithm for the sentiment analysis domain.

In BoW, the informative words are selected with the feature selection process as the features are the words. Therefore, the sentiment-oriented vocabulary of the dataset is decided with this representation. The classification accuracy increased with this sentiment-oriented vocabulary for all datasets, respectively. Similar to BoW, the proposed model decreased the number of features significantly and increased the accuracy noticeably with the GloVe representation. However, the semantics of feature selection with these two representations are different. A word embedding represents each word with a vector of latent features. Therefore, each dimension of the vector carries different hidden information. In GloVe, each dimension of the 50-dimensional word vectors represents one feature in our study. In addition, since the vectors are concatenated based on the words' order in the sentence, the word's position in the sentence also becomes important. As a result, the algorithm may select a different number of features from different word positions in the sentences to improve the sen-

Table 4.2: Comparison of the proposed feature selection model with off-the-shelf feature selection methods for all datasets.

Method	SST		MR		S140		IMF		WHO	
	LR	SVM	LR	SVM	LR	SVM	LR	SVM	LR	SVM
Fisher score	0.7594	0.7589	0.7607	0.7620	0.7688	0.7716	0.8172	0.8118	0.8514	0.8503
ReliefF	0.6907	0.6916	0.6865	0.6866	0.7493	0.7437	0.7826	0.7827	0.8391	0.8391
Trace ratio	0.7329	0.7339	0.7604	0.7602	0.7688	0.7716	0.8138	0.8125	0.8495	0.8501
Chi-square	0.7339	0.7339	0.7629	0.7626	0.7772	0.7772	0.8140	0.8116	0.8506	0.8510
F-statistics	0.7325	0.7339	0.7599	0.7608	0.7688	0.7716	0.8136	0.8120	0.8489	0.8499
Gini index	0.7637	0.7579	0.7593	0.7610	0.7604	0.7577	0.8151	0.8112	0.8503	0.8502
T-score	0.7363	0.7397	0.7606	0.7612	0.7688	0.7660	0.8146	0.8135	0.8535	0.8532
Proposed model	0.8686	0.8796	0.7989	0.7955	0.9192	0.9415	0.8337	0.8368	0.8744	0.8790

timent classification performance. With this approach, our model infers which words and their hidden features contribute more to the sentiment classification task. Moreover, representing texts with word embeddings has become a de facto standard in the NLP literature [51]. Once sentences are built using word embeddings, they are fed into deep learning architectures, such as Convolutional Neural Networks or Long-Short Term Memory networks, as input. These networks determine the weights of each feature in the input separately, hence, possibly approximating the weights of some features to zero. Even though our model does not utilize a neural network architecture, it employs a similar idea and nullifies the weights of non-selected features.

There are many reasons why our proposed algorithm can obtain competitive results. Even though evolutionary algorithms evolve through generations and approximate the optimal solution, their computation cost increases excessively as the chromosome size increases. NLP tasks, such as sentiment analysis, are known to have enormous data sizes. As we target to improve the sentiment classification task, we employ an intelligent technique, i.e., filter-based feature selection based on information gain values, on our data before we run our evolutionary algorithm. With this approach, we shrink the chromosome size for our evolutionary algorithm, which boosts the performance in return. In addition, many algorithms depend on an extensive parameter tuning step to achieve better results. On the other hand, our proposed model does not rely on parameter tuning before execution, making it a compelling approach for sentiment classification problems.

CHAPTER 5

ENHANCING WORD EMBEDDINGS FOR SENTIMENT ANALYSIS

Word embeddings have become the de-facto tool for representing text in NLP tasks, as they can capture semantic and syntactic relations, unlike their precedents, such as BoW. Although word embeddings have been employed in various studies in recent years and have proven to be effective in many NLP tasks, they are still immature for sentiment analysis, as they suffer from insufficient sentiment information. General word embedding models pre-trained on large corpora with methods such as Word2Vec or GloVe achieve limited success in domain-specific NLP tasks. On the other hand, training domain-specific word embeddings from scratch requires a high amount of data and computation power. In this chapter, we target both shortcomings of pre-trained word embeddings to boost the performance of domain-specific sentiment analysis tasks. We propose a model that refines pre-trained word embeddings with context information and leverages the sentiment scores of sentences obtained from a lexicon-based method to further improve performance. Experiment results show that the proposed model significantly increases the accuracy of sentiment classification.

5.1 Related Work

In an effort to improve the accuracy of sentiment analysis, recent work in NLP literature has focused on creating better word embedding models for text representation. Tang et al. [62] developed several neural networks which encode context and sentiment information into word embeddings. However, their approach cannot handle unknown words. Similarly, Liu et al. [127] stated that most word embeddings do not discriminate words that have the same phonetics but different meanings. To address this problem, they presented a topic-specific word embedding technique. They utilized Latent Dirichlet Allocation to extract topics and Collapsed Gibbs Sampling to match each word with the topics. Experiment results demonstrated that the proposed model outperforms off-the-shelf word embeddings. However, they noted that the necessity of defining the topic number in advance is a limitation of the study. Bojanowski et al. [128] proposed a method that learns word representations by utilizing character n-grams. They aimed to enhance word embeddings by considering the morphology of the words. Kamkarhaghighi and Makrehchi [129] introduced a method to enhance document representation using two well-known word embeddings, GloVe and Word2Vec. They created a content tree-based word embedding technique by tuning the values of the available word vectors via the correlation information between words. Yu et al. [61] proposed updating the positions of words in existing embeddings according to their ten closest words and their valence (sentiment) scores. They moved each word towards the positive or negative field, according to their neighbors. Their proposed method improves the accuracy of the sentiment analysis task when compared with GloVe, Word2Vec, and HyRank. Recently, Rezaeinia et al. [51] proposed enhancing word vectors with extra information and named it Improved Word Vector. The method combines the vectors retrieved from Word2Vec/GloVe, lexicons, POS tagging, and word position algorithm. Then it uses this final (combined) vector as the input to their deep learning model. The effectiveness of the proposed method was verified on the well-known movie and customer review datasets.

Although these word representation methods are reported to be effective, there is no method that integrates the context and sentiment information efficiently for the sentiment analysis task, handles unknown words, and does not require hyperparameter tuning.

5.2 Model

Our proposed model for word embeddings refines word vectors obtained using effective pre-trained models in two steps: contextual refinement and valence addition. Figure 5.1 shows a graphical representation of how the model works.

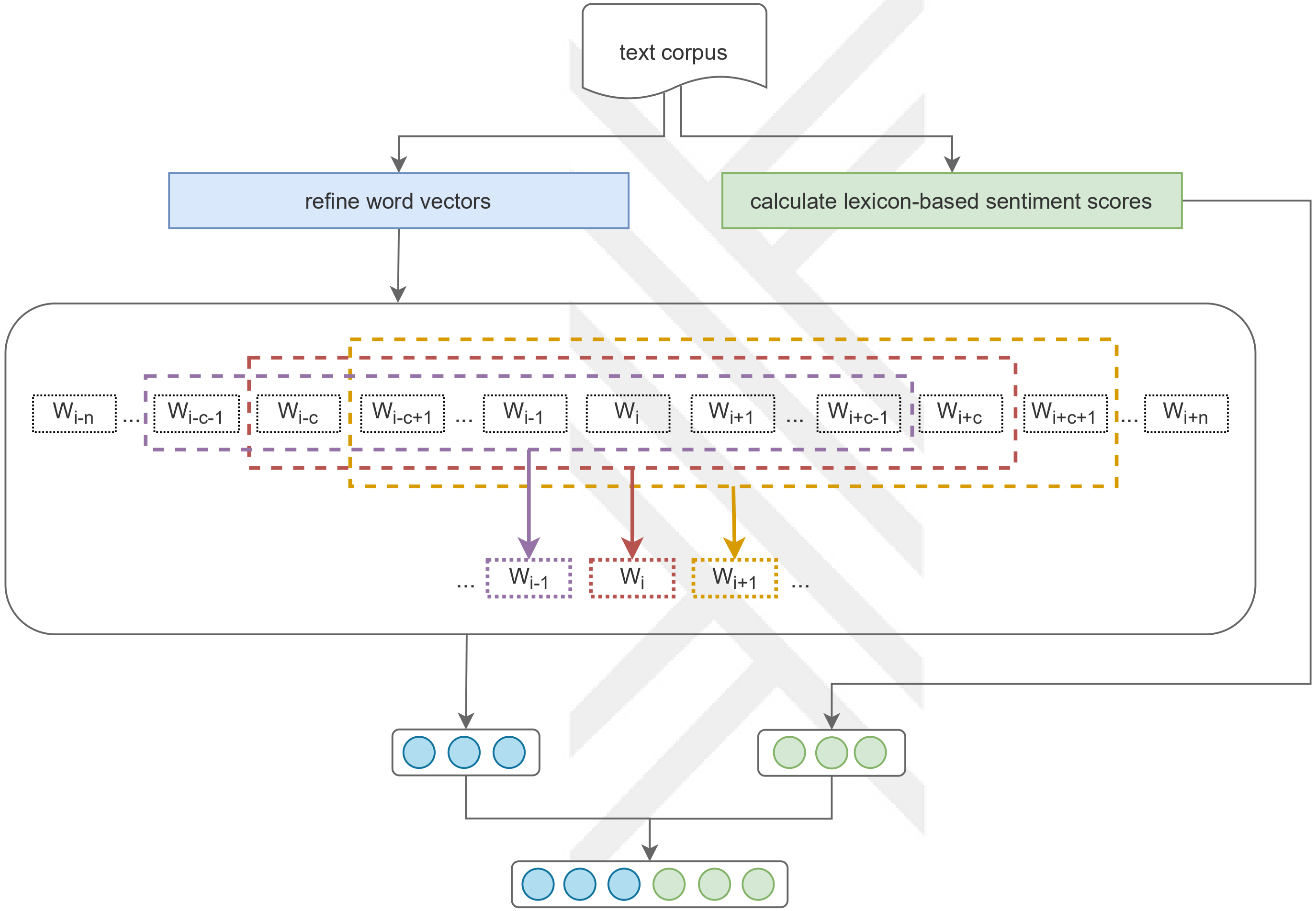

text corpus
refine word vectors
calculate lexicon-based sentiment scores
W_{i-n} … W_{i-c-1} W_{i-c} W_{i-c+1} … W_{i-1} W_i W_{i+1} … W_{i+c-1} W_{i+c} W_{i+c+1} … W_{i+n}
… W_{i-1} W_i W_{i+1} …

Figure 5.1: Proposed sentiment- and context-refined word embeddings model.

The contexts of words, which are constituted by surrounding words [62], carry critical information for the sentiment analysis task [130]. In our model, we aim to add context information via the preceding and following words. For this purpose, we update every token's vector values by averaging the word vectors of its preceding and following c tokens. In addition to context, this approach helps with the unknown words (words that do not exist in the dictionary of pre-trained word embeddings) as these words may be represented through their neighbors' vectors.

Let a sentence s consist of n tokens as follows:

$s = [t_1, t_2, ..., t_n]$. The word vector w_i^* of a token t_i is calculated as below:

$$w_i^* = \frac{\sum_{i-c}^{i+c} w_i}{2c+1} \tag{5.1}$$

where w_i is the pre-trained word vector of the i-th token, c is the neighbor radius, and w_i^* is the refined word vector. The selection of neighbor radius depends on the dataset specifications. In order to choose the optimal c value, we performed a preliminary study, details of which are given in Section 5.3.2.

A sample context refinement for a sentence is shared in Table 5.1. The table presents the GloVe pre-trained word embeddings for each word and their context-refined versions. The sample sentence, "This is a truly truly bad movie.", is retrieved from one of our benchmark datasets, SST. The neighbor radius value, c, is set as 2 in this sample. Accordingly, the refined word embedding of the first appearing 'truly' is calculated by averaging the pre-trained vectors of 'is', 'a', 'truly', 'truly', and 'bad'. For instance, the first dimension of the refined word vector for 'truly' is computed as follows: $((\text{-}0.175) + (\text{-}0.297) + (0.267) + (0.267) + (0.309))/5 = 0.074$. When there are fewer words than c in the neighborhood, the maximum number of available neighbors is considered in the calculation. For example, the refined word embedding of the word 'this' is calculated by averaging the vectors of 'this', 'is', and 'a' since the neighbor radius is 2. Similarly, the refined word embedding of 'is' is calculated by averaging the vectors of 'this', 'is', 'a', and 'truly'. In GloVe, every word has one specific corresponding vector without concerning the context of the word, e.g., the word 'truly' appears twice with the same vector in the sample. After our proposed

Table 5.1: Sample context refinement in the proposed word embedding model.

Word	Plain word embeddings	Context-refined word embeddings
this	[-0.204, 0.164, 0.042, -0.137, -0.298, …]	[-0.225, 0.163, 0.065, -0.229, -0.202, …]
is	[-0.175, 0.230, 0.249, -0.205, -0.123, …]	[-0.102, 0.131, 0.100, -0.264, -0.056, …]
a	[-0.297, 0.094, -0.097, -0.344, -0.185, …]	[-0.028, 0.112, 0.121, -0.285, 0.032, …]
truly	[0.267, 0.035, 0.206, -0.369, 0.383, …]	[0.074, 0.053, 0.097, -0.260, 0.062, …]
truly	[0.267, 0.035, 0.206, -0.369, 0.383, …]	[0.082, -0.017, 0.048, -0.221, 0.113, …]
bad	[0.309, -0.127, -0.078, -0.011, -0.146, …]	[0.176, -0.045, 0.085, -0.190, 0.188, …]
movie	[-0.138, -0.122, 0.005, -0.010, 0.131, …]	[0.146, -0.071, 0.044, -0.130, 0.123, …]

model is applied, the two appearances of ‘truly’ are represented with different vectors since context information is integrated into the vectors according to the word’s position in the sentence and neighbors.

Moreover, we hypothesize that adding sentiment information along with contextual information would improve prediction accuracy. Therefore, in the second step of our model, we add sentiment predictions retrieved from VADER (see Section 2.1 for details) to our model. The sentiment vector is filled with positive labels when the compound score of VADER is greater than or equal to 0.05. When the compound score is less than or equal to -0.05, we use negative. The scores between -0.05 and 0.05 are considered neutral.

Finally, the combination of the contextually refined word vectors and sentiment predictions constitutes the proposed domain-specific word embeddings. The combination is performed through a concatenation operation, i.e., the final word embedding is generated by end-to-end concatenation of the context-refined word embeddings and the vector of lexicon-based sentiment scores. Then, model training is carried out with these refined word embeddings.

This algorithm’s time complexity is equal to the multiplication of the maximum number of tokens of all sentences, neighbor radius, and the number of instances.

We note that our model can also be applied on both binary and multi-class datasets (e.g. positive, negative, and neutral). For this purpose, the categorization regarding the VADER's compound score should be adjusted according to the label classes.

5.3 Experiments

5.3.1 Settings

For the SST and S140 datasets, the instances were split into the train and test sets in the original data. Therefore, we utilized the predefined test data in the testing part of our study. On the other hand, we performed 10-fold cross-validation on the other datasets, as there were no predefined splits.

For the word embeddings, we initialized the unknown words (not present in the dictionary of pre-trained word embeddings) with random numbers between -0.25 and 0.25 to comply with the variance of the available word embeddings.

In this study, differently from other models, the *solver* parameter of LR is set to *liblinear*, and the regularization parameter, *C*, of SVM is set to *0.01*.

5.3.2 Results and Discussion

We held a preliminary study to find out the most promising context window, i.e., c. Experiments for this study were carried out on different dimensions of GloVe using both LR and SVM as the classifiers for various c values. Table 5.2 presents the accuracy results of the experiments for the SST dataset without applying any cleaning operation.

In the table, the accuracy consistently increases as the GloVe dimension increases. Therefore, we opted for 300-dimensional word embeddings in the experiments carried out to test the proposed model. In 300-dimensional embedding results, we detected high accuracy values when $c = 5$ for both LR and SVM. Therefore, we selected the c value as 5. This choice is in line with the theory as smaller values of c could risk not capturing the context, while larger values could lead to overlapped contexts.

Table 5.2: Preliminary study results to find the most promising context window (c) for the proposed word embedding model.

Neighbor radius	GloVe dimension							
	50		100		200		300	
	LR	SVM	LR	SVM	LR	SVM	LR	SVM
c = 1	0.6907	0.6936	0.6916	0.6931	0.6964	0.6936	0.6859	0.6830
c = 2	0.6763	0.6763	0.6859	0.6888	0.6979	0.6931	0.7037	0.7022
c = 3	0.6772	0.6763	0.6960	0.6921	0.7032	0.7003	0.6931	0.6931
c = 4	0.6883	0.6854	0.6988	0.6955	0.7080	0.7075	0.7137	0.7123
c = 5	0.6849	0.6835	0.7037	0.7003	0.7118	0.7109	0.7277	0.7277
c = 6	0.6902	0.6902	0.6993	0.6984	0.7128	0.7109	0.7075	0.7075
c = 7	0.6768	0.6763	0.7008	0.6984	0.7142	0.7113	0.7109	0.7118
c = 8	0.6787	0.6806	0.7027	0.7012	0.7262	0.7205	0.7214	0.7233
c = 9	0.6830	0.6825	0.7032	0.7012	0.7181	0.7171	0.7253	0.7257
c = 10	0.6796	0.6806	0.7070	0.7037	0.7277	0.7219	0.7229	0.7286

Table 5.3 presents the experimental results of the proposed model for all datasets. In the table, we provide accuracy results for two pre-trained word embeddings (GloVe and Word2Vec) on two different machine learning architectures (LR and SVM). Moreover, we present the incremental results as we build our model: 'Plain' provides the baseline model accuracies, 'Context-refined' provides the results of the model with context-refined word embeddings, and 'Sentiment & context-refined' provides the results for our proposed model. The results show that context-refined word embeddings increase the accuracy for all word embedding and machine learning models when compared with the baseline model. The only exception to that phenomenon is the S140 dataset when the text representation technique is GloVe.

Our proposed model outperforms the baseline model regardless of the word embedding or dataset when executed with either machine learning technique. For the SST, it increases the classification accuracy by approximately 5% for both text representation and machine learning techniques. The gained performance percentage is similar

Table 5.3: Accuracy results of the proposed word embedding model for all datasets.

(a) The SST dataset.

		LR	SVM
GloVe	Plain	0.7592	0.7605
	Context-refined	0.7772	0.8062
	Sentiment & context-refined	0.7991	0.8255
Word2Vec	Plain	0.7894	0.7927
	Context-refined	0.8294	0.8442
	Sentiment & context-refined	0.8319	0.8461

(b) The MR dataset.

		LR	SVM
GloVe	Plain	0.6682	0.6795
	Context-refined	0.7204	0.7364
	Sentiment & context-refined	0.7219	0.7418
Word2Vec	Plain	0.7052	0.7210
	Context-refined	0.7504	0.7659
	Sentiment & context-refined	0.7516	0.7717

(c) The S140 dataset.

		LR	SVM
GloVe	Plain	0.6797	0.6769
	Context-refined	0.6212	0.6657
	Sentiment & context-refined	0.6964	0.7493
Word2Vec	Plain	0.7382	0.7298
	Context-refined	0.7521	0.7855
	Sentiment & context-refined	0.7660	0.8273

(d) The IMF dataset.

		LR	SVM
GloVe	Plain	0.7391	0.7472
	Context-refined	0.7772	0.7942
	Sentiment & context-refined	0.7849	0.7474
Word2Vec	Plain	0.7618	0.7733
	Context-refined	0.8015	0.8120
	Sentiment & context-refined	0.8080	0.7847

(e) The WHO dataset.

		LR	SVM
GloVe	Plain	0.8268	0.8259
	Context-refined	0.8399	0.8480
	Sentiment & context-refined	0.8434	0.8508
Word2Vec	Plain	0.8431	0.8512
	Context-refined	0.8593	0.8604
	Sentiment & context-refined	0.8604	0.8569

in other datasets. The proposed model increases the classification accuracy by up to 7%, 10%, 5%, and 3% for the MR, S140, IMF, and WHO datasets, respectively. Therefore, it is obvious that integrating context and sentiment information into the pre-trained word embeddings has merit in enhancing the performance of sentiment classification.

CHAPTER 6

UNSUPERVISED LEARNING FOR SENTIMENT ANALYSIS

Supervised learning algorithms developed for sentiment analysis have significantly improved recently. However, unsupervised learning algorithms could be more practical in many settings, especially when there is a lack of experts to perform the labeling. In this chapter, we present a sentiment-aware unsupervised model that utilizes the confidence scores provided by pre-trained BERT models and propagates the sentiment information through similarity information. We present empirical results that illustrate the improvements achieved in different domains by the proposed model.

6.1 Related Work

Research on sentiment analysis is rapidly evolving, especially in terms of supervised learning algorithms. Various approaches have been proposed to build sentiment classification models, including feature-based [34] or neural-network-based [131] models. Many studies improve the performance of their approach by incorporating internal or external knowledge into existing models, such as sentiment-aware BERT [74] or knowledge-enabled BERT [132].

Although supervised learning algorithms generally provide high-performance solutions, they require experts for data annotation and too much computation power to ignore for the training. Therefore, in many settings, unsupervised learning algorithms could be more convenient than supervised ones. Moreover, sentiment classification can be considered a domain-dependent task since the sentiment of the content may contradict in different contexts [63]. Therefore, a model trained for a domain may not be suitable to be applied to another domain. However, a sufficient amount of labeled data for every domain is often impractical. Therefore, the research on unsupervised learning methods becomes worthwhile to handle the domains with limited resources.

Pre-trained language models, such as BERT [54] or ELMo [73], have shown great success in various NLP tasks [133, 134]. Many studies utilized pre-trained models by fine-tuning them for various tasks such as question-answering [135], machine translation [136], and sentiment analysis [137]. For the unlabeled datasets, however, pre-trained language models might not be effective, especially when the domains of the training data and target data are not similar [138]. Nonetheless, they provide a good basis for the evaluation of unlabeled datasets. Inspired by the existing studies, we aim to improve the performance of existing pre-trained language models on unlabeled datasets by utilizing the intrinsic properties of the data with the help of different similarity metrics.

Unsupervised learning studies mostly focus on statistical approaches that use lexicon-sentiment pairs. Some of the well-known lexicon-based sentiment analysis tools are SentiStrength [64], SentiWordNet [65], and VADER [66]. Other than the lexicon-based approaches, BERT provides an off-the-shelf language model that does not necessarily require labeled data, which makes it appealing for unsupervised NLP tasks [139]. However, it is still essential to find a pre-trained model whose training material aligns with the target domain. Other than that, when training is not possible, self-training may boost pre-training [140]. There exist many studies that improve unsupervised learning performance via self-training [141, 142, 143]. Although they perform better than the lexicon-based approaches, they execute slower due to the nature of the training process [45]. There also exist studies that utilize co-occurrence information to improve the performance of the classification task [144, 145, 146]. For example, Angin and Bhargava [147] proposed a model that identifies objects in highly complex scenes. The model first extracts the objects with the most confident estimates. Then, they propagate this information according to the co-occurrence relations and identify the other objects in the scene in an iterative fashion. In our study, we employ a similar strategy by propagating sentiment information retrieved from a pre-trained BERT model from the most confident estimate to the least in a recursive way.

6.2 Model

Our approach consists of two stages, as presented in Figure 6.1. The first stage involves the utilization of a pre-trained BERT model fine-tuned for the sentiment classification task to predict all sentences as positive or negative. Then, we determine a threshold value based on a predetermined percentile. We assume that all instances having a confidence score larger than this threshold are correctly identified by the pre-trained model and assign their predicted values as their final predictions. This assignment concludes the first stage.

In the second stage of our algorithm, we propagate the information within the sentences having final predictions onto the sentences that are yet to be finalized. For this purpose, we employ the k-Nearest Neighbors (*kNN*) algorithm to assign the final predictions of the non-finalized instances. In our implementation, the pre-trained BERT model's prediction is considered the first neighbor regardless of its confidence score. The remaining $k - 1$ neighbors are selected among the finalized instances. Additionally, at the beginning of the second stage, we sort each sentence based on its confidence score in descending order. The reason for sorting is to process the sentences with higher confidence scores first. Accordingly, those instances may contribute to the prediction of sentences with lower confidence scores.

In order to apply the *kNN* algorithm to propagate the information to similar sentences, we need to be able to calculate the similarities between them. For this purpose, we leverage two measurements.

BERT does not map sentences to a vector space considering the common similarity measures, such as cosine similarity. In other words, the cosine similarity of two sentences does not imply any meaningful information in terms of how similar they semantically are. To overcome this issue, Reimers and Gurevych [148] proposed Sentence-BERT (or SBERT), which fine-tunes BERT in this regard. In our proposed model, we utilize SBERT to calculate the similarities between sentences. The cosine similarity between the vector embeddings of the sentences constitutes the first metric.

We use SentiWordNet [65] as the second metric to calculate the distances between the sentiments of the sentences. For this purpose, we find all the nouns, adjectives, and

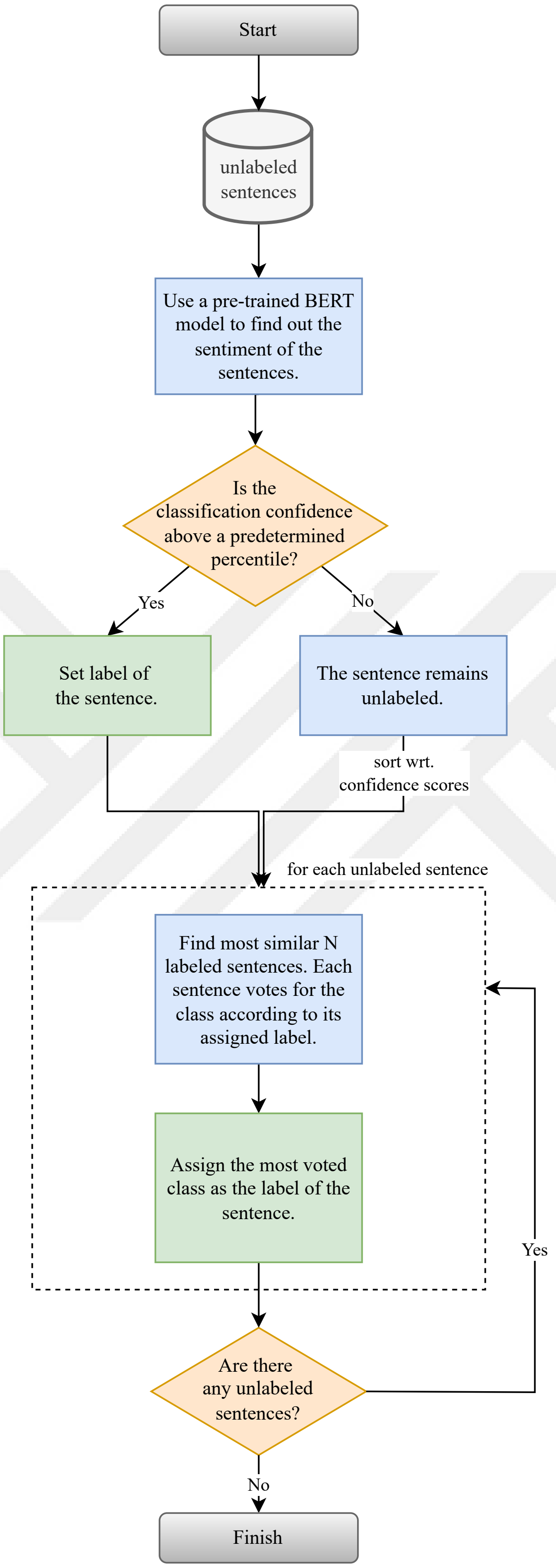

Start
unlabeled sentences
Use a pre-trained BERT model to find out the sentiment of the sentences.
Is the classification confidence above a predetermined percentile?
Yes
No
Set label of the sentence.
The sentence remains unlabeled.
sort wrt. confidence scores
for each unlabeled sentence
Find most similar N labeled sentences. Each sentence votes for the class according to its assigned label.
Assign the most voted class as the label of the sentence.
Yes
Are there any unlabeled sentences?
No
Finish

Figure 6.1: Proposed unsupervised learning model.

adverbs in the sentences and filter out the rest. Then, we lemmatize each word and query it over WordNet [149]. If the lemma exists in WordNet, we retrieve the positive and negative sentiment scores of its most common definition from SentiWordNet. We sum the positive and negative scores of all lemmas separately and concatenate both values into a vector as $<positive\_score, negative\_score>$. The cosine similarity between these vectors constitutes the second metric.

Once we obtain both similarities, we simply use their average as our similarity metric within the *kNN* algorithm. A single pass over all the non-finalized sentences allows us to finalize their predictions.

We emphasize that the proposed model never uses the actual labels for classification. The actual labels are only utilized to calculate the performance of the proposed model.

The time complexity of the algorithm highly depends on the selected BERT models' inference times and dataset size.

Although we used a pre-trained model that provides predictions for two classes (positive and negative) in the experiments, we note that our model can also be applied on multi-class datasets (e.g. positive, negative, and neutral). For this purpose, a pre-trained BERT model providing multi-label predictions should be employed in the first stage of our model. The rest of our model would handle multi-class datasets gracefully as it uses sub-methods that are automatically applicable for multi-class datasets.

6.3 Experiments

6.3.1 Settings

The SST and S140 datasets consist of predetermined train and test sets. In this study, we only used the train set instances of these datasets.

For initial sentiment classification, we used the pre-trained *distilbert-base-uncased-finetuned-sst-2-english* BERT model.

To calculate sentence similarities, we used the pre-trained *bert-base-nli-mean-tokens*

SBERT model for sentence embeddings. We provided the tokenizer with three parameters, as follows. Both *padding* and *truncation* are set to *True*, and the *max_length* parameter is selected as *512*.

6.3.2 Results and Discussion

In Figure 6.2, we present the changes in accuracy as the pre-trained BERT model's confidence level varies. The results are similar for the S140, IMF, and WHO datasets: it is larger than 0.97 accuracy for the instances in the top 1% confidence rankings. It linearly drops down to a range between 0.70 and 0.80, as the instances with lower confidence rankings are included in the calculation. This finding shows that our initial assumption is correct, i.e., generic pre-trained models perform well for at least a specific portion of the dataset, even though it was not trained specifically for that domain.

We observe that the results for the SST and MR datasets are not in line with the remaining three datasets. The reason for that is the utilized pre-trained *distilbert-base-*

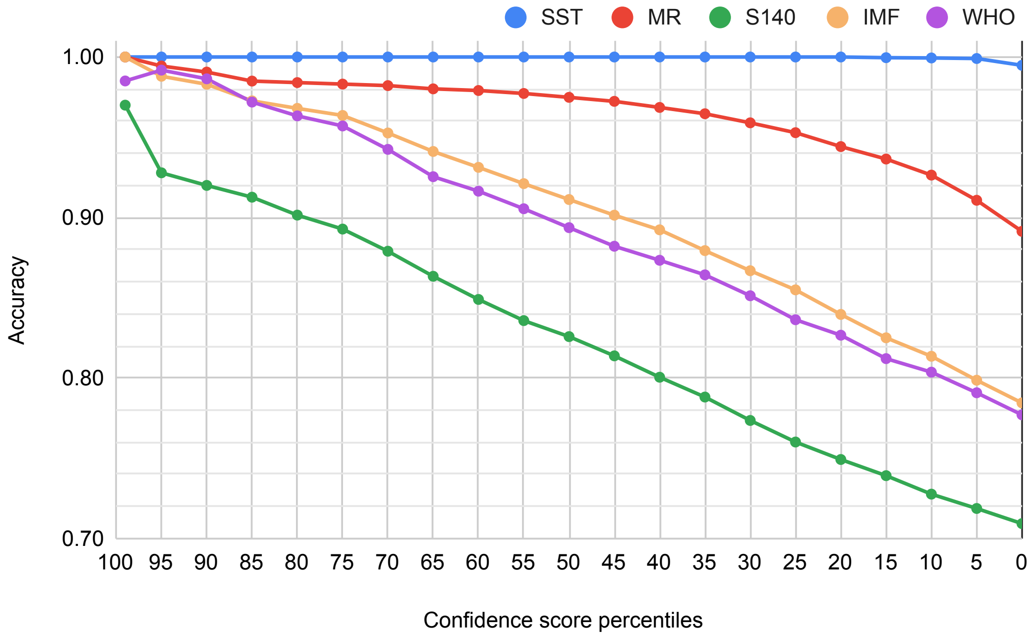


Figure 6.2: Change in accuracy of finalized instances with varying confidence score percentiles for all datasets.

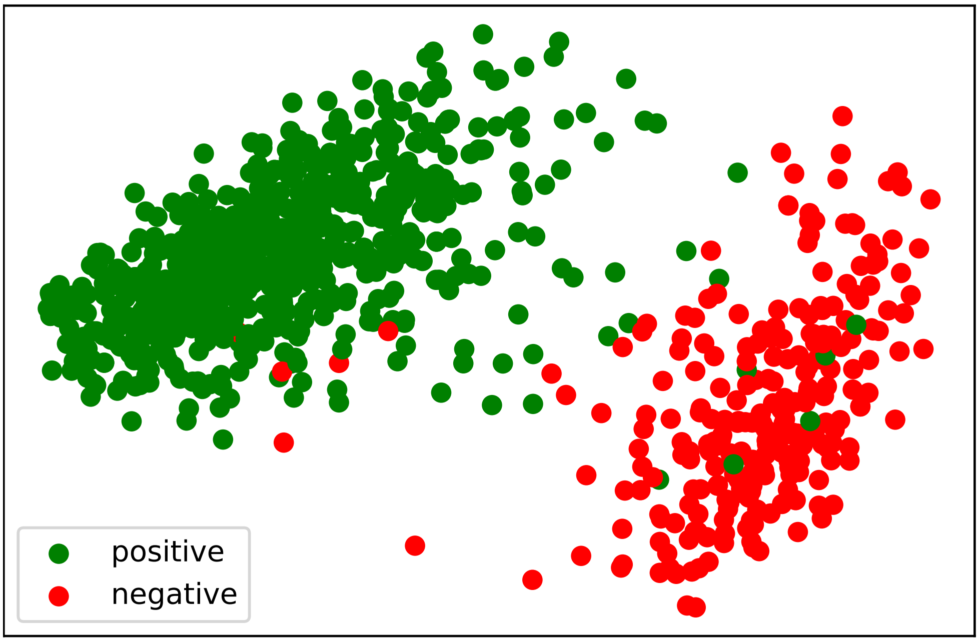


Figure 6.3: PCA plot of the instances within the top 10% confidence scores and their actual labels in the IMF dataset.

uncased-finetuned-sst-2-english BERT model was fine-tuned on the SST dataset[1]. Therefore, the model has already overfit the SST data. Moreover, SST and MR datasets share the same domain, i.e., movie reviews. Hence, the MR dataset lies in between SST and the datasets from other domains. Accordingly, in our following analysis, we will include the results for all the datasets, but we will focus on the results of S140, IMF, and WHO datasets, which conform to the nature of unsupervised learning better.

In light of the information gathered from the figure, we select the instances in the top 10% confidence rankings as the initial finalized set of instances in the following part of this study.

In Figure 6.3, we present the sentences of the IMF dataset in the top 10% confidence rankings plotted with Principal Component Analysis (PCA). PCA is useful for reducing the dimensionality of the dataset into two, thus making it suitable for plotting. The colors represent the actual labels of the sentences. The green color represents the sentences with a positive sentiment, and the red color indicates that its respective

[1] `https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english`

Table 6.1: Accuracy values for varying number of neighbours (k) for all datasets.

k	SST	MR	S140	IMF	WHO
3	0.9149	0.8241	0.7255	0.8188	0.8105
5	0.8989	0.8086	0.7286	0.8278	0.8241
7	0.8929	0.8014	0.7301	0.8314	0.8285
9	0.8882	0.7950	0.7299	0.8342	0.8347
11	0.8852	0.7916	0.7297	0.8346	0.8393
13	0.8830	0.7915	0.7287	0.8322	0.8402
15	0.8784	0.7872	0.7277	0.8329	0.8414
17	0.8771	0.7864	0.7283	0.8331	0.8416
19	0.8768	0.7846	0.7277	0.8320	0.8436
21	0.8735	0.7834	0.7282	0.8328	0.8450

sentence has a negative sentiment. It is clear from the figure that the two classes are almost perfectly separable. This is the main reason our proposed model, which utilizes the information within these separated instances in determining the classes of less confident sentences, should boost the classification performance.

In Table 6.1, we present the changes in accuracy with varying numbers of neighbors (k) voting for the final prediction of each sentence. For the IMF dataset, the optimum k value is 11. For the S140 dataset, the accuracy is close to the maximum value when k is 11. For the WHO dataset, the accuracy increases as the k value increases. However, the amount of increment decreases for growing k values. Therefore, in the rest of our study, we select the k value as 11.

We present the experiment results in Table 6.2. This table also presents the ablation study that reflects the effects of including each step in our proposed model. From the table, we see that the lexicon-based SentiWordNet has the lowest accuracy, and it performs merely better than random guessing. The pre-trained BERT model substantially outperforms SentiWordNet. The positive effect of propagating the sentiment information from highly confident instances to the neighboring low confident instances is clearly seen on the third line, i.e., the proposed model without initial sort-

Table 6.2: Accuracy results of the proposed unsupervised learning model for each dataset.

Method	Accuracy				
	SST	MR	S140	IMF	WHO
SentiWordNet	0.6308	0.5902	0.5707	0.6368	0.5710
Pre-trained BERT model	0.9946	0.8914	0.7093	0.7845	0.7771
Proposed model without initial sorting	0.8707	0.8321	0.7239	0.8293	0.8257
Proposed model	0.8851	0.7916	0.7297	0.8346	0.8393

Table 6.3: Comparison of the proposed unsupervised learning model with the state-of-the-art methods for the S140, IMF, and WHO datasets.

Method	S140	IMF	WHO
Htait and Azzopardi [139]	0.7146	0.8088	0.7253
Gupta et al. [142]	0.6818	0.8023	0.8120
Proposed model	0.7297	0.8346	0.8393

ing. However, it is outperformed by our proposed model, which sorts and processes the unlabeled sentences in descending order according to their confidence scores. Our proposed model achieves the highest accuracy for the three datasets in discussion, i.e., S140, IMF, and WHO. For SST and MR, the highest accuracy is obtained with the pre-trained BERT model. As discussed before, this outcome is expected as the BERT model was fine-tuned with data from this domain.

Additionally, we compare the accuracy results of S140, IMF, and WHO datasets with state-of-the-art unsupervised sentiment analysis methods in Table 6.3. The first method [139] is an unsupervised sentiment analysis framework that leverages existing lexicon and word embedding models. The second one [142] is an unsupervised sentiment analysis method that employs self-training after labeling the data with an off-the-shelf pre-trained language model. Our proposed model outperforms the other methods in all three datasets.

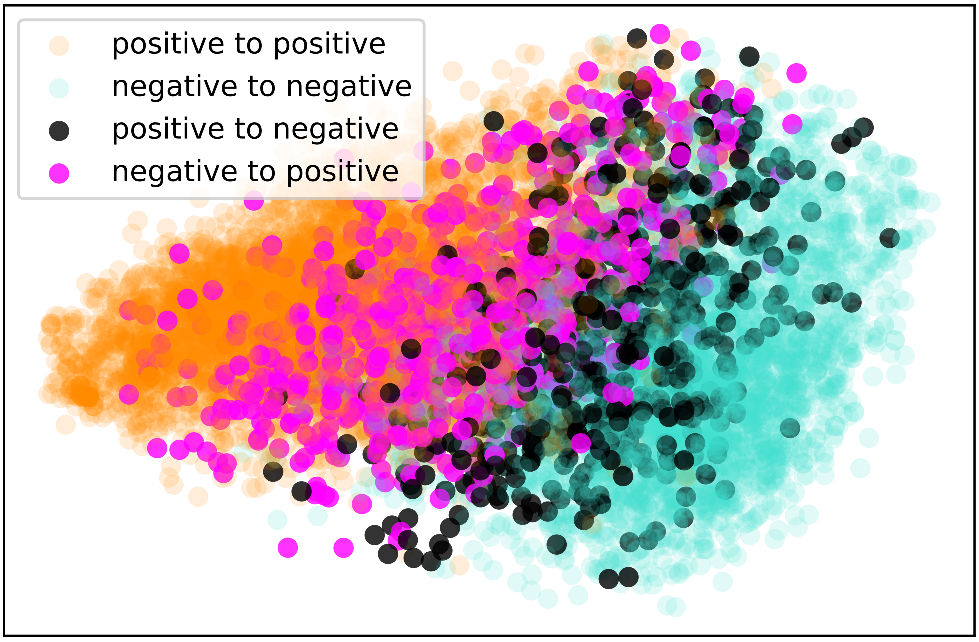


Figure 6.4: PCA plot depicting the changes from the pre-trained BERT model's initial prediction to the final prediction on the IMF dataset.

Finally, in Figure 6.4, we utilize PCA again to plot all the sentences of the IMF dataset. In this figure, the colors depict the change in the predicted labels for the sentences. A transparent orange and turquoise color are used for the instances with unchanged positive (positive to positive) and negative (negative to negative) predictions, respectively. If the prediction changes with our proposed model, the color becomes less transparent. The change from positive to negative is presented with a black color, and the change from negative to positive is depicted with a magenta color. This figure is especially useful to see how the proposed model changes the false predictions deep in the positive and negative regions into the correct labels, e.g., the magenta-colored instances on the left or the black-colored instances on the right. These changes provide an improvement in the overall classification performance.

CHAPTER 7

FEATURE ENSEMBLE MODEL FOR SENTIMENT ANALYSIS

One of the biggest challenges in sentiment analysis is context change. Many deep learning studies overcome this issue through their multi-layered learning process. However, they generally require a high amount of labeled data to train. Although there exist pre-trained language models as a remedy, it is still essential to find a model whose domain aligns with the target domain. Therefore, we explore a domain-independent method for the sentiment analysis task. In this chapter, we present a feature ensemble model that leverages context and sentiment information extracted from the data. To this end, we build a graph-based representation of the data to exploit contextual information. Additionally, we use off-the-shelf language models to support our model in terms of sentiment information. We present empirical results that illustrate the effectiveness of the proposed model in different domains.

7.1 Related Work

The natural language processing field has undergone revolutionary changes in recent years [150]. Especially sentiment analysis is a rapidly growing research field due to its wide range of application areas. Although it has been broadly studied in the literature, novel methodologies continue to emerge as sentiment analysis shows its effectiveness in new fields.

One of the biggest challenges in sentiment analysis is the context changes [151]. To mitigate the drawbacks of non-contextual word embeddings, Deniz et al. [138] proposed a refined word embedding model for sentiment analysis. Their model included context by updating the word vectors according to their positions in the sentence. On top of this model, they included valence information in the vectors with the help of a lexicon-based sentiment analysis tool. Similarly, Dashtipour et al. [152] proposed a context-aware sentiment analysis framework for multimodal datasets having textual, audio, and visual features. Other than these, researchers have mostly focused on deep learning architectures recently [47, 48, 50] as they provide contextual information intrinsically via their gradual learning process [53]. On the other hand, they require a huge amount of data to train the models [150].

Researchers utilize various techniques to disambiguate the polarity of the words in different contexts, such as co-occurrence patterns of the words or graph representations. Daudert [153] approached the sentiment analysis task as a time series problem. They generated a graph representation of the texts along with their timestamps. They stated that BERT's performance was enhanced when concatenated with the proposed graph representation followed by fully connected layers. Castillo et al. [144] presented a frequency of co-occurrence vector to construct a sentiment classification model. Their model first generated a co-occurrence graph where the nodes represented the words and edges represented the co-occurrence counts of the words within a predefined window. Before converting the graph into vectors, they applied data reduction by identifying the most valuable nodes using some common closeness measures. Devi Sri Nandhini and Pradeep [154] proposed an algorithm that detected implicit aspects for the aspect-based sentiment analysis tasks. First, they set the adverb and adjectives as sentiment words and nouns as explicit aspects. Then, they built a co-occurrence matrix for these sentiment and aspect words. Finally, they identified the hidden aspects by using the frequency of the sentiment words. Pinto et al. [155] presented a feature selection approach that extracted valuable features/words via a graph that consisted of the relation between the words and their corresponding part-of-speech tags.

Ensemble models have been widely utilized in the literature. Studies apply ensemble techniques to different layers of the process, such as feature extraction, preprocessing, or classification. Ensemble learning combines several models using aggregation methods such as averaging, majority voting, or concatenation in order to obtain better generalization performance.

Ensemble techniques have also been shown to be effective in sentiment analysis tasks. Some studies propose feature ensembles that apply ensemble techniques to feature extraction methods. Ghosh and Sanyal [156] combined feature subsets extracted by three feature selection methods: information gain, chi-square, and Gini index. Similarly, Al-Twairesh and Al-Negheimish [157] proposed a feature ensemble model that incorporated manually selected features and word embeddings, which they named surface and deep features, respectively. Phan et al. [158] presented a feature ensemble model that concatenated various feature vectors extracted utilizing different aspects of texts such as part-of-speech tags, negation words, word positions, and sentiment polarity of words. Onan [159] analyzed feature ensemble models that utilized various combinations of psycholinguistic features, categorized as linguistic processes, psychological processes, personal concerns, spoken categories, and punctuation.

In addition to feature ensembles, classifier ensembles are known to enhance classification performance as they combine the predictive powers of various classifiers [83]. Fouad et al. [160] proposed a method that applied majority voting on the decisions of three classifiers: SVM, Naive Bayes, and LR. Even though majority voting provides favorable performance, Saleena et al. [161] presented the superiority of weighted majority voting over majority voting using four classifiers: Naive Bayes, Random Forest, SVM, and LR.

Some studies apply ensemble techniques in multiple layers. Görmez et al. [162] concatenated the features extracted by different methods: TF-IDF, Continuous Bag of Words, and Skip-gram. Then, they fed these features to SVM along with predictions of two classifiers: LR and Multi-layer Perceptron. Similarly, Araque et al. [163] proposed ensembles of classifiers and features for the sentiment analysis task.

7.2 Model

Context information has a critical importance for sentiment analysis [130] as a word may have different meanings along with different sentiments in different contexts, e.g. an increase in inflation vs an increase in the gross domestic product (mostly known as GDP). In this work, we built a feature ensemble model that leverages context and

Algorithm 5: Algorithm of the proposed feature ensemble model.

```
Input: the sentences as separate instances: sentences,
sentiment labels of sentences: labels,
co-occurrence range: context_range
Output: feature vector: vector

// feature extraction & selection
cleaned_data ← CleanData(sentences);  // Alg. 1
bow_embeddings ← ExtractFeatures(cleaned_data);
filtered_bow_embedding ←
  SelectFeatures(bow_embeddings, labels);  // Alg. 6

// graph-based representation
graph ← BuildGraph(filtered_bow_embedding, labels, context_range);
n2v_word_embeddings ← Node2Vec(graph);
n2v_sentence_embeddings ←
  VectorizeSentences(sentences, n2v_word_embeddings);  // Alg. 7

// pre-trained language models
bert_sentiment_predicted_label, bert_prediction_confidence ←
  BERT(sentences);

// lexicon-based sentiment analysis
vader_sentiment_predicted_label ← VADER(sentences);

// proposed ensemble model
vector ← filtered_bow_embedding ++ n2v_sentence_embeddings;
vector ← vector ++ bert_sentiment_predicted_label;
vector ← vector ++ bert_prediction_confidence;
vector ← vector ++ vader_sentiment_predicted_label;

// vector is ready for the machine learning techniques
```

valence information to improve classification performance.

Algorithm 5 presents our proposed model. Fundamentally, the model creates an ensemble by combining various feature types. This process involves several steps as

Algorithm 6: Algorithm of the feature selection process in the proposed feature ensemble model.

```
Input: Bag-of-Words embeddings of sentences: bow_embeddings,
Sentiment labels of sentences: labels
Output: Bag-of-Words embeddings of selected features:
        filtered_bow_embeddings

Function SelectFeatures(bow_embeddings, labels):
    $V \leftarrow$ CalculateInformationGain(bow_embeddings, labels);
    $D \leftarrow length(V)$;
    $threshold \leftarrow$ ThirdQuartile(V);
    $selected\_features \leftarrow$ InformationGainFiltering(V, threshold); // Alg. 2
    $filtered\_bow\_embeddings \leftarrow$ [ ];
    foreach embedding in bow_embeddings do
        $filtered\_embedding \leftarrow$ [ ];
        for i=1,...,$D$ do
            if i in selected_features then
                $filtered\_embedding \leftarrow filtered\_embedding +\!\!+$
                  $embedding[i]$;
        $filtered\_bow\_embeddings \leftarrow$
          $filtered\_bow\_embeddings \cup filtered\_embedding$;
    return $filtered\_bow\_embeddings$
```

follows. First, we generate vector representations of the sentences using BoW after we clean them with common preprocessing steps (see Algorithm 1). With this representation, every unique word becomes a feature. To eliminate uninformative features, we employ a filter-based feature selection technique (see Algorithm 6). Simply, we calculate the information gain values of all features and select the third quartile value as the threshold. Then, we remove all the features whose IG values are lower than this threshold. The filtered vectors constitute the first part of our feature pool. We also utilize these vectors in the second step of our algorithm, where we involve the context information of the words. At this point, we build a network from the features by calculating their co-occurrence relation within a specified window, also called their context range, as the contexts of words are built around their surrounding words [62]. In this graph, nodes represent words, and edges represent the sentiment intensity between the two words. If two words appear together within the context range in a sentence, their relation increases or decreases by one unit regarding the sentence's sentiment, positive or negative, respectively.

To provide a clear understanding of this procedure, we present the graph of a sample dataset in Figure 7.1. Our sample dataset consists of the following four sentences:

⋆ The reduce in the budgetary wage bill increased unemployment. (Negative)

⋆ Consolidating into one basic wage also increased transparency. (Positive)

⋆ Inflation has increased to more than 50 percent. (Negative)

⋆ Reforms to increased fiscal transparency are welcome. (Positive)

To keep the graph simple, we set the context range as two. The nodes in the graph represent the words in the sentences after the data-cleaning process. The edges between the nodes carry the sentiment-oriented relation. The sign of the edge weights represents the polarity of the sentiment, while the magnitude represents how strong the relationship is. For example, the edge weight between the nodes *increased* and *transparency* indicates a strong relationship towards the positive sentiment. Other than that, the relationship between the nodes *increased* and *wage* is noteworthy with its weight of zero. The reason for this is that there exist two sentences with contradicting sentiments in our sample dataset that have both of the words within their context range. Accordingly, the relation between these nodes does not carry conclusive sentiment information for this domain. As seen from the sample figure, we obtain a domain-specific representation that contains sentiment-oriented relationship information.

We aim to obtain encoded relationships of the words by projecting this graph into

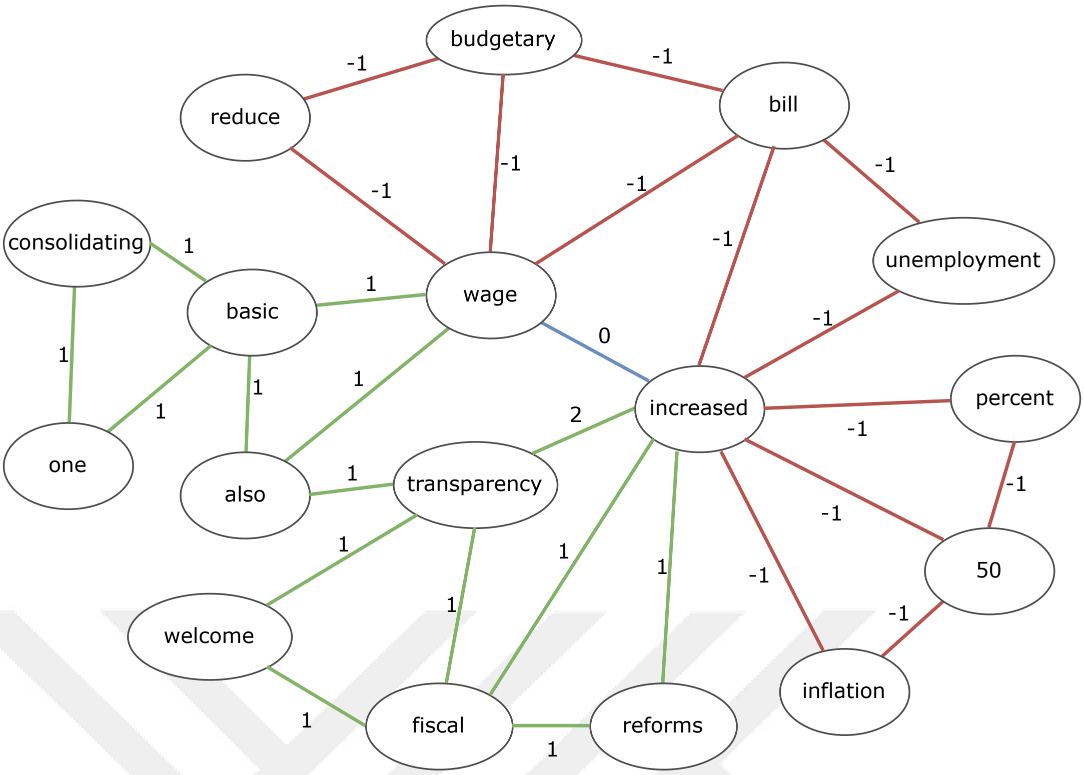


Figure 7.1: Sample graph-based representation of the proposed feature ensemble model.

low-dimensional space with Node2Vec. Node2Vec is a recent technique for embedding graph-like data into machine learning models [164]. It takes a graph as input, analyzes the existences and weights of edges between the vertices in the graph, and produces a fixed-length vector for each vertex of the graph as its output. Finally, we compute the sentence vector by averaging the vectors of all words in the sentence (see Algorithm 7).

After incorporating the context information, we also include valence information in our model. For this purpose, we first execute a pre-trained BERT model [165] on our original sentences. Then, we convert the labels and confidence scores provided by the model into features. In addition, we include the VADER score in our feature pool in order to enrich our feature ensemble model in terms of valence information. The addition of this score concludes the external computation required for the feature pool. As the final step, we combine the vectors in the feature pool by concatenating them and provide this combined vector as input for the machine learning techniques.

Algorithm 7: Algorithm of the vectorization process in the proposed feature ensemble model.

```
Input: the sentences as separate instances: sentences,
Node2Vec word embeddings: n2v_word_embeddings
Output: sentence vectors: sentence_embeddings

Function VectorizeSentences(sentences, n2v_word_embeddings):
    // vector size is equal to the length
    // of each vector in n2v_word_embeddings
    $x_{empty} \leftarrow \langle 0, 0, \ldots, 0 \rangle$;
    $i \leftarrow 0$;

    foreach sentence in sentences do
        $i \leftarrow i + 1$;
        $counter \leftarrow 0$;
        $x^i \leftarrow x_{empty}$
        foreach token in sentence do
            if token in n2v_word_embeddings then
                $x^i \leftarrow x^i + n2v\_word\_embeddings[token]$;
                $counter \leftarrow counter + 1$;

        if counter > 0 then
            $x^i \leftarrow x^i / counter$;
        $sentence\_embeddings[i] \leftarrow x^i$

    return $sentence\_embeddings$
```

The two main components contributing to the time complexity of this algorithm are Node2Vec learning time and BERT's inference time. The dataset size and the selected context range are other factors in the equation.

Our model can also be applied on multi-class datasets (e.g. positive, negative, and neutral). For this purpose, a pre-trained BERT model providing multi-label predictions should be employed. Similarly, VADER's compound score should be adjusted according to the label classes. Finally, building the graph may need a rework as the weights of the edges depend on the labels of the sentences.

7.3 Experiments

7.3.1 Settings

For the IMF, MR, and WHO datasets, we applied a 5-fold cross-validation technique in our experiments as there was no specification for the train and test sets in the original data. For the other ones, we used the train and test sets provided by the original data in our experiments.

For Node2Vec, we set the *walk_length* to *30*, *num_walks* to *200*, *workers* to *4*, *window* to *10*, *min_count* to *1*, and *batch_words* to *4*.

We set the threshold for IG as the third quartile value.

As the pre-trained language model, we used the pre-trained *siebert/sentiment-roberta-large-english* BERT model with the *sentiment-analysis* task.

7.3.2 Results and Discussion

Tables 7.1, 7.2, 7.3, 7.4, and 7.5 present the experiment results for all datasets; SST, MR, S140, IMF, and WHO, respectively. All the tables involve two sub-tables. The first one gives the ablation study results, i.e., the performance results after including each feature type of our proposed model one by one for varying feature vector sizes (64 and 128) and machine learning techniques (LR and SVM). The included feature types are as follows. We begin with the Bag-of-Words results as the baseline method (BoW). Then, we present the results after applying feature selection with information gain (IG). The third row (Node2Vec) is for the feature vectors obtained from our context-aware graph-based representation utilizing Node2Vec. The following row gives the accuracy results after concatenating the feature vectors of IG and Node2Vec. In the fifth row, we include the BERT features into our feature pool. Finally, in the last row, we enhance the pool with the VADER features. The second sub-table shares the comparison results with off-the-shelf methods employed in our proposed model;

Table 7.1: Experiment results of the proposed feature ensemble model for the SST dataset.

(a) Ablation study results.

Method	Node2Vec vector size			
	64		128	
	LR	SVM	LR	SVM
BoW	0.8139	0.8184	0.8139	0.8184
IG	0.8609	0.8667	0.8609	0.8667
Node2Vec	0.8667	0.8674	0.8738	0.8738
IG + Node2Vec	0.8809	0.8815	0.8835	0.8822
IG + Node2Vec + BERT	0.9569	0.9549	0.9562	0.9562
IG + Node2Vec + BERT + VADER (Proposed model)	0.9607	0.9588	0.9601	0.9562

(b) Comparison with off-the-shelf methods employed in the model.

Method	Accuracy
BERT	0.9498
VADER	0.5892
Proposed model	0.9607

(c) Comparison with state-of-the-art studies.

Method	Accuracy
Biesialska et al. (2021) [166]	0.9140
Xiang et al. (2021) [167]	0.9470
Zhang et al. (2021) [168]	0.9400
Proposed model	0.9607

BERT and VADER. In addition to these, Tables 7.1, 7.2, and 7.3 consist of a third sub-table that presents the comparison results of the respective dataset with state-of-the-art studies since SST, MR, and S140 are well-known and widely used datasets for the sentiment analysis task.

In Table 7.1, the maximum accuracy, 96.1%, is achieved by our proposed model with the Node2Vec feature vector size of 64 and the LR classifier. It is clear from Table 7.1a that the accuracy consistently increases as we introduce new features into our feature pool, regardless of the selected Node2Vec feature vector size or classifier. Table 7.1b shows that our proposed model achieves higher accuracy than off-the-shelf methods. Finally, in Table 7.1c, we compare our proposed model with the follow-

Table 7.2: Experiment results of the proposed feature ensemble model for the MR dataset.

(a) Ablation study results.

Method	Node2Vec vector size			
	64		128	
	LR	SVM	LR	SVM
BoW	0.7536	0.7555	0.7536	0.7555
IG	0.8165	0.8180	0.8165	0.8180
Node2Vec	0.8100	0.8092	0.8145	0.8135
IG + Node2Vec	0.8373	0.8386	0.8369	0.8379
IG + Node2Vec + BERT	0.9109	0.9114	0.9109	0.9115
IG + Node2Vec + BERT + VADER (Proposed model)	0.9112	0.9113	0.9112	0.9115

(b) Comparison with off-the-shelf methods employed in the model.

Method	Accuracy
BERT	0.9109
VADER	0.5403
Proposed model	0.9115

(c) Comparison with state-of-the-art studies.

Method	Accuracy
Cheng et al. (2021) [88]	0.8530
Perikos et al. (2021) [87]	0.8051
Xiang et al. (2021) [167]	0.9070
Proposed model	0.9115

ing state-of-the-art studies. Biesialska et al. [166] proposed a multilingual sentiment classifier that uses a self-attention neural network model and contextual embeddings. Xiang et al. [167] presented a data augmentation method that takes advantage of part-of-speech tags to identify lexical substitution points. Zhang et al. [168] introduced a transformers-based neural network model that leverages feature-based and fine-tuning methods. The results in Table 7.1c show that our model outperforms these studies.

The ablation study results in Tables 7.2a, 7.3a, 7.4a, and 7.5a are in line with the results in Table 7.1a, i.e., extending the feature pool with new feature vectors consistently increases the classification performance. There are two exceptions to this

Table 7.3: Experiment results of the proposed feature ensemble model for the S140 dataset.

(a) Ablation study results.

Method	Node2Vec vector size			
	64		128	
	LR	SVM	LR	SVM
BoW	0.7660	0.7855	0.7660	0.7855
IG	0.8050	0.8078	0.8050	0.8078
Node2Vec	0.8301	0.8301	0.8357	0.8384
IG + Node2Vec	0.8357	0.8329	0.8301	0.8217
IG + Node2Vec + BERT	0.8830	0.8886	0.8942	0.8942
IG + Node2Vec + BERT + VADER (Proposed model)	0.8914	0.8886	0.8914	0.8914

(b) Comparison with off-the-shelf methods employed in the model.

Method	Accuracy
BERT	0.8858
VADER	0.6908
Proposed model	0.8914

(c) Comparison with state-of-the-art studies.

Method	Accuracy
Al-deen et al. (2021) [169]	0.8217
Basiri et al. (2021) [90]	0.8182
Kamyab et al. (2021) [91]	0.8712
Proposed model	0.8914

phenomenon. The first one is in Table 7.2a, where the Node2Vec vector size is 64, and the classifier is SVM; and the second one is in Table 7.3a, where the Node2Vec vector size is 128. Adding VADER features to the pool of IG, Node2Vec, and BERT features slightly decreases the accuracy by 0.01% and 0.28%, respectively, yet the proposed model's obtained result is still greater than all other methods in the respective setting.

Similarly, the comparison results with off-the-shelf methods employed in our study in Tables 7.2b, 7.3b, 7.4b, and 7.5b are in line with Table 7.1b, i.e., the proposed model achieves better classification performances than the existing methods. More-

Table 7.4: Experiment results of the proposed feature ensemble model for the IMF dataset.

(a) Ablation study results.

Method	Node2Vec vector size			
	64		128	
	LR	SVM	LR	SVM
BoW	0.8212	0.8212	0.8212	0.8212
IG	0.8331	0.8345	0.8331	0.8345
Node2Vec	0.8193	0.8188	0.8241	0.8232
IG + Node2Vec	0.8378	0.8391	0.8388	0.8393
IG + Node2Vec + BERT	0.8687	0.8690	0.8696	0.8698
IG + Node2Vec + BERT + VADER (Proposed model)	0.8696	0.8704	0.8704	0.8714

(b) Comparison with off-the-shelf methods employed in the model.

Method	Accuracy
BERT	0.8131
VADER	0.6545
Proposed model	0.8714

over, the performance increases in Tables 7.4b and 7.5b are more notable than those in Tables 7.1b, 7.2b, and 7.3b. We believe that the reason for this is the domain of the datasets. Since the SST, MR, and S140 datasets are widely known datasets, they or their domains were used in the training of BERT vectors. Therefore, the pre-trained BERT vectors already achieve high performance in those domains. Nonetheless, our proposed algorithm improves the classification performance. On the other hand, the pre-trained BERT vectors may have never seen the economy (IMF) or health (WHO) domains when training. Hence, the improvement provided by our proposed model is more visible in comparison.

Finally, similar to Table 7.1c, our model outperforms all the state-of-the-art studies listed in Tables 7.2c and 7.3c. We briefly mention these studies as follows. Cheng

Table 7.5: Experiment results of the proposed feature ensemble model for the WHO dataset.

(a) Ablation study results.

Method	Node2Vec vector size			
	64		128	
	LR	SVM	LR	SVM
BoW	0.8641	0.8626	0.8641	0.8626
IG	0.8720	0.8721	0.8720	0.8721
Node2Vec	0.8702	0.8687	0.8728	0.8717
IG + Node2Vec	0.8811	0.8832	0.8820	0.8834
IG + Node2Vec + BERT	0.8885	0.8894	0.8885	0.8903
IG + Node2Vec + BERT + VADER (Proposed model)	0.8887	0.8902	0.8892	0.8907

(b) Comparison with off-the-shelf methods employed in the model.

Method	Accuracy
BERT	0.8293
VADER	0.5722
Proposed model	0.8907

et al. [88] and Al-deen et al. [169] proposed a deep learning architecture that uses a multi-head attention mechanism. Perikos et al. [87] introduced Hidden Markov models for sentiment analysis tasks. Xiang et al. [167] presented a data augmentation method that takes advantage of part-of-speech tags to identify lexical substitution points. Basiri et al. [90] and Kamyab et al. [91] proposed attention-based deep learning models. The former utilized Convolutional Neural Networks and Bidirectional Long Short-Term Memory in their model, while the latter leveraged Convolutional Neural Networks and Recurrent Neural Networks.

We held additional experiments to examine the effects of parameter selection in our proposed algorithm. Figure 7.2 provides the classification performance results on the MR dataset with varying context ranges. It can be seen from the figure that the classi-

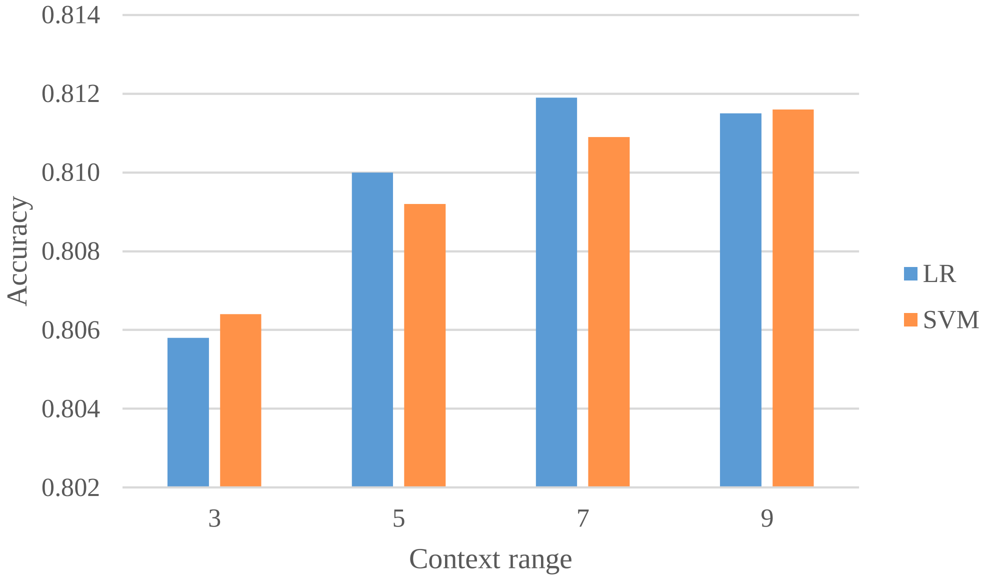


Figure 7.2: Effects of context range on the classification performance in the proposed feature ensemble model for the MR dataset.

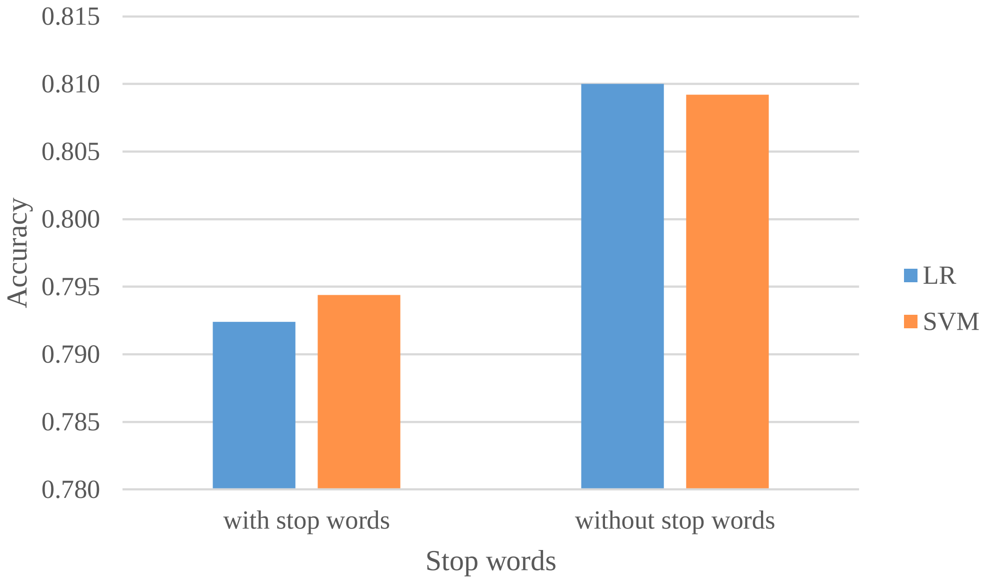


Figure 7.3: Effects of stop words on the classification performance in the proposed feature ensemble model for the MR dataset.

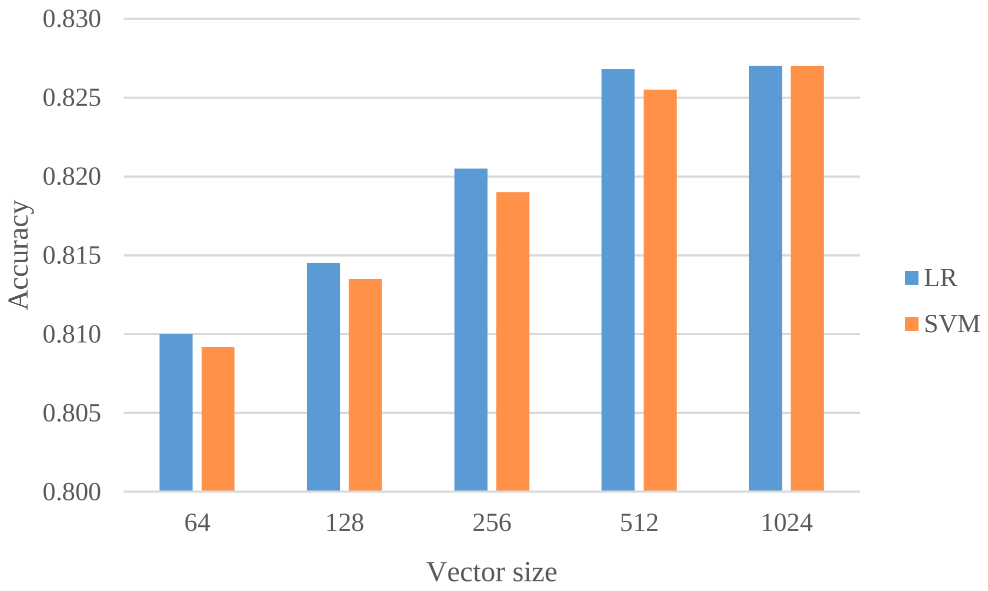


Figure 7.4: Effects of vector size on the classification performance in the proposed feature ensemble model for the MR dataset.

fication performance increases as the context ranges increase for both LR and SVM. However, the amount of improvement decreases as the context range gets larger: the difference between 5 and 7 is more significant than between 7 and 9. We can infer that it is essential to choose the context range wisely, as small ones may not capture the context, and big ones could cause overlapped contexts.

The question of removing or not removing stop words in the preprocessing step is common in NLP studies [170, 171]. Therefore, we included this analysis for our model in Figure 7.3. In this analysis, the results of "without stop words" are collected from our proposed model. For the "with stop words" part, we updated our data cleaning code and discarded the stop word removal part, and kept everything else the same. The obtained results show that the removal of the stop words has a positive effect on sentiment classification.

Finally, we analyzed the effect of changing the vector size of Node2Vec features in Figure 7.4. We began from 64 and doubled the vector size up to 1024 for this analysis. According to the figure, the classification performance increases as the vector size increases for both LR and SVM classifiers. We believe that a larger vector size

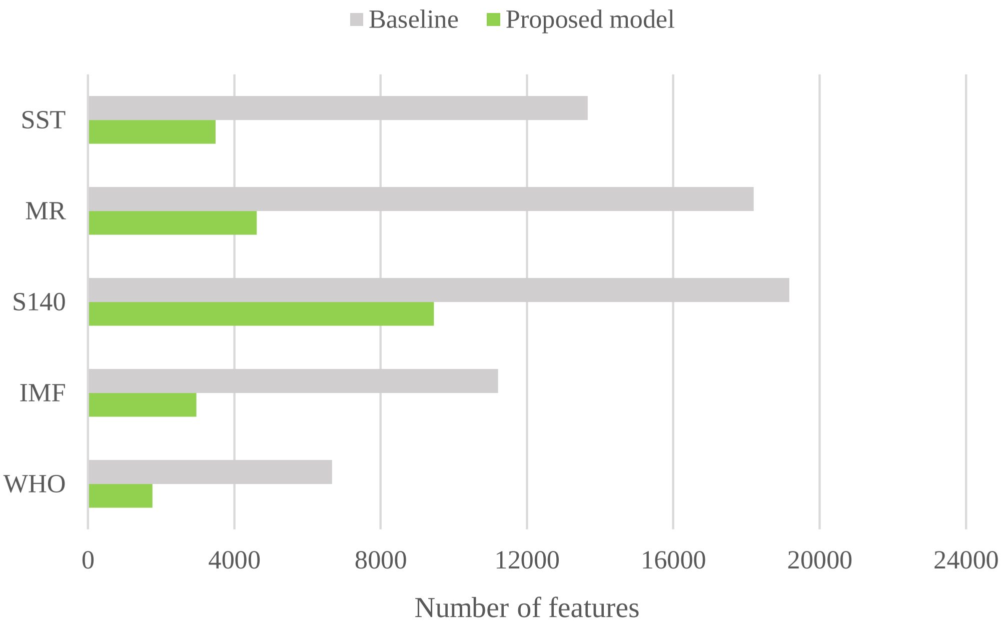


Figure 7.5: Improvements in the number of features for all datasets after the proposed feature ensemble model is applied.

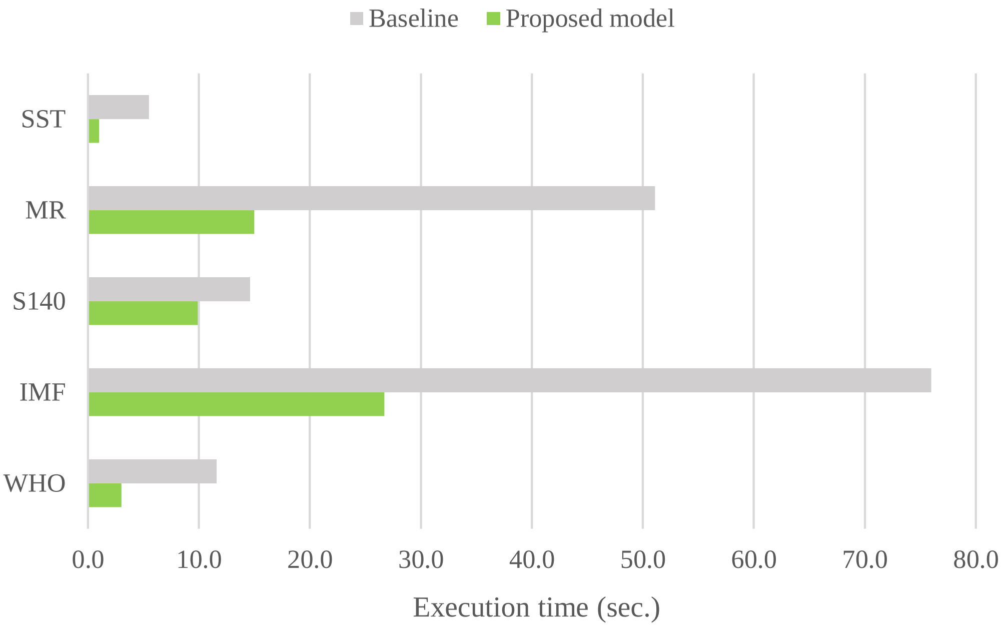


Figure 7.6: Improvements in the execution time for all datasets after the proposed feature ensemble model is applied.

captures the relationship within the graph more accurately; hence, the performance increases.

As a final analysis, we present the improvements in the number of features and execution time after applying the proposed model in Figures 7.5 and 7.6, respectively. In this experiment, we collected the results under the following settings: the context range is set to 5, the vector size is set to 64, and the classifier is set to LR. It is clear from the figures that the proposed model requires less execution time than the baseline (BoW) for all datasets as it tremendously decreases the number of features.

CHAPTER 8

CONCLUSION

Recent developments in data acquisition and storage technologies along with machine learning techniques have enabled NLP to make great progress. Sentiment analysis is one of the NLP tasks that lead decision-making processes. Its applicability to a wide range of areas has made it even more popular.

We identify the challenges of sentiment analysis research as follows. The first challenge is the data in the NLP domain is generally huge and is often bloated with out-of-context information. The irrelevant or redundant data makes it harder to build a model that correctly identifies the underlying sentiment. A second challenge is the generalization problem. It is possible to fine-tune a model for a specific domain. Although transferring knowledge into another domain is possible, it is often cumbersome. Moreover, the lack of sufficient labeled data in particular domains makes the process almost impossible.

Our attempts to overcome these challenges are four-fold. In all four attempts, we held extensive experiments with three well-known benchmark datasets and two real-world datasets we have formed. The summary of our studies is as follows.

First, we proposed a hybrid multiobjective feature selection algorithm to improve the performance of the sentiment classification task in various domains. Our model combines a filter-based and a wrapper-based approach. Experiment results showed that our proposed model significantly improved learning performance. It increased the accuracy by up to 15% and decreased the number of features by up to 79% over baseline sentiment classification models, which eventually reduced computation time and space. We presented the progression of our algorithm using both textual and visual representations of the results in a multiobjective fashion, including both accuracy and feature size. Moreover, we verified the effectiveness of our model by comparing our results with off-the-shelf feature selection techniques. The results showed that the proposed model is promising to improve sentiment classification performance in datasets of different domains in terms of accuracy and computation costs by selecting the most informative features.

Second, we proposed a model to enhance the effectiveness of available pre-trained word embeddings used for the sentiment analysis task. The proposed model consists of refined word embeddings with context-based information and lexicon-based sentiment scores. The context information is obtained from the neighbors of each word while we leverage VADER for the sentiment score. Experiments were carried out using two off-the-shelf word embeddings, i.e., GloVe and Word2Vec. The results showed that the proposed model improves the performance of the sentiment classification task regardless of the word embeddings or machine learning techniques. Nearly 10% increase in the prediction accuracy indicates that integrating context and sentiment knowledge into the word embeddings has merit in enhancing sentiment classification performance.

Third, we proposed a model that uses the information within pre-trained BERT models to boost the sentiment classification performance on unlabeled datasets. For this purpose, we determine the instances where the pre-trained BERT model is highly confident about its prediction. Then, we propagate this information to the instances where the model is less confident. The propagation strategy is based on sentence similarity, and we utilize two metrics to determine the similarity between sentences. Our experiment results showed that the proposed model improves the classification performance of the pre-trained BERT model by up to 7% for the datasets from unseen domains. Moreover, comparison results with the state-of-the-art models verify the model's efficiency.

Finally, we proposed a feature ensemble model for the sentiment analysis task. The model essentially builds a context- and sentiment-aware feature pool representing the data. To include context awareness, we generate a graph-based representation of the data and convert the graph into fixed-length vectors. For sentiment awareness, we take advantage of the existing language models. We held extensive experiments on different datasets to verify that our model improves the classification performance independently from the dataset domain. The results showed that our model boosts the classification performance regardless of the domain compared to the traditional and state-of-the-art methods. Moreover, it remarkably reduced the number of features, leading to less execution time. In fact, the feature ensemble model provides the maximum performance improvement among all models, by up to 16% increment in accuracy. The main reason for this is it benefits from the power of diversity.

To sum up, we approached the sentiment analysis task as a binary classification problem and developed various models to improve its performance. In future work, we plan to analyze multi-class classification performance including the neutral-labeled sentences in the model. Moreover, we intend to enhance our feature ensemble model by adding a high amount of various context and sentiment-related features.

REFERENCES

[1] X. Chi, T. P. Siew, and E. Cambria, "Adaptive Two-Stage Feature Selection for Sentiment Classification," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 1238–1243.

[2] J. A. Morente-Molinera, G. Kou, K. Samuylov, R. Ureña, and E. Herrera-Viedma, "Carrying Out Consensual Group Decision Making Processes Under Social Networks Using Sentiment Analysis Over Comparative Expressions," *Knowledge-Based Systems*, vol. 165, pp. 335–345, 2019.

[3] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A Survey on Sentiment Analysis Methods, Applications, and Challenges," *Artificial Intelligence Review*, pp. 1–50, 2022.

[4] B. Liu, "Sentiment Analysis and Opinion Mining," *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[5] H. Zhuang, F. Guo, C. Zhang, L. Liu, and J. Han, "Joint Aspect-Sentiment Analysis with Minimal User Guidance," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1241–1250.

[6] K. Ravi and V. Ravi, "A Survey on Opinion Mining and Sentiment Analysis: Tasks, Approaches and Applications," *Knowledge-Based Systems*, vol. 89, pp. 14–46, 2015.

[7] E. Georgiadou, S. Angelopoulos, and H. Drake, "Big Data Analytics and International Negotiations: Sentiment Analysis of Brexit Negotiating Outcomes," *International Journal of Information Management*, vol. 51, p. 102048, 2020.

[8] S. Hu, A. Kumar, F. Al-Turjman, S. Gupta, S. Seth, and Shubham, "Reviewer Credibility and Sentiment Analysis Based User Profile Modelling for Online Product Recommendation," *IEEE Access*, vol. 8, pp. 26 172–26 189, 2020.

[9] R. L. Rosa, G. M. Schwartz, W. V. Ruggiero, and D. Z. Rodríguez, "A Knowledge-Based Recommendation System That Includes Sentiment Analysis and Deep Learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2124–2135, 2018.

[10] C. Ng, K. M. Law, and A. W. Ip, "Assessing Public Opinions of Products Through Sentiment Analysis: Product Satisfaction Assessment by Sentiment Analysis," *Journal of Organizational and End User Computing (JOEUC)*, vol. 33, no. 4, pp. 125–141, 2021.

[11] X. Xu, "What are Customers Commenting On, and How is Their Satisfaction Affected? Examining Online Reviews in the On-demand Food Service Context," *Decision Support Systems*, p. 113467, 2020.

[12] M. Masarifoglu, U. Tigrak, S. Hakyemez, G. Gul, E. Bozan, A. H. Buyuklu, and A. Özgür, "Sentiment Analysis of Customer Comments in Banking Using BERT-Based Approaches," in *2021 29th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2021, pp. 1–4.

[13] Y. Liu, X. Huang, A. An, and X. Yu, "ARSA: A Sentiment-aware Model for Predicting Sales Performance Using Blogs," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 607–614.

[14] S. Çalı and Ş. Y. Balaman, "Improved Decisions for Marketing, Supply and Purchasing: Mining Big Data Through an Integration of Sentiment Analysis and Intuitionistic Fuzzy Multi Criteria Assessment," *Computers & Industrial Engineering*, vol. 129, pp. 315–332, 2019.

[15] T. B. Mirani and S. Sasi, "Sentiment Analysis of ISIS Related Tweets Using Absolute Location," in *2016 International Conference on Computational Science and Computational Intelligence*. IEEE, 2016, pp. 1140–1145.

[16] G. A. Ruz, P. A. Henríquez, and A. Mascareño, "Sentiment Analysis of Twitter Data during Critical Events through Bayesian Networks Classifiers," *Future Generation Computer Systems*, vol. 106, pp. 92 – 104, 2020.

[17] D. Won, Z. C. Steinert-Threlkeld, and J. Joo, "Protest Activity Detection and Perceived Violence Estimation from Social Media Images," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 786–794.

[18] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, "TwitInfo: Aggregating and Visualizing Microblogs for Event Exploration," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2011, pp. 227–236.

[19] N. Singh, N. Roy, and A. Gangopadhyay, "Analyzing the Sentiment of Crowd for Improving the Emergency Response Services," in *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2018, pp. 1–8.

[20] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A System for Real-time Twitter Sentiment Analysis of 2012 US Presidential Election Cycle," in *Proceedings of the ACL 2012 system demonstrations*, 2012, pp. 115–120.

[21] W. Budiharto and M. Meiliana, "Prediction and Analysis of Indonesia Presidential Election from Twitter Using Sentiment Analysis," *Journal of Big data*, vol. 5, no. 1, pp. 1–10, 2018.

[22] S. M. Yimam, H. M. Alemayehu, A. Ayele, and C. Biemann, "Exploring Amharic Sentiment Analysis from Social Media Texts: Building Annotation Tools and Classification Models," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 1048–1060.

[23] A. Ceron, L. Curini, and S. M. Iacus, "Using Sentiment Analysis to Monitor Electoral Campaigns: Method Matters — Evidence from the United States and Italy," *Social Science Computer Review*, vol. 33, no. 1, pp. 3–20, 2015.

[24] K. Mishev, A. Gjorgjevikj, I. Vodenska, L. T. Chitkushev, and D. Trajanov, "Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers," *IEEE access*, vol. 8, pp. 131 662–131 682, 2020.

[25] T. Renault, "Sentiment Analysis and Machine Learning in Finance: A Comparison of Methods and Models on One Million Messages," *Digital Finance*, vol. 2, pp. 1 – 13, 2020.

[26] F. Xing, L. Malandri, Y. Zhang, and E. Cambria, “Financial Sentiment Analysis: An Investigation into Common Mistakes and Silver Bullets,” in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 978–987.

[27] A. R. Alaei, S. Becken, and B. Stantic, “Sentiment Analysis in Tourism: Capitalizing on Big Data,” *Journal of Travel Research*, vol. 58, no. 2, pp. 175–191, 2019.

[28] E. Saad, S. Din, R. Jamil, F. Rustam, A. Mehmood, I. Ashraf, and G. S. Choi, “Determining the Efficiency of Drugs Under Special Conditions from Users’ Reviews on Healthcare Web Forums,” *IEEE Access*, vol. 9, pp. 85 721–85 737, 2021.

[29] L. Abualigah, H. E. Alfar, M. Shehab, and A. M. A. Hussein, “Sentiment Analysis in Healthcare: A Brief Review,” in *Recent Advances in NLP: The Case of Arabic Language*, 2020, pp. 129–141.

[30] A. Alamoodi, B. Zaidan, A. Zaidan, O. Albahri, K. Mohammed, R. Malik, E. Almahdi, M. Chyad, Z. Tareq, A. Albahri, H. Hameed, and M. Alaa, “Sentiment Analysis and Its Applications in Fighting COVID-19 and Infectious Diseases: A Systematic Review,” *Expert Systems with Applications*, p. 114155, 2020.

[31] E. Gabarron, E. Dorronzoro, O. Rivera-Romero, and R. Wynn, “Diabetes on Twitter: A Sentiment Analysis,” *Journal of diabetes science and technology*, vol. 13, no. 3, pp. 439–444, 2019.

[32] F. Greaves, D. Ramirez-Cano, C. Millett, A. Darzi, L. Donaldson *et al.*, “Use of Sentiment Analysis for Capturing Patient Experience from Free-text Comments Posted Online,” *Journal of medical Internet research*, vol. 15, no. 11, p. e2721, 2013.

[33] L. Rognone, S. Hyde, and S. S. Zhang, “News Sentiment in the Cryptocurrency Market: An Empirical Comparison with Forex,” *International Review of Financial Analysis*, vol. 69, p. 101462, 2020.

[34] B. Pang, L. Lee *et al.*, “Opinion Mining and Sentiment Analysis,” *Foundations and Trends® in information retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[35] M. Duşcu and D. Günneç, “Polarity Classification of Twitter Messages Using Audio Processing,” *Information Processing & Management*, vol. 57, no. 6, p. 102346, 2020.

[36] R. Feldman, “Techniques and Applications for Sentiment Analysis,” *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[37] Y. Noh, S. Park, and S.-B. Park, “Aspect-Based Sentiment Analysis Using Aspect Map,” *Applied Sciences*, vol. 9, no. 16, p. 3239, 2019.

[38] P. Karagoz, B. Kama, M. Ozturk, I. H. Toroslu, and D. Canturk, “A Framework for Aspect Based Sentiment Analysis on Turkish Informal Texts,” *Journal of Intelligent Information Systems*, vol. 53, no. 3, pp. 431–451, 2019.

[39] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, “Sentiment Analysis Based on Deep Learning: A Comparative Study,” *Electronics*, vol. 9, no. 3, p. 483, 2020.

[40] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, “Lexicon-Based Methods for Sentiment Analysis,” *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[41] A. Rasool, R. Tao, M. Kamyab, and S. Hayat, “GAWA - A Feature Selection Method for Hybrid Sentiment Classification,” *IEEE Access*, vol. 8, pp. 191 850–191 861, 2020.

[42] R. Xia, C. Zong, and S. Li, “Ensemble of Feature Sets and Classification Algorithms for Sentiment Classification,” *Information sciences*, vol. 181, no. 6, pp. 1138–1152, 2011.

[43] A. Deniz, M. Angin, and P. Angin, “Evolutionary Multiobjective Feature Selection for Sentiment Analysis,” *IEEE Access*, vol. 9, pp. 142 982–142 996, 2021.

[44] S. Poria, E. Cambria, G. Winterstein, and G.-B. Huang, “Sentic Patterns:

Dependency-Based Rules for Concept-level Sentiment Analysis," *Knowledge-Based Systems*, vol. 69, pp. 45–63, 2014.

[45] M. Birjali, M. Kasri, and A. Beni-Hssane, "A Comprehensive Survey on Sentiment Analysis: Approaches, Challenges and Trends," *Knowledge-Based Systems*, vol. 226, p. 107134, 2021.

[46] O. Appel, F. Chiclana, J. Carter, and H. Fujita, "A Hybrid Approach to the Sentiment Analysis Problem at the Sentence Level," *Knowledge-Based Systems*, vol. 108, pp. 110–124, 2016, new Avenues in Knowledge Bases for Natural Language Processing.

[47] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An Attention-Based Bidirectional CNN-RNN Deep Model for Sentiment Analysis," *Future Generation Computer Systems*, vol. 115, pp. 279–294, 2021.

[48] D. Li, R. Rzepka, M. Ptaszynski, and K. Araki, "HEMOS: A Novel Deep Learning-Based Fine-Grained Humor Detecting Method for Sentiment Analysis of Social Media," *Information Processing & Management*, vol. 57, no. 6, p. 102290, 2020.

[49] M. Usama, B. Ahmad, E. Song, M. S. Hossain, M. Alrashoud, and G. Muhammad, "Attention-Based Sentiment Analysis Using Convolutional and Recurrent Neural Network," *Future Generation Computer Systems*, vol. 113, pp. 571–578, 2020.

[50] A. Chatterjee, U. Gupta, M. K. Chinnakotla, R. Srikanth, M. Galley, and P. Agrawal, "Understanding Emotions in Text Using Deep Learning and Big Data," *Computers in Human Behavior*, vol. 93, pp. 309–317, 2019.

[51] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi, "Sentiment Analysis Based on Improved Pre-trained Word Embeddings," *Expert Systems with Applications*, vol. 117, pp. 139–147, 2019.

[52] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, and E. Cambria, "Learning Word Representations for Sentiment Analysis," *Cognitive Computation*, vol. 9, no. 6, pp. 843–851, 2017.

[53] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent Trends in Deep Learning Based Natural Language Processing [Review Article],” *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[54] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[55] K. J. Madukwe, X. Gao, and B. Xue, “A GA-Based Approach to Fine-tuning BERT for Hate Speech Detection,” in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 2821–2828.

[56] N. F. Da Silva, E. R. Hruschka, and E. R. Hruschka Jr, “Tweet Sentiment Analysis with Classifier Ensembles,” *Decision Support Systems*, vol. 66, pp. 170–179, 2014.

[57] D. M. E.-D. M. Hussein, “A Survey on Sentiment Analysis Challenges,” *Journal of King Saud University-Engineering Sciences*, vol. 30, no. 4, pp. 330–338, 2018.

[58] M. A. Hall and L. A. Smith, “Practical Feature Subset Selection for Machine Learning,” in *Computer Science ’98 Proceedings of the 21st Australasian Computer Science Conference*, February 1998, pp. 181–191.

[59] M. Dash and H. Liu, “Feature Selection for Classification,” *Intelligent data analysis*, vol. 1, no. 1-4, pp. 131–156, 1997.

[60] Y. Wang, G. Huang, J. Li, H. Li, Y. Zhou, and H. Jiang, “Refined Global Word Embeddings Based on Sentiment Concept for Sentiment Analysis,” *IEEE Access*, vol. 9, pp. 37 075–37 085, 2021.

[61] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang, “Refining Word Embeddings for Sentiment Analysis,” in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 534–539.

[62] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, “Sentiment Embeddings With Applications to Sentiment Analysis,” *IEEE transactions on knowledge and data Engineering*, vol. 28, no. 2, pp. 496–509, 2015.

[63] J. Blitzer, M. Dredze, and F. Pereira, “Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification,” in *Proceedings of the 45th annual meeting of the association of computational linguistics*, 2007, pp. 440–447.

[64] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, “Sentiment Strength Detection in Short Informal Text,” *Journal of the American society for information science and technology*, vol. 61, no. 12, pp. 2544–2558, 2010.

[65] S. Baccianella, A. Esuli, and F. Sebastiani, “SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining,” in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10)*, 2010.

[66] C. Hutto and E. Gilbert, “VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text,” in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, no. 1, 2014.

[67] T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features,” in *European conference on machine learning*. Springer, 1998, pp. 137–142.

[68] E. Rudkowsky, M. Haselmayer, M. Wastian, M. Jenny, Š. Emrich, and M. Sedlmair, “More Than Bags of Words: Sentiment Analysis with Word Embeddings,” *Communication Methods and Measures*, vol. 12, no. 2-3, pp. 140–157, 2018.

[69] G. Salton and C. Buckley, “Term-weighting Approaches in Automatic Text Retrieval,” *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[70] B. Naderalvojoud and E. A. Sezer, “Sentiment Aware Word Embeddings Using Refinement and Senti-contextualized Learning Approach,” *Neurocomputing*, vol. 405, pp. 149–160, 2020.

[71] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and Their Compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[72] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[73] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep Contextualized Word Representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237.

[74] D. Yin, T. Meng, and K. Chang, “SentiBERT: A Transferable Transformer-Based Architecture for Compositional Sentiment Semantics,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 2020, pp. 3695–3706.

[75] A. Yafoz and M. Mouhoub, “Analyzing Machine Learning Algorithms for Sentiments in Arabic Text,” in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 2150–2156.

[76] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language Models are Few-Shot Learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[77] D. Araci, “FinBERT: Financial Sentiment Analysis with Pre-trained Language Models,” *arXiv preprint arXiv:1908.10063*, 2019.

[78] I. Beltagy, K. Lo, and A. Cohan, “SciBERT: A Pretrained Language Model for Scientific Text,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3615–3620.

[79] K. Huang, J. Altosaar, and R. Ranganath, "ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission," *arXiv preprint arXiv:1904.05342*, 2019.

[80] M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri, "ParsBERT: Transformer-Based Model for Persian Language Understanding," *arXiv preprint arXiv:2005.12515*, 2020.

[81] H. Chouikhi, H. Chniter, and F. Jarray, "Arabic Sentiment Analysis Using BERT Model," in *International Conference on Computational Collective Intelligence*. Springer, 2021, pp. 621–632.

[82] Y. Arslan, K. Allix, L. Veiber, C. Lothritz, T. F. Bissyandé, J. Klein, and A. Goujon, "A Comparison of Pre-Trained Language Models for Multi-Class Text Classification in the Financial Domain," in *Companion Proceedings of the Web Conference 2021*, ser. WWW '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 260–268.

[83] H. E. Kiziloz, "Classifier Ensemble Methods in Feature Selection," *Neurocomputing*, vol. 419, pp. 97–107, 2021.

[84] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature Selection: A Data Perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 94, 2018.

[85] C. T. Tran, M. Zhang, P. Andreae, and B. Xue, "Improving Performance for Classification with Incomplete Data Using Wrapper-Based Feature Selection," *Evolutionary Intelligence*, vol. 9, no. 3, pp. 81–94, 2016.

[86] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[87] I. Perikos, S. Kardakis, and I. Hatzilygeroudis, "Sentiment Analysis Using Novel and Interpretable Architectures of Hidden Markov Models," *Knowledge-Based Systems*, vol. 229, p. 107332, 2021.

[88] Y. Cheng, H. Sun, H. Chen, M. Li, Y. Cai, Z. Cai, and J. Huang, "Sentiment Analysis Using Multi-head Attention Capsules with Multi-channel CNN and Bidirectional GRU," *IEEE Access*, vol. 9, pp. 60 383–60 395, 2021.

[89] B. Pang and L. Lee, "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2005, pp. 115–124.

[90] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An Attention-Based Bidirectional CNN-RNN Deep Model for Sentiment Analysis," *Future Generation Computer Systems*, vol. 115, pp. 279–294, 2021.

[91] M. Kamyab, G. Liu, and M. Adjeisah, "Attention-Based CNN and Bi-LSTM Model Based on TF-IDF and GloVe Word Embedding for Sentiment Analysis," *Applied Sciences*, vol. 11, no. 23, p. 11255, 2021.

[92] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification Using Distant Supervision," *CS224N project report, Stanford*, vol. 1, no. 12, p. 2009, 2009.

[93] C. Couharde, H. Bennani, and Y. Wallois, "Do IMF Reports Affect Market Expectations? A Sentiment Analysis Approach," University of Paris Nanterre, EconomiX Working Papers 2021-6, 6 2021.

[94] M. Breen, D. Hodson, and M. Moschella, "Incoherence in Regime Complexes: A Sentiment Analysis of EU-IMF Surveillance," *JCMS: Journal of Common Market Studies*, vol. 58, no. 2, pp. 419–437, 2020.

[95] A. Deniz, M. Angin, and P. Angin, "Understanding IMF Decision-Making with Sentiment Analysis," in *2022 30th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2022, pp. 1–4.

[96] World Health Organization, "WHO Coronavirus (COVID-19) Dashboard," https://covid19.who.int/, accessed: 2023-01-24.

[97] A. K. Uysal and S. Gunal, “The Impact of Preprocessing on Text Classification,” *Information Processing & Management*, vol. 50, no. 1, pp. 104–112, 2014.

[98] M. Y. Kiang, “A Comparative Assessment of Classification Methods,” *Decision support systems*, vol. 35, no. 4, pp. 441–454, 2003.

[99] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[100] Z. Wang and Z. Lin, “Optimal Feature Selection for Learning-Based Algorithms for Sentiment Classification,” *Cognitive Computation*, vol. 12, no. 1, pp. 238–248, 2020.

[101] A. Sharma and S. Dey, “A Comparative Study of Feature Selection and Machine Learning Techniques for Sentiment Analysis,” in *Proceedings of the 2012 ACM research in applied computation symposium*, 2012, pp. 1–7.

[102] A. Madasu and S. Elango, “Efficient Feature Selection Techniques for Sentiment Analysis,” *Multimedia Tools and Applications*, vol. 79, no. 9, pp. 6313–6335, 2020.

[103] S. R. Ahmad, A. A. Bakar, and M. R. Yaakub, “Metaheuristic Algorithms for Feature Selection in Sentiment Analysis,” in *2015 Science and Information Conference (SAI)*. IEEE, 2015, pp. 222–226.

[104] L. Shang, Z. Zhou, and X. Liu, “Particle Swarm Optimization-Based Feature Selection in Sentiment Classification,” *Soft Computing*, vol. 20, no. 10, pp. 3821–3834, 2016.

[105] A. Kumar and R. Khorwal, “Firefly Algorithm for Feature Selection in Sentiment Analysis,” in *Computational Intelligence in Data Mining*. Springer, 2017, pp. 693–703.

[106] O. Gokalp, E. Tasci, and A. Ugur, “A Novel Wrapper Feature Selection Algorithm Based on Iterated Greedy Metaheuristic for Sentiment Classification,” *Expert Systems with Applications*, vol. 146, p. 113176, 2020.

[107] S. Maldonado, R. Weber, and F. Famili, "Feature Selection for High-dimensional Class-imbalanced Data Sets Using Support Vector Machines," *Information sciences*, vol. 286, pp. 228–246, 2014.

[108] H. Liu and L. Yu, "Toward Integrating Feature Selection Algorithms for Classification and Clustering," *IEEE Transactions on knowledge and data engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[109] T. Bhattacharyya, B. Chatterjee, P. K. Singh, J. H. Yoon, Z. W. Geem, and R. Sarkar, "Mayfly in Harmony: A New Hybrid Meta-Heuristic Feature Selection Algorithm," *IEEE Access*, vol. 8, pp. 195 929–195 945, 2020.

[110] Q. Al-Tashi, S. J. Abdulkadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Approaches to Multi-objective Feature Selection: A Systematic Literature Review," *IEEE Access*, vol. 8, pp. 125 076–125 096, 2020.

[111] H. E. Kiziloz, A. Deniz, T. Dokeroglu, and A. Cosar, "Novel Multiobjective TLBO Algorithms for the Feature Subset Selection Problem," *Neurocomputing*, vol. 306, pp. 94–107, 2018.

[112] R. Sihwail, K. Omar, K. A. Z. Ariffin, and M. Tubishat, "Improved Harris Hawks Optimization Using Elite Opposition-Based Learning and Novel Search Mechanism for Feature Selection," *IEEE Access*, vol. 8, pp. 121 127–121 145, 2020.

[113] Y. Hu, Y. Zhang, and D. Gong, "Multiobjective Particle Swarm Optimization for Feature Selection with Fuzzy Cost," *IEEE Transactions on Cybernetics*, vol. 51, no. 2, pp. 874–888, 2020.

[114] Y. Zhang, S. Cheng, Y. Shi, D.-w. Gong, and X. Zhao, "Cost-sensitive Feature Selection Using Two-archive Multi-objective Artificial Bee Colony Algorithm," *Expert Systems with Applications*, vol. 137, pp. 46–58, 2019.

[115] Y. Zhang, D.-w. Gong, X.-z. Gao, T. Tian, and X.-y. Sun, "Binary Differential Evolution with Self-learning for Multi-objective Feature Selection," *Information Sciences*, vol. 507, pp. 67–85, 2020.

[116] G. Ansari, T. Ahmad, and M. N. Doja, “Hybrid Filter-Wrapper Feature Selection Method for Sentiment Classification,” *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9191–9208, 2019.

[117] A. C. Pandey, D. S. Rajpoot, and M. Saraswat, “Twitter Sentiment Analysis Using Hybrid Cuckoo Search Method,” *Information Processing & Management*, vol. 53, no. 4, pp. 764–779, 2017.

[118] M. Tubishat, M. A. Abushariah, N. Idris, and I. Aljarah, “Improved Whale Optimization Algorithm for Feature Selection in Arabic Sentiment Analysis,” *Applied Intelligence*, vol. 49, no. 5, pp. 1688–1707, 2019.

[119] M. A. Hassonah, R. Al-Sayyed, A. Rodan, A.-Z. Ala’M, I. Aljarah, and H. Faris, “An Efficient Hybrid Filter and Evolutionary Wrapper Approach for Sentiment Analysis of Various Topics on Twitter,” *Knowledge-Based Systems*, vol. 192, p. 105353, 2020.

[120] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, and M. Lang, “Benchmark for Filter Methods for Feature Selection in High-dimensional Classification Data,” *Computational Statistics & Data Analysis*, vol. 143, p. 106839, 2020.

[121] S. R. Ahmad, A. A. Bakar, and M. R. Yaakub, “Ant Colony Optimization for Text Feature Selection in Sentiment Analysis,” *Intelligent Data Analysis*, vol. 23, no. 1, pp. 133–158, 2019.

[122] J. R. Quinlan, “Induction of Decision Trees,” *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[123] T. Kucukyilmaz, A. Deniz, and H. E. Kiziloz, “Boosting Gender Identification Using Author Preference,” *Pattern Recognition Letters*, vol. 140, pp. 245–251, 2020.

[124] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[125] A. Deniz, H. E. Kiziloz, T. Dokeroglu, and A. Cosar, “Robust Multiobjective Evolutionary Feature Subset Selection Algorithm for Binary Classification Us-

ing Machine Learning Techniques," *Neurocomputing*, vol. 241, pp. 128–146, 2017.

[126] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[127] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical Word Embeddings," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15. AAAI Press, 2015, p. 2418–2424.

[128] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[129] M. Kamkarhaghighi and M. Makrehchi, "Content Tree Word Embedding for Document Representation," *Expert Systems with Applications*, vol. 90, pp. 241–249, 2017.

[130] G. Katz, N. Ofek, and B. Shapira, "ConSent: Context-Based Sentiment Analysis," *Knowledge-Based Systems*, vol. 84, pp. 162–178, 2015.

[131] C. Dos Santos and M. Gatti, "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts," in *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, 2014, pp. 69–78.

[132] A. Zhao and Y. Yu, "Knowledge-enabled BERT for Aspect-Based Sentiment Analysis," *Knowledge-Based Systems*, vol. 227, p. 107220, 2021.

[133] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained Models for Natural Language Processing: A Survey," *Science China Technological Sciences*, vol. 63, no. 10, pp. 1872–1897, 2020.

[134] J. Zhou, J. Tian, R. Wang, Y. Wu, W. Xiao, and L. He, "SentiX: A Sentiment-aware Pre-trained Model for Cross-Domain Sentiment Analysis," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 568–579.

[135] H. Xu, B. Liu, L. Shu, and P. Yu, "BERT Post-training for Review Reading Comprehension and Aspect-Based Sentiment Analysis," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2019.

[136] Z. Zhang, S. Wu, D. Jiang, and G. Chen, "BERT-JAM: Maximizing the Utilization of BERT for Neural Machine Translation," *Neurocomputing*, vol. 460, pp. 84–94, 2021.

[137] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to Fine-tune BERT for Text Classification?" in *China national conference on Chinese computational linguistics*. Springer, 2019, pp. 194–206.

[138] A. Deniz, M. Angin, and P. Angin, "Sentiment and Context-refined Word Embeddings for Sentiment Analysis," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 927–932.

[139] A. Htait and L. Azzopardi, "AWESSOME: An Unsupervised Sentiment Intensity Scoring Framework Using Neural Word Embeddings," in *European Conference on Information Retrieval (ECIR)*. Springer, 2021, pp. 509–513.

[140] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. Le, "Rethinking Pre-training and Self-training," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[141] J. Du, É. Grave, B. Gunel, V. Chaudhary, O. Celebi, M. Auli, V. Stoyanov, and A. Conneau, "Self-training Improves Pre-training for Natural Language Understanding," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 5408–5418.

[142] A. Gupta, S. Menghani, S. K. Rallabandi, and A. W. Black, "Unsupervised Self-Training for Sentiment Analysis of Code-Switched Data," in *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*. Online: Association for Computational Linguistics, Jun. 2021, pp. 103–112.

[143] Y. Zhuang, T. Jiang, and E. Riloff, “Affective Event Classification with Discourse-enhanced Self-training,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 5608–5617.

[144] E. Castillo, O. Cervantes, D. Vilarino, D. Báez, and A. Sánchez, “UDLAP: Sentiment Analysis Using a Graph-Based Representation,” in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 556–560.

[145] K. Schouten, O. Van Der Weijde, F. Frasincar, and R. Dekker, “Supervised and Unsupervised Aspect Category Detection for Sentiment Analysis with Co-occurrence Data,” *IEEE Transactions on Cybernetics*, vol. 48, no. 4, pp. 1263–1275, 2017.

[146] V. Setlur and A. Kumar, “Sentifiers: Interpreting Vague Intent Modifiers in Visual Analysis Using Word Co-occurrence and Sentiment Analysis,” in *2020 IEEE Visualization Conference (VIS)*. IEEE, 2020, pp. 216–220.

[147] P. Angin and B. Bhargava, “A Confidence Ranked Co-Occurrence Approach for Accurate Object Recognition in Highly Complex Scenes,” *Journal of Internet Technology*, vol. 14, no. 1, pp. 13–19, 2013.

[148] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[149] G. A. Miller, “WordNet: A Lexical Database for English,” *Commun. ACM*, vol. 38, no. 11, p. 39–41, nov 1995.

[150] D. W. Otter, J. R. Medina, and J. K. Kalita, “A Survey of the Usages of Deep Learning for Natural Language Processing,” *IEEE transactions on neural networks and learning systems*, vol. 32, no. 2, pp. 604–624, 2020.

[151] I. Chaturvedi, E. Cambria, R. E. Welsch, and F. Herrera, “Distinguishing Be-

tween Facts and Opinions for Sentiment Analysis: Survey and Challenges," *Information Fusion*, vol. 44, pp. 65–77, 2018.

[152] K. Dashtipour, M. Gogate, E. Cambria, and A. Hussain, "A Novel Context-aware Multimodal Framework for Persian Sentiment Analysis," *Neurocomputing*, vol. 457, pp. 377–388, 2021.

[153] T. Daudert, "Exploiting Textual and Relationship Information for Fine-grained Financial Sentiment Analysis," *Knowledge-Based Systems*, vol. 230, p. 107389, 2021.

[154] M. Devi Sri Nandhini and G. Pradeep, "A Hybrid Co-occurrence and Ranking-Based Approach for Detection of Implicit Aspects in Aspect-Based Sentiment Analysis," *SN Computer Science*, vol. 1, no. 3, pp. 1–9, 2020.

[155] D. Pinto, D. Vilariño, S. León, M. Jasso, and C. Lucero, "BUAP: Polarity Classification of Short Texts," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 154–159.

[156] M. Ghosh and G. Sanyal, "An Ensemble Approach to Stabilize the Features for Multi-domain Sentiment Analysis Using Supervised Machine Learning," *Journal of Big Data*, vol. 5, no. 1, pp. 1–25, 2018.

[157] N. Al-Twairesh and H. Al-Negheimish, "Surface and Deep Features Ensemble for Sentiment Analysis of Arabic Tweets," *IEEE Access*, vol. 7, pp. 84 122–84 131, 2019.

[158] H. T. Phan, V. C. Tran, N. T. Nguyen, and D. Hwang, "Improving the Performance of Sentiment Analysis of Tweets Containing Fuzzy Sentiment Using the Feature Ensemble Model," *IEEE Access*, vol. 8, pp. 14 630–14 641, 2020.

[159] A. Onan, "Sentiment Analysis on Twitter Based on Ensemble of Psychological and Linguistic Feature Sets," *Balkan Journal of Electrical and Computer Engineering*, vol. 6, no. 2, pp. 69–77, 2018.

[160] M. M. Fouad, T. F. Gharib, and A. S. Mashat, "Efficient Twitter Sentiment Analysis System with Feature Selection and Classifier Ensemble," in *International conference on advanced machine learning technologies and applications*. Springer, 2018, pp. 516–527.

[161] N. Saleena *et al.*, “An Ensemble Classification System for Twitter Sentiment Analysis,” *Procedia computer science*, vol. 132, pp. 937–946, 2018.

[162] Y. Görmez, Y. E. Işık, M. Temiz, and Z. Aydın, “FBSEM: A Novel Feature-Based Stacked Ensemble Method for Sentiment Analysis,” *International Journal of Information Technology and Computer Science*, vol. 6, pp. 11–22, 2020.

[163] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, “Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications,” *Expert Systems with Applications*, vol. 77, pp. 236–246, 2017.

[164] A. Grover and J. Leskovec, “Node2Vec: Scalable Feature Learning for Networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 855–864.

[165] J. Hartmann, M. Heitmann, C. Siebert, and C. Schamp, “More Than a Feeling: Accuracy and Application of Sentiment Analysis,” *International Journal of Research in Marketing*, 2022.

[166] M. Biesialska, K. Biesialska, and H. Rybinski, “Leveraging Contextual Embeddings and Self-attention Neural Networks with Bi-attention for Sentiment Analysis,” *Journal of Intelligent Information Systems*, vol. 57, no. 3, pp. 601–626, 2021.

[167] R. Xiang, E. Chersoni, Q. Lu, C.-R. Huang, W. Li, and Y. Long, “Lexical Data Augmentation for Sentiment Analysis,” *Journal of the Association for Information Science and Technology*, vol. 72, no. 11, pp. 1432–1447, 2021.

[168] T. Zhang, X. Gong, and C. P. Chen, “BMT-Net: Broad Multitask Transformer Network for Sentiment Analysis,” *IEEE Transactions on Cybernetics*, 2021.

[169] H. S. S. Al-deen, Z. Zeng, R. Al-sabri, and A. Hekmat, “An Improved Model for Analyzing Textual Sentiment Based on a Deep Neural Network Using Multi-head Attention Mechanism,” *Applied System Innovation*, vol. 4, no. 4, p. 85, 2021.

[170] A. Deniz and H. E. Kiziloz, “Effects of Various Preprocessing Techniques to Turkish Text Categorization Using N-gram Features,” in *2017 International*

Conference on Computer Science and Engineering (UBMK). IEEE, 2017, pp. 655–660.

[171] Z. Rahimi and M. M. Homayounpour, “The Impact of Preprocessing on Word Embedding Quality: A Comparative Study,” *Language Resources and Evaluation*, pp. 1–35, 2022.

CURRICULUM VITAE

Personal Information

Surname, Name:	Deniz Kızılöz, Firdevsi Ayça
Google Scholar profile:	`https://scholar.google.com.tr/citations?user=Ys1WOCkAAAAJ`
ORCID profile:	`https://orcid.org/0000-0002-9276-4811`

Education

Date	Degree	Institution
Jan 2023	Doctor of Philosophy in Computer Engineering	Middle East Technical University Ankara, Turkey
Aug 2016	Master of Science in Computer Engineering	Middle East Technical University Ankara, Turkey
Aug 2012	Bachelor of Science in Computer Engineering	TOBB University of Economics and Technology Ankara, Turkey

Work Experience

Years	Position	Institution
2022 -	Software Engineer	Google London, United Kingdom
2018 - 2020	Software Engineer	The Open University Milton Keynes, United Kingdom
2015 - 2018	Research Assistant	TED University Ankara, Turkey
2013 - 2015	Software Engineer	TOBB University of Economics and Technology Ankara, Turkey

Research Interests

Machine Learning — Natural Language Processing — Feature Selection — Evolutionary Computation — Multiobjective Optimization

Publications

Ph.D. Thesis Related Publications

▷ Deniz, A., Angin, M., & Angin, P. (2022, May). Understanding IMF Decision-Making with Sentiment Analysis. *In 2022 30th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4).

DOI: 10.1109/SIU55565.2022.9864926

▷ Deniz, A., Angin, M., & Angin, P. (2021, October). Sentiment and Context-refined Word Embeddings for Sentiment Analysis. *In 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 927-932).

DOI: 10.1109/SMC52423.2021.9659189

▷ Deniz, A., Angin, M., & Angin, P. (2021). Evolutionary Multiobjective Feature Selection for Sentiment Analysis. *IEEE Access*, 9, 142982-142996.

DOI: 10.1109/ACCESS.2021.3118961

▷ Deniz, A., Angin, M., & Angin, P. (*under review*) A Confidence Ranked Propagation Method for Unsupervised Sentiment Analysis via Pre-trained Language Models.

▷ Deniz, A., Angin, M., & Angin, P. (*under review*) A Context- and Sentiment-Oriented Feature Ensemble Model for Sentiment Analysis.

Other Publications

▷ Deniz, A., & Kiziloz, H. E. (2022, July). Boosting Initial Population in Multiobjective Feature Selection with Knowledge-based Partitioning. In 2022 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8).

▷ Dokeroglu, T., Deniz, A., & Kiziloz, H. E. (2022). A Comprehensive Survey on Recent Metaheuristics for Feature Selection. Neurocomputing, 494, 269-296.

▷ Deniz, A., Kiziloz, H. E., Sevinc, E., & Dokeroglu, T. (2022). Predicting the Severity of COVID-19 Patients using a Multi-threaded Evolutionary Feature Selection Algorithm. Expert Systems, e12949.

▷ Dokeroglu, T., Deniz, A., & Kiziloz, H. E. (2021). A Robust Multiobjective Harris' Hawks Optimization Algorithm for the Binary Classification Problem. Knowledge-Based Systems, 227, 107219.

▷ Kiziloz, H. E., & Deniz, A. (2021). An Evolutionary Parallel Multiobjective Feature Selection Framework. Computers & Industrial Engineering, 159, 107481.

▷ Kucukyilmaz, T., Deniz, A., & Kiziloz, H. E. (2020). Boosting Gender Identification using Author Preference. Pattern Recognition Letters, 140, 245-251.

▷ Kiziloz, H. E., & Deniz, A. (2020, October). Feature Selection with Dynamic Classifier Ensembles. In 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 2038-2043).

▷ Deniz, A., & Kiziloz, H. E. (2020, September). Parallel Multiobjective Feature Selection for Binary Classification. In 2020 5th International Conference on Computer Science and Engineering (UBMK) (pp. 1-5).

▷ Deniz, A., & Kiziloz, H. E. (2019). On Initial Population Generation in Feature Subset Selection. Expert Systems with Applications, 137, 11-21.

▷ Kiziloz, H. E., Deniz, A., Dokeroglu, T., & Cosar, A. (2018). Novel Multiobjective TLBO Algorithms for the Feature Subset Selection Problem. Neurocomputing, 306, 94-107.

▷ Deniz, A., & Kiziloz, H. E. (2017, October). Effects of Various Preprocessing Techniques to Turkish Text Categorization using N-gram Features. In 2017 International Conference on Computer Science and Engineering (UBMK) (pp. 655-660).

▷ Deniz, A., Kiziloz, H. E., Dokeroglu, T., & Cosar, A. (2017). Robust Multiobjective Evolutionary Feature Subset Selection Algorithm for Binary Classification using Machine Learning Techniques. Neurocomputing, 241, 128-146.

▷ Mani, G., Kim, M., Bhargava, B., Angin, P., Deniz, A., & Pasumarti, V. (*under review*) Malware Speaks! Deep Learning Based Assembly Code Processing for Detecting Evasive Cryptojacking.