12-2019

Social Media Sentiment Analysis with a Deep Neural Network: An Enhanced Approach Using User Behavioral Information

Ahmed Sulaiman M. Alharbi
Western Michigan University, ahmed.s.m.alharbi@gmail.com

SOCIAL MEDIA SENTIMENT ANALYSIS WITH A DEEP NEURAL NETWORK: AN ENHANCED APPROACH USING USER BEHAVIORAL INFORMATION

by

Ahmed Sulaiman M Alharbi

A dissertation submitted to the Graduate College
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
Computer Science
Western Michigan University
December 2019

Doctoral Committee:

Dr. Elise de Doncker, Ph.D., Chair
Dr. Alvis Fong, Ph.D.
Dr. Ikhlas Abdel-Qader, Ph.D.

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious and the Most Merciful. I thank Almighty Allah for giving me the opportunity, determination, and strength to do my research.

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Elise de Doncker, for the continuous support of my Ph.D. research and motivation. Her guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Ikhlas Abdel-Qader and Dr. Alvis Fong, for their insightful comments and encouragement.

A special thanks to my family. Words can not express how grateful I am to my parents: Sulaiman Alharbi and Asmaa Alwafi, for all of the sacrifices that you have made on my behalf. Your prayer for me was what sustained me thus far. Also, I would like to thank my brothers: Mohammed, Omar, Abdulrahman, and Ammar, and my sister Sarah who supported me and helped me throughout my life.

To my lovely children, Eyad, Lena, and Khalid, who are the pride and joy of my life, I love you more than anything and I appreciate all your patience and support during my studies. You are the reason why I always keep the spirit to finish things that I have fired.

Last but not least, I would like to express my deepest gratitude to my wife, Hanen Alwafi, for her patience and tolerance over the last ten years. Hanen, having you is the best thing that has ever occurred to me in my entire life. I could not be able to finish this work without your support. Thank you for being with me and for your appreciated sacrifices. I thank Allah for enlightening my life with your presence.

Ahmed Sulaiman M Alharbi

SOCIAL MEDIA SENTIMENT ANALYSIS WITH A DEEP NEURAL NETWORK: AN ENHANCED APPROACH USING USER BEHAVIORAL INFORMATION

Ahmed Sulaiman M Alharbi, Ph.D.

Western Michigan University, 2019

Sentiment analysis on social media such as Twitter has become a very important and challenging task. Due to the characteristics of such data (including tweet length, spelling errors, abbreviations, and special characters), the sentiment analysis task in such an environment requires a non-traditional approach. Moreover, social media sentiment analysis constitutes a fundamental problem with many interesting applications, such as for Business Intelligence, Medical Monitoring, and National Security. Most current social media sentiment classification methods judge the sentiment polarity primarily according to textual content and neglect other information on these platforms. In this research, we propose deep learning based frameworks that also incorporate user behavioral information within a given document (tweet). Within these frameworks, there are several models based on a variety of neural network architectures. Each of these models is trained on a specific aspect of user behavior. Then, the frameworks exploit these multi-aspect learning models to jointly take on a mutual task (the sentiment analysis task). The results of the preliminary experiments, which are reported in [1]–[3], demonstrate that going beyond the content of a document is beneficial in sentiment classification, because it provides the classifier with a deeper understanding of the task.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii

LIST OF TABLES vi

LIST OF FIGURES vii

1. INTRODUCTION 1

   1.1. Research Questions 2

   1.2. Overview of this Dissertation 7

2. LITERATURE REVIEW 9

   2.1. What is Sentiment Analysis? 9

      2.1.1. The Opinion Holder 9

      2.1.2. The Opinion Target 10

      2.1.3. The Opinion Itself 10

   2.2. Sentiment Analysis Methodology 10

   2.3. Deep Learning 11

      2.3.1. Deep Learning for Twitter Sentiment Analysis 12

   2.4. Learning Text Representation 15

      2.4.1. Learning Text Representation in DL 15

      2.4.2. Word Level Representation (Word Embedding) 16

      2.4.3. Character Level Embeddings 18

3. TWITTER SENTIMENT ANALYSIS WITH A DEEP NEURAL NETWORK: AN ENHANCED APPROACH USING USER BEHAVIORAL INFORMATION . . . 20

3.1. Introduction 20
3.2. Related Work 21
3.3. Methodology 22
3.3.1. Deep Learning Architecture 22
3.3.2. Features 24
3.4. Experiments and Results 31
3.4.1. Experimental Settings 31
3.4.2. Datasets 33
3.4.3. Results 34
3.4.4. Baseline Methods 34
3.4.5. Results and Analysis 36
3.5. Conclusion 39

4. EMOTIONAL AWARENESS BASED CLASSIFICATION MODEL FOR TWITTER SENTIMENT ANALYSIS USING A DEEP NEURAL NETWORK 40
4.1. Introduction 40
4.2. Related Work 41
4.3. Proposed Approach 42
4.4. Experiments and Results 43
4.4.1. Data Sources 43
4.4.2. Preprocessing 46
4.4.3. Results 46
4.5. Conclusions 49

5. INCORPORATING SOCIAL RELATIONS BETWEEN USERS To IMPROVE THE PERFORMANCE OF TWITTER SENTIMENT ANALYSIS 50

5.1. Methodology (Proposed Approach) 50
5.1.1. Profile Level Similarity (PLS) 51
5.1.2. Timeline Level Similarity (TLS) 54
5.1.3. Content Level Similarity (CLS) 56
5.2. Proposed Models 57
5.3. Experiments and Results 59
5.3.1. Dataset Sources 59
5.3.2. Preprocessing 60
5.3.3. Usefulness of Topic Context 60
5.3.4. Results 62

6. THE EFFECT OF DETECTING POLARITY AT THE TOPIC LEVEL ON SENTIMENT CLASSIFICATION PERFORMANCE 68
6.1. Methodology (Proposed Approach) 68
6.2. Experiments and Results 75
6.2.1. Dataset Sources 75
6.2.2. Preprocessing 76
6.2.3. Results 77

7. CONCLUSIONS AND FUTURE WORK 82

BIBLIOGRAPHY 86

APPENDIX 101
A. Some Source Codes Used in this Work 102

LIST OF TABLES

3.1 List of features used in this paper 25
3.2 Label distribution for two Twitter datasets 34
3.3 User distribution in datasets 34
3.4 The feature sets in different combinations 36
3.5 Accuracy rates of CNN, SVM, NB, J48, KNN, and LSTM 37
3.6 . F1, precision, and recall scores of the six models in sets No. 4 and 7 38

4.1 The number of tweets for each dataset. 44
4.2 The number of tweets for each SemEval dataset. 44
4.3 Performance comparison of models. Best scores are in bold. 47

5.1 Statistics of datasets 60
5.2 Experimental results of our models, compared with baselines on the SemEval dataset 64
5.3 Experimental results of our models, compared with baselines on the HCR dataset 64

6.1 SemEval-(2015-2016)-BD datasets statistics 75
6.2 SemEval-2016-CE dataset statistics 75
6.3 Experimental results of our model, compared with baselines 78

LIST OF FIGURES

3.1 The architecture of the proposed model 23
3.2 Steps for conducting the experiments 32
3.3 Precision for the negative label of CNN and SVM 37
3.4 Recall for the negative label of CNN and SVM. 38

4.1 Percentage of tweets in each emotional category. 45
4.2 Averaged precision and recalls of the models. 48

5.1 Percentage of agreement in opinions among authors, based on a given topic in the datasets HCR (on the right) and SemEval (on the left). 61
5.2 Recall and precision values comparison of the positive label on the HCR dataset. 65
5.3 Comparing average precision, recall, and F1 values of our models for the SemEval and HCR datasets. 66

6.1 High-level overview of our framework. 70
6.2 Recall and precision values comparison of the positive label. 79
6.3 Recall and precision values comparison of the negative label. 80

CHAPTER 1

INTRODUCTION

The emergence of social media platforms has given web users a space for expressing and sharing their thoughts and opinions on all kinds of topics and events. One of the most popular social networking platforms is Twitter[1]. It allows people to publish messages to express their interests, favorites, opinions, and sentiments towards various topics and issues they encounter in their daily life. The messages are called tweets, which are real-time and at most 140 characters.

Twitter gives access to the unprompted views of a wide set of users on particular products or events. The opinions or expressions of sentiment about organizations, products, events, and people have proven extremely useful for marketing [4] and social studies [5]. Twitter has about 200 billion tweets per year, 500 million tweets per day, 350,000 tweets per minute, and 6,000 tweets per second [2]. These are some reasons for choosing Twitter as a case study in this research.

In contrast to standard texts with many words that help gather sufficient statistics, the texts in social media, especially Twitter, only consist of a limited number of characters. Moreover, when a user posts a message (tweet), it may have new abbreviations or acronyms that appear infrequently in conventional text documents. Therefore, applying traditional methods to such an environment will not yield acceptable performance.

In addition, most existing approaches to sentiment analysis on microblogs, such as Twitter, treat the texts written by different users equally and leave out user characteristics [6]. Such approaches carry out this task (sentiment analysis) as a linguistic phenomenon, without

[1] http://twitter.com/

[2] http://www.internetlivestats.com/twitter-statistics/

emphasis on the aspects of the writer (user) and microblog-specific aspects, which emerge from the non-content aspects. For example, the user aspects (or characteristics) include emotional states (mood) and writing styles; and microblog-specific aspects include user social activity, network density, topic sentiment quantification and so on. We look at sentiment analysis on microblogs as a task of integrating information from multiple dimensions, not only from one dimension (i.e., the content).

As traditional machine learning based methods extract the common sentiment features of texts, they often fail to distinguish the above aspects, resulting in a decrease of performance of sentiment classifiers.

Therefore, it is necessary to address the above issues and develop a more fine-grained method for sentiment classification, especially for microblog data. In this thesis, several deep learning based approaches are proposed that go beyond the content of a given document (tweet). It takes into account, besides a given tweet, the characteristics of the user who wrote that tweet, and microblog-specific aspects. It has been demonstrated by the experiments conducted in [1] that this type of approach increases the performance and the accuracy of sentiment classification tasks on social media.

This thesis is structured as follows. Section 1.1 provides a detailed description of the research questions investigated in this thesis. We outline the background and literature review for our work in Chapter 2. Chapters 3, 4, 5 and 6 address our four research questions, chapter for each question. Finally, we give conclusions and discuss future work in Chapter 7. Appendix A provides some source codes that used in this work.

1.1. Research Questions

The main research questions investigated in this thesis are:

RQ1: Is there a relationship between the user's behavior and his/her posts?

And, if such a relationship exists, can it be used to enhance sentiment analysis performance?

To address this question, we propose a neural network based model that classifies tweets into positive and negative categories based on a proposed set of features to enhance the classification performance. These features help the model to understand the user's behavior [1].

The main focus, as noted, is toward improving the performance of Twitter sentiment analysis classifiers by developing solutions that consider user behavior in the sentiment detection work-flow. We aspire to achieve improvements in sentiment analysis performance by addressing the following research sub-questions:

RQ2: What emotion (mood) did the author express prior to the tweet that is to be classified? Can this information enhance the model performance?

Most machine learning (or even deep learning) based methods often treat sentiment analysis on Twitter as a traditional problem of text categorization, for which a fundamental issue is to extract effective textual features from a corpus regardless of the emotional state of the tweet author. In other words, the focus of such method is on the target tweet only (i.e., a tweet to be classified).

According to [7], the emotional state can influence an author's opinion. Moreover, most current research on machine learning based sentiment classification leaves out user properties in the classification process. Our hypothesis with respect to extracting and detecting the emotional state of an author in the sentiment analysis task is:

H2: Determining the emotional state of the author of a tweet can enhance the sentiment analysis performance.

Thus, to test the above hypothesis we propose an emotion based sentiment classification model. The proposed model classifies the target tweet by predicting the user's emotional state according to the tweets posted combined with textual features in the target tweet.

The main objective of this question is that given historical tweets of a user, determine the intensity of the emotion E (anger, fear, joy, or sadness) that best represents the emotional state of the tweeter, and assigns a real-valued score between 0 (least E) and 1 (most E).

RQ3: Can incorporating social relations between users improve the performance of Twitter sentiment analysis?

Besides textual content, a networking characteristic is a distinct feature of social media settings through user relationships because it contains rich social context information, as inspired by the social sciences findings that friends usually hold a similar opinion bias towards the same targets, which is formulated as homophily [8]. According to [9] and [10], homophily also exists in social media platforms. This suggests that connected individuals are more likely to have similar behaviors or hold similar opinions.

There is another finding of social sciences, besides the homophily, called co-citation regularity [11]. It indicates that similar users tend to refer or connect to the same things. For instance, when two users write tweets on similar topics, they probably have similar judgments or attitudes in other things or have other common interests. The co-citation regularity concept helps with discovering implicit connections between users, while the homophily helps with explicit ones.

Our hypothesis for this question is:

H3: Incorporating the implicit social relations of microblogging users may help strengthen the learning of a personalized sentiment classifier.

To test this hypothesis, a method needs to be developed to determine how social relationships can be modeled and utilized for a deep learning approach for sentiment analysis. This includes a methodology for creating an effective representation to capture social relationships features between users in implicit ways.

Another issue that needs to be addressed is the lack of adequately labeled training data.

The publicly available datasets lack social relationship information between users. Furthermore, the number of labeled tweets in these datasets is also not sufficient for learning representations using a deep learning approach. One possible solution is to use the distant supervision approach [12]. It assumes that emoticons, hashtags, emojis, etc., are good indicators of the sentiment in tweets and can be used to automatically assign sentiment labels. There are several papers on Twitter sentiment analysis [13]–[15] have relied on the distant supervision approach to automatically construct large corpora of annotated tweets for sentiment classifier training.

Potential contributions in this line of work are summarized as follows:

- Empirically confirm that the probability that two users share the same opinion is indeed correlated with how they are connected in the social network.
- Propose a method for generating an embedding of users that can help to reveal their social characteristics.
- Propose a three-phase semi-supervised deep learning model to tackle the social media sentiment analysis.
- Evaluate the proposed model on real-world Twitter datasets empirically and examine the effects of social relationships on sentiment analysis.

RQ4: What is the effect of detecting polarity at the topic level on sentiment classification performance?

Users post tweets on different topics in Twitter to express their opinions and thoughts. In Twitter, hashtags are used to represent the topics. Twitter users create these hashtags by simply prefixing a word or a phrase with a hash symbol. The extensive use of hashtags makes Twitter more expressive and welcome to people.

While tweet level sentiment analysis results provide very useful information, the overall or general sentiment tendencies towards topics are more appealing in some cases [16]. The sentiment analysis level addressed in this research question is carried out at the topic level, also known as topic-level polarity.

The main task at this level of analysis is to automatically generate the overall sentiment polarity for a given hashtag or a generated topic. These topics could be classified on a two-point scale (positive or negative) or five-point scale (highly positive, positive, neutral, negative, or highly negative).

Our hypothesis for this question is:

H4: Incorporating topic-level polarity can enhance Twitter sentiment analysis performance.

To test the above hypothesis we propose a model based on a deep learning approach for topic-based Twitter sentiment analysis. This model consists of three modules: Input module, Topic sentiment quantification module, and Merging module. The following is an overview of these modules.

Input module (IM): The input to this module is a tweet, treated as a sequence of words. After performing pre-processing tasks on the tweet, the input module encodes the tweet into word embeddings via a recurrent neural network, which are given as inputs to the Merging module. This module also extracts the topics from the tweet and sends them to the Topic sentiment quantification module.

Topic sentiment quantification module (TSQM): The aim of this module is to estimate the distribution of tweets on a given topic across the different classes. There are two main approaches to perform this task [17]. One is the Classify and Count (CC) (i.e., aggregative) approach, where we compute the fraction of a topic's messages that belong to each class mentioned in the dataset. Another approach is the non-aggregative approach where the estimating of a class distribution is done holistically [18]. Most papers in the literature use an aggregative approach, where the classification of each individual tweet is required as an intermediate step to produce a probability distribution over the classes. Therefore, a deep learning based classifier is proposed to classify all tweets for the given topic. Then, to estimate a class distribution, a Probabilistic Classify and Count (PCC) quantification based approach is applied on the classified tweets. The PCC, originally proposed in [19], is a variant of CC. The produced probability distributions are given as inputs to the merging module.

Merging module (MM): The outputs from both networks (the word embeddings from IM and the topic probability distribution results from TSQM) are combined to form a single feature vector. This vector is then fed to a fully connected softmax layer, which outputs a probability distribution over all labels. The label having the highest probability is chosen as the final prediction.

Potential contributions in this line of work are summarized as follows:

- Proposing a deep learning model for sentiment analysis utilizing the topics information.
- Introducing a deep learning based model in quantification approaches for topic sentiment analysis.
- Evaluating the proposed model extensively using real-world datasets to understand the working of the proposed model.

1.2. Overview of this Dissertation

This dissertation is structured as follows. Section 1.1 introduces the four research questions that will be addressed in this work. Chapter 2 gives a literature review. The research questions RQ1, RQ2, RQ3, and RQ4 are detailed in 3, 4, 5, and 6, respectively. RQ1

(Chapter 3) was published in [1], with an extended version as a journal paper in [3]. RQ2 (Chapter 4) was published in [2]. Chapter 7 gives conclusions and discusses future directions. Appendix A provides some source codes that used in this work.

CHAPTER 2

LITERATURE REVIEW

2.1. What is Sentiment Analysis?

Sentiment analysis, also known as opinion mining, is an important type of text analysis that addresses the problems of detecting, extracting and analyzing opinion oriented text, identifying positive and negative opinions, and measuring how positively or negatively an entity (i.e., people, organization, event, location, product, topic, etc.) is regarded [20]. It is an exciting research field with the potential for a number of real world applications where discovered opinion information can be used to help people, companies, or organizations to make better decisions [21].

Essentially, there are three main elements of sentiment analysis: (i) the opinion holder, (ii) the opinion target, and (iii) the opinion itself [22]. In the following we provide a brief discussion of these three elements.

2.1.1. The Opinion Holder

The opinion holder is the entity that has the opinion being expressed, i.e., it is the owner of the opinion. This can be an individual, an organization, a group, a corporation, etc. It is divided into two types: direct and indirect opinion (holder). The direct opinion is expressed directly on an entity whereas the indirect opinion is expressed on an entity based on some positive or negative effects related to some other entities [20], [22]. In a social media setting, in particular, Twitter, opinion holders are usually the authors of the tweets.

2.1.2. The Opinion Target

The opinion target, also known as the sentiment target, of an opinion is the entity that the sentiment has been expressed upon [22]. Often it is implicit, rather than directly mentioned. That is quite frequently the case in reviews, blog comments or tweet replies, where the general context of the tweet indicates what the author is referring to [23]. Several papers focus on extracting opinion targets, for example [24] and [25].

2.1.3. The Opinion Itself

The opinion is the actual affective state that is being expressed. Extraction of opinions has been one of the main focal points of sentiment analysis [22]. For example, given a set of tweets, provide an informed estimate of the sentiment they express. This estimate takes several forms based on many aspects, such as the specific precondition of the analysis, the domain of application, etc. Typical examples of opinion analysis can include a binary classification or multi-class classification. In the binary classification, the text is classified into one of two predefined classes, usually positive or negative. More than two classes can be used in a multi-class classification [20].

2.2. Sentiment Analysis Methodology

Broadly, there are three main methods of sentiment analysis: (i) machine learning based method (supervised approach), (ii) lexicon based method (unsupervised approach), and (iii) hybrid method [20]. The following gives a brief description of each one.

The majority of papers written about sentiment analysis of social media content use a machine learning approach [26], where a labeled dataset is used to train a standard algorithm. Typical examples of such an approach include [27] and [28].

According to Paltoglou and Giachanou [20], the major drawback of this approach is the development of suitable datasets, as the trained classifiers tend to be mainly domain dependent. The main machine learning algorithms used in sentiment classification tasks are Naive Bayes (NB) and Support Vector Machine (SVM) [29].

In the lexicon based method, one or more affective dictionaries are utilized to estimate the affective content of text segments. These are word lists in which each lemma has been assigned an affective value, for instance the level of positivity or negativity it typically conveys [20]. It is important to mention that this approach combines various prose and syntactic-based rules in order to increase its accuracy [30]. There is a significant number of dictionaries that have been developed either automatically or semi-automatically [31], for example, General Inquirer (GI) [32], SentiWordNet [33] and WordNet-Affect [34]. Several studies [35], [36] have reported that models based on this approach performed adequately in a number of various social media platforms. However, the effectiveness of the lexicon based sentiment approach is characterized by the coverage and accuracy of the dictionary [17]. Also, in many cases, relying on affect words is often insufficient, and it does not lead to satisfactory results in sentiment detection [37].

The hybrid method is a mixture of the machine learning and lexicon based approaches [38]. The text is analyzed initially using a lexicon-based approach and the produced output is fed into a machine-learning algorithm as training data. The output of the second phase is utilized subsequently to expand the affective dictionary. The whole process is repeated until some termination criterion is satisfied [30].

2.3. Deep Learning

Deep Learning (DL) is a field of machine learning that uses neural networks (NNs) to learn many levels of abstraction [39]. It allows developing algorithms that provide computers with the ability to handle tasks such as image recognition and understanding natural languages [17]. DL has become popular in applications of computer vision, speech recognition, and natural language processing [40].

2.3.1. Deep Learning for Twitter Sentiment Analysis

DL has been investigated and implemented for Twitter Sentiment Analysis (TSA) by many researchers. In this section, we introduce some successful deep learning algorithms for TSA.

In Tang, Wei, Yang, *et al.* [41], an approach is proposed to learn a sentiment-specific word embedding (SSWE) from collected tweets using distant supervision. Three neural networks are developed to learn the SSWE. The learned word embedding is then used to generate features for Twitter sentiment classification. The effectiveness of the SSWE is evaluated on the SemEval-2013 dataset. Their approach yields the best performance with F1 score 86.58%, which is achieved by combining the SSWE feature with sentiment lexicons and other features used by Mohammad, Kiritchenko, and Zhu [42]. The authors argue that the high performance of their approach is attained because SSWE encodes sentiment information of text rather than the syntactic context of words.

Recently, Severyn and Moschitti [43] used a deep learning approach to classify tweets at both document and phrase levels, based on convolutional neural networks. According to the authors, the key to develop a deep neural network model that can obtain a new state-of-the-art result lies in the initialization of the parameters. Therefore, the authors provide an in-depth description of a 3-step process to train the parameters of the proposed deep neural network. The first step is word embeddings initialization. In this step a neural network model is used to initialize the word embeddings. This model is trained on a large collection of tweets. The second step consists of refining the word embeddings. To do that, the authors use a convolutional neural network and a large distant supervised corpus. Finally, the created word embeddings and the network parameters obtained from the second step are used to initialize the model. The performance of the proposed deep learning model is evaluated on benchmark datasets from the Twitter Sentiment Analysis (Task 10) of SemEval-2015. Their model ranks first in terms of accuracy on the phrase-level subtask (84.79%) and second on message-level subtask (64.59%).

Unlike most sentiment analysis models that largely rely on textual content, the research study by You, Luo, Jin, *et al.* [44] proposes a joint visual-textual model of sentiment analysis. The aim of their research is to demonstrate the effectiveness of utilizing both visual and textual content to analyze the sentiment of a document. Accordingly, a deep learning model based on a convolutional neural network for the image sentiment analysis and a paragraph vector model for textual sentiment analysis are developed. After collecting a large number of tweets using the Twitter API, VADER, a rule based sentiment classifier proposed by Hutto and Gilbert [45] is used to weakly label the tweets. Then, distributional representations are learned by these labeled tweets. In addition, a Convolutional Neural Network (CNN) model is fine-tuned by images obtained from sending a query list of positive and negative sentiment keywords to Getty Images[1] . Besides the weakly labeled data, Amazon Mechanical Turk (AMT) is used to manually label additional data for comparison purposes. A wide range of experiments on both weakly and manually labeled image tweets have been conducted. The experimental results show that the proposed model achieves better performance than the textual and visual sentiment analysis algorithms alone.

Most existing DL approaches are based on a Recurrent Neural Network (RNN), which is a type of conventional feed-forward neural network [46]. Due to its chain-like structure, the RNN is best suited for making predictions on input data that is sequential in nature, such

[1] https://www.gettyimages.com/

as Natural Language Processing (NLP) related tasks [47].

In Huang, Cao, and Dong [48], work is reported using Long Short-Term Memory (LSTM). LSTM is a specific kind of the basic RNN [47]. The authors use a hierarchical LSTM model for TSA. The model contains two levels: word and tweet level. Each level has its own LSTM model. The purpose of the first LSTM (on the word level) is to generate a representation of a single tweet. The second LSTM (on the tweet level) takes the generated representations as input and uses them to model the context of the current tweet. Apart from text-based context, the authors also adapt additional context, such as social context, conversation-based context, and topic-based context to help increasing the accuracy of the proposed model. These additional contexts are formed as binary features (i.e., features denoting presence or absence in the tweet). The dataset used in that work has 15,000 tweets consisting of 1,600 threads and covering 51 topics. Each tweet is labeled by two independent annotators into three classes: positive, negative, and neutral. The proposed model is compared with various baselines, such as SVM, LSTM-RNN, and CNN. The experimental results show that additional context information increases the performance of the model.

Most existing classification algorithms for social media sentiment analysis focus on tweet contents (document-level); for instance [49], [50]. As an illustration, let us consider the algorithm named ConSent (for “Context-based Sentiment Analysis”), which is proposed in Katz, Ofek, and Shapira [50]. ConSent has two phases, learning and detection. A set of key terms and context terms from a training set is produced in the learning phase. Then, feature vectors are generated based on these terms, to train a classifier that will be used to analyze sentiment in the detection phase. The detection phase is where the classification task takes place. It scans all tweets in search of the key and context terms, produces feature vectors based on the terms that were found, and uses the classifier to identify sentiment in the document. It is clear that the ConSent algorithm focuses mainly on the tweet content.

However, there are other proposed platforms that use some assistant features such as the emotional state of a tweet's writer and relationships among users. The main goal of such platforms is to cluster users. In Tsagkalidou, Koutsonikola, Vakali, *et al.* [51], the authors propose an emotional aware clustering approach to group tweets based on a model of eight primary emotions developed by Gill, French, Gergle, *et al.* [52]. These emotions are acceptance, fear, anger, joy, anticipation, sadness, disgust, and surprise. Their proposed model relies on using an existing dictionary (WordNet). However, Mudinas, Zhang, and Levene [53] point out that WordNet is not a very reliable source since it introduces too much noise.

Going beyond the content of a document benefits sentiment classification because it provides the classifier with a deeper understanding of the task. To investigate its usefulness, Davidov, Tsur, and Rappoport [54] develop a multidimensional framework in order to analyze the spatial, temporal, and sentiment aspects of tweets discussing the same topic in an online social network. The topic mentioned in the paper is the Mediterranean refugees' crisis. The authors point out that the combination of the sentiment aspects with the temporal and spatial dimension is an added value that allows them to infer interesting insights about the topic.

2.4. Learning Text Representation

2.4.1. Learning Text Representation in DL

Recently, DL approaches have been used in various text classification tasks such as sentiment analysis [55]. The objective of these approaches is to learn the continuous representations of text in different levels of representation: characters, words, phrases, sentences, and documents levels. In other words, an algorithm requires to learn mapping the words, phrases or sentences into a continuous vector. Exiting DL approaches to sentiment classification in social media are mainly based on characters and words levels [56].

2.4.2. Word Level Representation (Word Embedding)

The task of the word level representation is to map each word to appropriate distributional feature representation [57]. A straightforward approach is to encode a token (word) as a one-hot vector with only one dimension as 1 and zeros everywhere else. The size of the vocabulary is used as the size of the dimension [58]. The key issue with the one-hot vector representation is that it fails to capture the semantic relations between different words [56]. To overcome this issue, another representation is proposed by using a low-dimensional dense vector, a.k.a. word embedding, which can encode the semantic meaning of words by mapping each word to a continuous low-dimensional vector[59].

There are two basic approaches to creating the low dimensional dense vector. One is the count based approach and the other one is the prediction based approaches [60]. Both approaches are based on the distributional hypothesis which says: words occur in similar context tends to have similar meanings [61].

In the count based method, a word-context co-occurrence statistic matrix is generated. Then, a matrix factorization is carried out to obtain the final word embedding [62]. Point-wise Mutual Information (PMI), positive point-wise mutual information (PPMI), and the log of co-occurrences are some examples of features used to create the matrix [60]. Several works are built based on this method. For example, In [63], a word representation method called Global Vectors for Word Representation (GloVe) is proposed by utilizing a word-context co-occurrence statistic matrix. Another representation that takes advantage of the factorization matrix is Singular Value Decomposition (SVD) [64]. SVD factorizes the matrix into the product of three matrices: two orthonormal matrices and a diagonal matrix of eigenvalues in decreasing order.

In the prediction based method, the context given the target word is predicted by maximizing the conditional probability of the context words given the target or vice versa [62]. A well-known example of this method is Google word2vec embeddings. The embeddings are trained on 100 billion words from Google News. They are of dimensionality 300 and publicly available [65]. The word2vec embeddings are proposed by Mikolov, Sutskever, Chen, *et al.* [59] at Google.

In fact, the word2vec embeddings are created by two distinct models: the skip-gram model and Continuous Bag-of-Words model (CBOW). Both models are shallow neural models [43]. The skip-gram model predicts surrounding words given the embeddings of the current word. In the CBOW model, the word representation is learned by predicting the word in the middle of a symmetric window based on the embeddings of its context words in the window [56]. These two models are trained either with or without negative sampling; however, Mikolov, Sutskever, Chen, *et al.* [59] recommend using the skip-gram model with negative sampling.

Beside these word-level representation methods, there are also some word embedding methods specifically designed for social media sentiment analysis. Tang, Wei, Yang, *et al.* [41] report that the traditional word embedding is not effective enough for Twitter sentiment classification since it fails to distinguish words with similar context but opposite sentiment polarity. To address this shortcoming, the authors propose a Sentiment-Specific Word Embedding (SSWE). It encodes sentiment information of each word in a continuous representation by using three neural networks. Similar to Tang, Wei, Yang, *et al.* [41], Maas, Daly, Pham, *et al.* [66] propose to learn sentiment-specific word embeddings with probabilistic topic modeling.

2.4.3. Character Level Embeddings

While word level representations are meant to extract syntactic and semantic information, character level representations extract morphological and shape information [67]. In Vosoughi, Vijayaraghavan, and Roy [68], Tweet2Vec, a generating general-purpose vector representation of tweets, is proposed. A character level CNN-LSTM encoder-decoder model is used to learn tweet embeddings. The dataset used to learn the model contains 3 million, randomly selected English language tweets. The proposed model was evaluated on two tasks: tweet semantic similarity and tweet sentiment categorization. The authors point out that Tweet2Vec is a language-independent vector representation; that is, the proposed model can be used to learn tweet embeddings for different languages.

A more sophisticated method to represent a tweet at different levels is proposed in [67]. The method is based on a deep neural network architecture. Specifically, they built a model called Character to Sentence Convolutional Neural Network (CharSCNN), which consists of two convolutional layers to capture relevant features from words and sentences of any length. To conduct sentiment analysis, the model jointly uses three levels of representations: character, word and sentence. The model is evaluated using two datasets: the Stanford Sentiment Tree-bank (SSTb) and the Stanford Twitter Sentiment corpus (STS). The experimental results of the CharSCNN model show that it outperforms the baselines in prediction accuracy. It obtains a sentiment prediction accuracy of 86.4%. The main reason, according to the authors, is that character-level information has a greater impact on Twitter datasets (as in STS).

Several researchers have utilized the character level representations to create models that can be generalized across multiple languages without relying on machine translation approaches [58]. For example, Wehrmann, Becker, Cagnini, *et al.* [69] propose a character-based neural network model for sentiment classification in multilingual scenarios called Conv-Char-S, which is not based on a machine translation approach. The Conv-Char-S processes the sentences at character-level. The model architecture includes three layers: a convolutional layer, a max-pooling-over-time layer, and fully-connected layer. The convolutional layer processes the input at character level to generate a sentence-level embedding. In the max-pooling-over-time layer, the dimensionality of the tensors is reduced. The last layer is responsible for producing class scores. A Twitter dataset that consists of 13 languages, provided by [70], is used to evaluate the model. The experimental results show that the Conv-Char-S model outperforms other deep learning and traditional approaches.

In Zhang, Zhang, and Chan [58], the authors propose a fully language-independent character-based CNN model for the classification of tweets in multiple languages and mixed languages, named Unicode character Convolutional Neural Networks (UniCNN). This model does not require any type of language detection. It is done by converting a sequence of characters of a tweet into a sequence of numerical UTF-8 codes. Then, a character-based CNN classifier is used to learn the representation (embedding). The model is evaluated on several Twitter classification tasks including sentiment classification. Similar to [69], the dataset used to train and test the model performance is provided by [70]. Several baselines are used for comparison with the proposed model. The UniCNN performs better than the baselines.

CHAPTER 3

TWITTER SENTIMENT ANALYSIS WITH A DEEP NEURAL NETWORK: AN ENHANCED APPROACH USING USER BEHAVIORAL INFORMATION[1]

3.1. Introduction

In contrast to standard texts with many words that help gather sufficient statistics, the texts in social media, especially Twitter, only consist of a limited number of characters. Moreover, when a user posts a message (tweet), it may have new abbreviations or acronyms that appear infrequently in conventional text documents. Therefore, applying traditional methods to such an environment will not yield acceptable performance.

To address this issue, we propose a neural network based model that goes beyond the content of a given document (tweet). It takes into account, besides a given tweet, the behavior of the user who wrote that tweet. It has been demonstrated by the experiments conducted in this paper that this approach increases the performance and the accuracy of sentiment classification tasks on social media.

The main motivation behind our approach is the intuition that the association between the written document (tweet) and the behavioral information of the user who wrote that document can provide a model with useful indicators to boost its performance.

Some examples of papers that show the efficiency of this approach are [71], [72], [73] and [74]. However, most of these are related to emotion; they try to classify the users themselves from a psychological point of view. For example, the authors in [73] develop a tool to classify users into positive and negative groups based on the network density and the

[1]The work in this chapter is published in [1] [3].

degree of social activity. Their purpose is to understand the relationship between positive and negative users, as well as, how emotion and mood can affect both a user's behavior and their interaction with other users. They conclude that there is a strong relationship between the users' emotions and the way users choose their friends. Accordingly, we believe that such a relationship also exists between users and their posts, which has been shown by the experiments conducted in this paper (see the experiment results in Section 4).

The main research question here can be posed as: is there a relationship between the user's behavior and his/her posts? And if such a relationship exists, can it be used to enhance sentiment analysis performance?

To address this question, we propose a neural network based model that classifies tweets into positive and negative categories based on a proposed set of features that enhance the classification performance. These features help the model to understand a user's behavior.

The behavior of a user can be identified by knowing two aspects of the user. The first aspect involves the personality traits, such as his/her general attitude; is it positive, negative or neutral? The second aspect regards the social activities of users, which can be revealed through the users' relationships and communication. A list of the features extracted and calculated in our experiments is explained in detail in subsequent sections .

The remainder of this paper is structured as follows. We outline the background for our work in Section 2. Next, Section 3 presents a detailed description of our proposed system. Section 4 shows the experiments and results achieved with the proposed system. Finally, we give conclusions and discuss future work in Section 5.

3.2. Related Work

Most existing sentiment classification methods for social media focus on document-level classification; for instance, the ConSent "Context-based Sentiment Analysis" algorithm is created by [50]. It has two phases, learning and detection. A set of key terms and context terms from a training set are produced in the learning phase. The detection phase is where the classification task takes place. It scans all tweets searching for the key and context terms, and uses the classifier to identify sentiments in the document. As it is clear, ConSent algorithm focuses mainly on the tweet content.

However, there are other proposed platforms use some assistance features such as the emotional state of a tweet's writer and relationships among users. The main goal of such platform is clustering users. In [51], the authors propose an emotional aware clustering approach to group tweets based on eight primary emotions. These emotions are acceptance, fear, anger, joy, anticipation, sadness, disgust and surprise. Their proposed model relies on using an existing dictionary (WordNet). However, [53] points out that WordNet is not a very reliable source since it introduces too much noise.

Going beyond the content of a document benefits sentiment classification because it is providing the classifier with a deep understanding of the task. To investigate its usefulness, [54] develops a multidimensional framework in order to analyze the spatial, temporal and sentiment aspects of tweets discussed the same topic. The authors point out that the combination of the sentiment aspects with the temporal and spatial dimension allow them to infer interesting insights about topics.

3.3. Methodology

3.3.1. Deep Learning Architecture

Our proposed model for sentiment analysis consists of a convolutional neural network. The system architecture is presented in Fig. 3.1. The model is implemented using the Weka[2]

[2] http://www.cs.waikato.ac.nz/ml/weka/

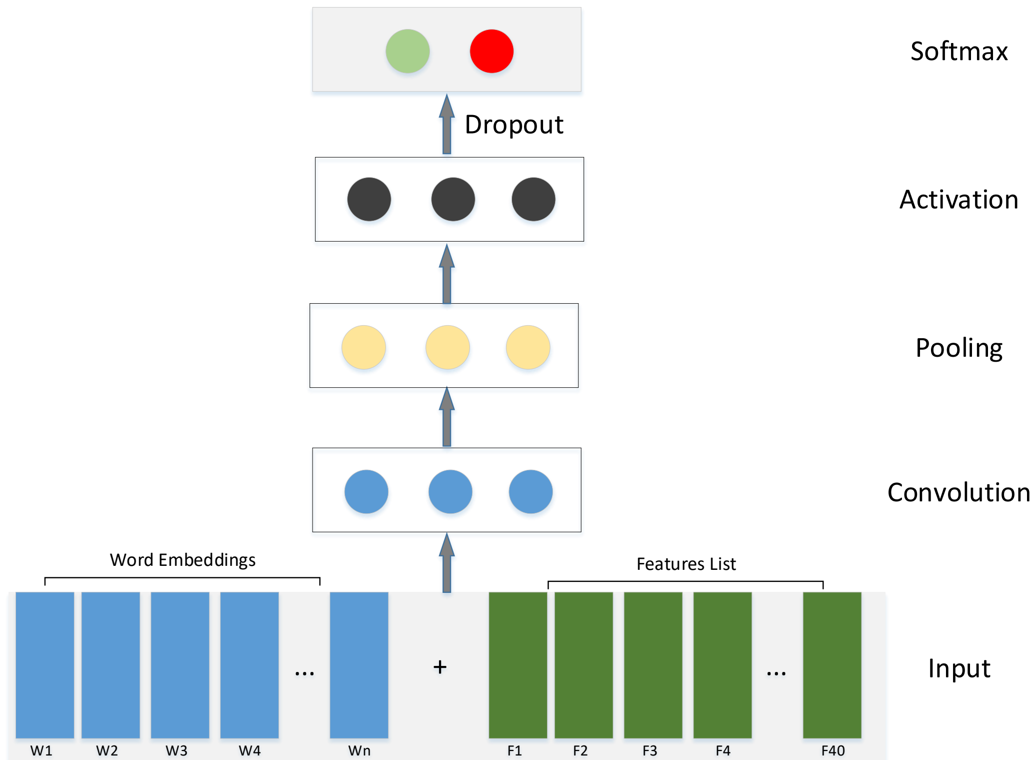


Figure 3.1: The architecture of the proposed model

library.

The main components of the network are the input, convolution, pooling, activation, and softmax layers.

The input layer consists of word embeddings and a list of features. The word embedding may be randomly initialized or pre-trained. For the purpose of this work, we utilize the publicly available word2vec embeddings [59]. We pre-train the 200-dimensional word embeddings on each dataset. For that purpose, each tweet is tokenized and each generated token is mapped to a distributional feature representation known as the word embedding. Moreover, the features that describe the writer of the tweet are appended to the generated vector and then fed into the next layer. That is the convolutional layer; its main goal is to extract patterns.

In order to allow the learning of non-linear decision boundaries, a non-linear activation function is located at the activation layer. There are a number of common choices of activation functions used with neural networks; for example, sigmoid (or logistic), hyperbolic tangent (tanh), and rectified linear (ReLU) functions.

In our model, we use ReLU because it is pointed out by several studies, such as in [43], [75], that ReLU speeds up the training and produces more accurate results than other activation functions.

The softmax layer is an activation function whose output is the probability distribution over labels. In this 2-classes task, given an input representation vector v, a softmax operation is computed as follows:

$$softmax(v, L) = \frac{exp(v)}{\sum_{l=1}^{L} exp(\hat{v})} \tag{3.1}$$

where L is the number of sentiment classes and $\hat{v}$ is the predicted probability of sentiment class l. The reason for adding the dropout layer before the softmax layer is to prevent overfitting.

3.3.2. Features

This section provides the details of all 40 features used to train the proposed model. Table 3.1 lists all features used to create the proposed model.

Number of a User's Tweets. This feature represents the number of a user's tweets retrieved from his or her timeline via the Twitter API. Twitter allows to return a collection of the most recent tweets posted by the user. It can only return up to 3,200 tweets of the user.

Table 3.1: List of features used in this paper

Feature ID	Feature Description
F1	Number of a User's Tweets
{F2, F3, F4}	Number of {Positive, Negative, Neutral} Tweets Posted by a User.
{F5, F6, F7}	The Probability of a User Having a {Positive, Negative, Neutral} Attitude.
{F8,F9,F10}	Average Number of {Positive, Negative, Neutral} Tweets Posted by a User.
F11	Number of Followers.
F12	Number of Friends.
F13	Verified.
F14, F15	Number of Adjectives, and Their Average.
F16, F17	Number of Nouns, and Their Average.
F18, F19	Number of Adverbs, and Their Average.
F20, F21	Number of Verbs, and Their Average.
F22, F23	Number of Hashtags, and Their Average.
F24, F25	Number of Mentions, and Their Average.
F26, F27	Number of URLs, and Their Average.
F28, F29	Number of Emoticons, and Their Average.
F30, F31	Number of Question Marks, and Their Average.
F32, F33	Number of Exclamation Marks, and Their Average.
F34, F35	Number of Words per Tweet, and Their Average.
F36, F73	Number of Positive Words in Bing Liu Lexicon, and Their Average.
F38, F39	Number of Negative Words in Bing Liu Lexicon and Their Average.
F40	Number of Retweets.

Number of Positive, Negative and Neutral Tweets Posted by a User. These features aim to measure the general attitude of a user. According to Lima and Castro [71], one of the ways that one can predict a user personality is through his or her words. Therefore, in our experiments we use the frequency of tweets based on three aspects, positive, negative, and neutral.

To achieve this, we need to collect tweets from each user who appear in the dataset, which in turn provides us with a huge number of unlabeled tweets. In some datasets, we got about 700,000 tweets posted by more than 3,500 users.

However, labeling all these tweets manually is not an easy task and will be very time consuming and costly. One of the solutions adopted by researchers is the SentiStrength algorithm, which is considered a state-of-the-art sentiment analysis system. It was developed in Thelwall, Buckley, Paltoglou, *et al.* [76], and uses a scoring range from -5 (very negative) to +5 (very positive). To explain in more detail, for each text, SentiStrength outputs two integers: 1 to 5 for positive sentiment strength and a separate score of 1 to 5 for negative sentiment strength. For example, a text with a score of 3, 5 would contain moderate positive sentiment and strong negative sentiment. A neutral text would be coded as 1, 1.

Accordingly, we create an application that receives SentiStrength's output and is interpreted as follows. A tweet is considered positive if its positive sentiment strength is higher than both 1 and its negative sentiment strength; otherwise it is considered a negative tweet. In case a tweet gets 1 as a score for its positive and negative sentiment strength, it is considered a neutral tweet. A tweet could be neutral if it has the same scores in positive and negative sentiment strength, for instance, a tweet with scores 3 for positive and 3 for negative strength. This can be formulated as:

$$SS(t_i) = \begin{cases} Positive & if\ SS(t_i)_{pos} > 1\ and\ SS(t_i)_{pos} > SS(t_i)_{neg} \\ Negative & if\ SS(t_i)_{neg} > 1\ and\ SS(t_i)_{neg} > SS(t_i)_{pos} \\ Neutral & if\ SS(t_i)_{pos} = SS(t_i)_{neg} \end{cases} \tag{3.2}$$

For each user $u \in U = \{u_1, u_2, ..., u_n\}$ where U is a user set and n is the number of users extracted from the dataset, the application calculates $SS(t_i)$. Here $SS(t_i)$ is a function that takes as input a tweet $t_i \in T_u$ (where T_u has all tweets retrieved from user u's timeline), and returns its label based on the SentiStrength algorithm. $SS(t_i)_{pos}$ and $SS(t_i)_{neg}$ are scores of the positive and negative sentiment strengths, respectively.

For our implementation we use Eq.(3.2) as follows:

$$SS(t_i) = \begin{cases} Positive & if\ SS(t_i)_{pos} > 1\ and\ SS(t_i)_{pos} > SS(t_i)_{neg} \\ Negative & if\ SS(t_i)_{neg} > 1\ and\ SS(t_i)_{neg} > SS(t_i)_{pos} \\ Both & if\ SS(t_i)_{pos} = SS(t_i)_{neg} \neq 1 \\ Neutral & Otherwise \end{cases} \tag{3.3}$$

The change implies that, in case of having a tweet scored with equal values of positive and negative sentiment strengths that are greater than 1, then the tweet will be counted twice, one for each label.

The Probability of User having a Positive, Negative, or Neutral Attitude. This feature can measure the attitude of a user as observed across a longer period than individual conversations or tweets on datasets. According to Lima and Castro [72], there is a relationship between a user's temperament and posts written by that user. This feature is calculated as follows:

$$P_U(l_i, T_u) = \frac{|\{t \in T_u : t = l_i\}|}{|T_u|} \tag{3.4}$$

where $P_U(l_i, T_u)$ is a function that computes the probability of a user $u \in U$ being labeled $l_i \in \{Positive, Negative, Neutral\}$; $|T_u|$ is the number of all tweets in the set T_u retrieved from user u's timeline, and $|\{t \in T_u : t = l_i\}|$ is the number of tweets labeled l_i that are in the user's timeline.

Number of Followers and Friends. These are Twitter-specific features. After collecting tweets in the dataset, we extract a list of unique users who have at least one post (tweet) in the dataset. Then, some information about these users is collected from their public profile by the Twitter API. Part of this collected information is the number of followers and friends a user has. Based on some work in the literature such as [17], it was found that the number of followers and friends of a user are considered among the "good" features that indicate the degree of sociality of that user.

Verified. This indicates whether or not a user's account is verified by Twitter. This feature takes two values: True or False. True means this account is verified and false otherwise. It is used to establish the authenticity of identities on Twitter. In general, tweets posted on verified accounts are considered high-quality sources of information.

Number of Verbs, Adjectives, Nouns, and Adverbs. All these features are calculated based on all tweets retrieved from a user's timeline. These features are good indicators of what type of language is used. For example, if a user includes many adjectives when he or she posts tweets, there is a high possibility that opinions are expressed in these tweets. The importance of such features is recognized as being good indicators of opinion polarity [77]. Moreover, some papers such as [78], [79] point out that nouns, adjectives, verbs, and adverbs are the most opinionated lexical types in texts.

Number of Hashtags. Hashtags are used to indicate the relevance of a tweet to a certain topic. They are created by users and can be used to retrieve all tweets posted with the same hashtag. To create a hashtag in Twitter, the # character should be inserted before the name of the topic. Like the number of followers and friends, hashtags are considered as one of the Twitter-specific features.

There are two reasons for using hashtags in the feature set. First, as discussed inLima and Castro [72] and Lima and Castro [71], they give a good indicator of a user providing his or her thoughts and opinions about the topic of the hashtag. The second reason is that hashtags could be used to detect spam tweets; indeed an experiment conducted in [80] shows that a hashtag is considered to be spam if its tweet frequency is high.

Number of Mentions. Mentions in Twitter are used to identify the user-recipients of a tweet. Moreover, users can also explicitly mention other users to draw their attention by adding an expression of the form @Username in their tweets.

This feature counts the number of mentions that a user accumulated through all tweets retrieved from his or her timeline. It is used in this paper because, as mentioned in [81], it indicates that the author of the tweet wants to have a conversation with or to show opinions toward who has been mentioned.

Number of Positive and Negative Words in Bing Liu Lexicon. In this paper, lexicon features are utilized by extracting them from publicly available lexicons; these prove to be one of the most powerful types of features [82]. Social media data, especially tweets, have a style of language that is quite different from other text data. Therefore, the Bing Liu lexicon is used, which is especially tailored to handle social media data. The Bing Liu lexicon is a list of positive and negative words. It has around 6,800 words [83].

The lexicon features are utilized as follows. The frequency of positive and negative words in each retrieved tweet from a user's timeline is calculated as well as the average number of occurrences of each label. These are then appended to the feature vector.

Number of Question Marks. The total of question mark occurrences are calculated. This feature is calculated as follows:

$$QM(T_u) = \sum_{i=1}^{|T_u|} |\{t_i \in T_u : HasQM(t_i) \neq 0\}| \tag{3.5}$$

$QM(T_u)$ is an aggregation function that takes as input T_u, which contains all tweets of user u in the user set U, and returns the total number of question marks appearing; $|T_u|$ is the number of tweets of user u, and $HasQM(t_i)$ is a function that is applied to tweet t_i and returns the number of question marks in t_i. Furthermore, the average number of question marks of user u is computed.

The reason for considering the question marks as a feature and adding it to the feature vector is that question marks can be a strong sign of subjective (opinionated) tweets [84].

Number of Exclamation Marks. This gives the total exclamation mark occurrences in the tweets of each user, and is calculated in the same way as the question marks feature (see Eq. (3.5)).

Number of Words per Tweet. This is used to remove a tweet based on its number of words. For example, if we would like to conduct an experiment that takes into account all tweets of each user $u \in U$ that are longer than four words, all we need is to apply this condition to the algorithm.

Number of Emoticons. An emoticon is a pictorial representation of a facial expression using the characters available on the standard keyboard. Its presence in a tweet is a good indicator of a writer's emotional state [85]. As pointed out in [86], the presence of an emoticon almost always conveys the underlying sentiment. Therefore, this feature is added to the feature vector and is calculated as follows:

$$E(T_u) = \sum_{i=1}^{|T_u|} |\{t_i \in T_u : HasE(t_i) \neq 0\}| \tag{3.6}$$

$E(T_u)$ is an aggregation function that takes as input T_u, which contains all tweets of user u in the user set U, and returns the total number of emoticons appearing; $|T_u|$ is the number of tweets of u, and $HasE(t_i)$ is a function that is applied to tweet t_i and returns the number of emoticons in t_i. Twitter NLP's[3] tokenizer is utilized to detect emoticons. In addition, the average number of question marks of user u is computed.

Number of URLs. This counts the total URL occurrences in tweets of each user, and is calculated in the same way as the emoticons feature (see Eq. (3.6)).

3.4. Experiments and Results

In this section we introduce the experimental setting and report empirical results on the tasks of sentiment classification.

3.4.1. Experimental Settings

We evaluate the effectiveness of our proposed model on two datasets (see Section 4.2 for the description of the datasets). Fig. 3.2 shows all the steps we follow to conduct the experiments.

The following is a brief explanation of the main components in the steps of Fig.3.2.

Pre-Process Step. In this step the normalizing, tokenizing, and part-of-speech tagging of all tweets is conducted. Tokens are stemmed (traced back to a common base form) using the Snowball stemmers[4] . Furthermore, all URLs and usernames are removed and replaced with the keywords HTTP and USER, respectively.

[3]http://www.cs.cmu.edu/ ark/TweetNLP/
[4]https://weka.wikispaces.com/Stemmers

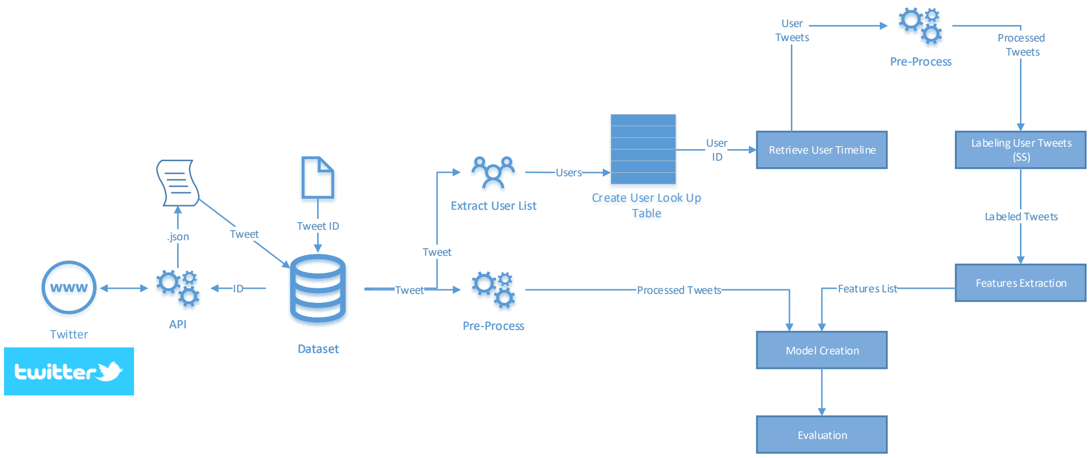


Figure 3.2: Steps for conducting the experiments

Extract User List Step. This step extracts the names of users who have at least one tweet in the dataset.

Create User Lookup Table. Here, a table is created that contains the users' names and their IDs, and is used to retrieve their tweets from their timeline.

Retrieve User Timeline Step. This retrieves the tweets from a user's timeline via his/her ID. The ID is sent to Twitter by using its API.

Label the User Tweets by SentiStrength Step. Here the retrieved tweets from the previous step are labeled by the SentiStrength algorithm.

Features Extraction Step. This step extracts all features mentioned above for each user.

Model Creation Step[5] Six classifiers (CNN, SVM, NB, J48, KNN and LSTM) are created in this step— for descriptions see Section 3.4.4.

[5]All the models and the source code of these algorithms used in this paper are freely available upon request.

Evaluation Step. Here, the results of all models are evaluated. The results are evaluated based on the most widely used performance measures in the classification task: precision, recall, F1, and accuracy. The following equations are used for the computation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.7}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.8}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.9}$$

$$F1 = 2\frac{Precision * Recal}{Precision + Recal} \tag{3.10}$$

where TP (True Positives) represents the number of positive tweets that were correctly predicted as positive and FP (False Positives) represents the number of negative tweets that were incorrectly predicted as positive; TN (True Negatives) and FN (False Negatives) have a corresponding meaning for the negative class.

3.4.2. Datasets

We train our model on two datasets, which are Twitter datasets published by SemEval-2016[6] . The first dataset (SemEval-2016_1) consists of 3,694 tweets and the second one (SemEval-2016_2) consists of 1,122 tweets. They are annotated manually.

The two datasets have tweet IDs along with their annotation, with positive and negative labels. They need to be downloaded by using the Twitter API. Table 3.2 shows statistical

[6]http://alt.qcri.org/semeval2016/

information about the two Twitter datasets used in our experiments.

Table 3.2: Label distribution for two Twitter datasets

Dataset	# of Tweets	Positive	Negative
SemEval-2016_1	3,694	3,054	643
SemEval-2016_2	1,122	832	290

For each dataset, a list of users who posted the tweets is created. Then, tweets in the timeline of each user in the list along with their public profiles data are retrieved. This provides us with a significant number of unlabeled tweets; therefore, as mentioned above, the SentiStrength algorithm is used. Information about the number of users, total number of tweets and the average number of tweets per user are shown in Table 3.3.

Table 3.3: User distribution in datasets

	Dataset	
	SemEval-2016_1	SemEval-2016_2
Number of Users	3,536	2,198
Total Number of Tweets	774,244	491,902
Average Number of Tweets per User	218.96	223.80

3.4.3. Results

All experimental results are reported using 10-fold cross-validation, which is carried out by partitioning the dataset into 10 subsets, performing the classification on one subset (the training set), and validating the model on the remaining (10 - 1) subsets, named the validation set or testing set. This operation is repeated 10 times for every subset. The validation results are averaged over the 10 iterations.

3.4.4. Baseline Methods

We compare the performance of our approach with the following baselines.

Naive Bayes (NB): Naive Bayes works well on text categorization [85]. It is a probabilistic algorithm that uses Bayes' rule. Eq. (3.11) represents Bayes' rule,

$$P(cd) = \frac{P(c)P(d|c)}{P(d)} \tag{3.11}$$

where c is the class and d is the document (tweet) under consideration.

Support Vector Machine (SVM): Another popular classification technique relies on the Support Vector Machine. According to Pang, Lee, Rd, *et al.* [87], SVM has been shown to be highly effective at text categorization.

J48 (Decision Tree): J48 is an algorithm to generate a decision tree, proposed by Quinlan [88]. J48 is the enhanced version of the C4.5 algorithm. It starts by creating a binary tree from labeled training data. Each data attribute can be used to make a decision by dividing the data into smaller subsets. This approach is most useful for task classification [89]. Once the tree is built, it can be used to construct the classification model.

K-Nearest Neighbors algorithm (KNN): This algorithm is commonly applied for classification in pattern recognition and machine learning [90], and relies on the assumption that samples placed close to each other are likely to belong to the same class [56]. Therefore, a given text is classified as follows: the KNN algorithm searches for the k nearest neighbors among labeled training instances based on some similarity measure, and lists those k neighbors based on their similarity scores; the label or class of the k nearest neighbors are used to determine the class of the given text [90]. In this paper, the number of nearest neighbors is set to two (i.e., $k = 2$).

Long Short-Term Memory (LSTM): The LSTM models have achieved impressive performance in the sentiment classification task [91]. They have the ability to handle long-range dependencies [68].

3.4.5. Results and Analysis

Here we present a comparative performance evaluation of each model in terms of correctly predicting polarity. The results for precision, recall, accuracy, and F1 are obtained for the six methods. Since many features are included in our experiments, we grouped them into sets. In addition, using sets provides us with more clarity of which set has more influence on the performance of the model. The sets are shown in Table 3.4.

Table 3.4: The feature sets in different combinations

Feature Set ID	Features Used
Set No. 1	F5, F6, and F7.
Set No. 2	F1 to F13.
Set No. 3	F2, F3, and F4.
Set No. 4	Word Embedding, F5, F6, and F7.
Set No. 5	Word Embedding.
Set No. 6	All features (F1 to F40).
Set No. 7	All features (F1 to F40) + Word Embedding.

Table 3.5 shows the accuracy of all classification methods. Generally, deep learning classifiers (CNN and LSTM) keep their performance at a steady trend unlike other classifiers. The best accuracy was 88.71% for CNN on set No. 4, followed by LSTM with 88.13% on the same set, and the lowest value was 48.31% for NB on set No. 2.

The effect of the unbalanced dataset[7] is observed clearly on NB and SVN. One possible reason the unbalanced dataset does not have a significant effect on the CNN classifier performance is the way the weights are calculated in this type of classifier.

[7]The majority class in our datasets is the positive class (see Table 3.2).

Table 3.5: Accuracy rates of CNN, SVM, NB, J48, KNN, and LSTM

Set No.	1	2	3	4	5	6	7
CNN	82.63%	82.85%	82.93%	88.71%	87.08%	83.90%	88.46%
SVM	82.58%	82.58%	82.58%	86.84%	86.67%	82.58%	86.75%
NB	82.63%	48.31%	83.62%	85.16%	85.16%	59.38%	70.40%
J48	82.58%	82.25%	82.22%	87.25%	87.61%	82.33%	85.44%
KNN	81.78%	81.56%	80.96%	82.72%	82.66%	82.14%	82.83%
LSTM	83.63%	77.14%	84.29%	88.13%	84.83%	83.74%	86.48%

It can be observed that the best classification performance accuracy is achieved by CNN, LSTM, and SVM. However, as shown in Fig. 3.3 and 3.4, CNN has higher precision and recall than SVM across all sets. This means that using accuracy as the only measurement for classifier performance is not sufficient. Although the number of negative instances in both datasets is lower than that of the positive ones, the proposed CNN model is able to give us a good and stable performance.

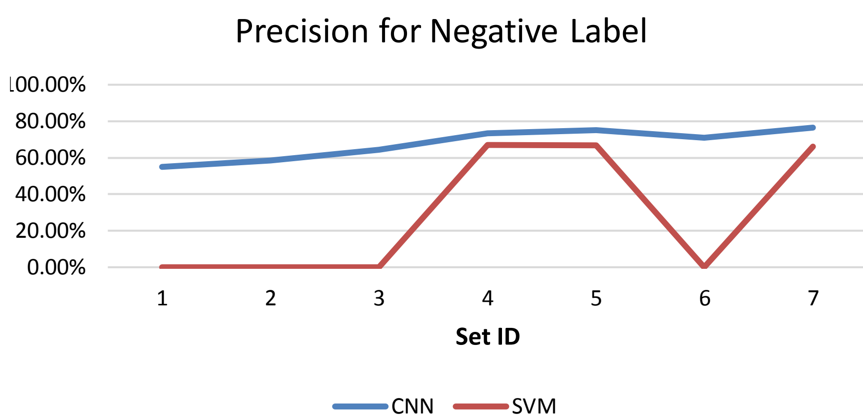


Figure 3.3: Precision for the negative label of CNN and SVM

The poor performance of NB is not surprising since it relies on the assumption of conditional independence among the features, which is clearly not true here. All features used in the experiments have some degree of dependency among them.

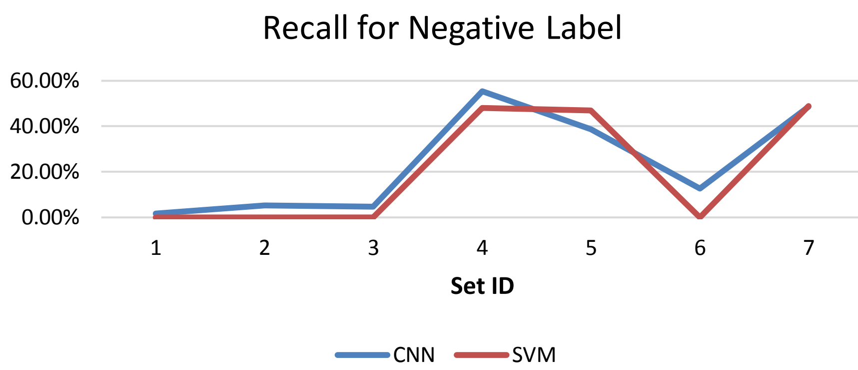


Figure 3.4: Recall for the negative label of CNN and SVM.

Based on the sets, the best accuracy was obtained by sets No. 4 and 7. This supports our motivation to utilize information beyond the content of a given tweet. To investigate the results of these two sets for sentiment analysis, F1 along with precision and recall scores of the six models in both sets 4 and 7 are listed in Table 3.6. These metrics provide us with more insight into the functionality of classifiers than the accuracy metric.

Table 3.6: . F1, precision, and recall scores of the six models in sets No. 4 and 7

	Precision		Recall		F1	
	Set No.		Set No.		Set No.	
	4	7	4	7	4	7
CNN	0.88	0.88	0.89	0.89	0.88	0.87
SVM	0.86	0.86	0.87	0.87	0.86	0.86
NB	0.85	0.83	0.85	0.70	0.85	0.74
J48	0.86	0.84	0.87	0.85	0.86	0.85
KNN	0.81	0.84	0.83	0.83	0.75	0.75
LSTM	0.87	0.86	0.88	0.86	0.86	0.83

By examining the results reported in Table 3.6, we can see that the high scores for precision and recall reinforce the observation of the high accuracy scores for sets No. 4 and 7 in Table 3.5. In Table 3.6, the highest F1 scores are obtained by the CNN model (0.88 in set No. 4 and .87 in set No.7). It is also worth noting that the CNN model shows consistent good performance in both sets, followed by the LSTM model. These results confirm that using user behavioral information besides textual content in sentiment analysis with improve the classification accuracy.

3.5. Conclusion

In this paper, we present a sentiment analysis model developed by combining a list of features. We propose the architecture of a Convolutional Neural Network (CNN) that takes into account not only the text (user tweets) but also user behavior. Our evaluation results demonstrate the efficiency of the model in a social media setting.

Our model outperforms the baseline methods in accuracy, recall, precision, and F1. In addition, the proposed model is affected less by unbalanced dataset issues. Moreover, the approach overcomes the issue of needing a large dataset to train deep learning models such as CNN and LSTM.

This work suggests interesting directions for future work. For example, it would be interesting to investigate the contributions of the produced list of features for non-binary sentiment classification tasks. In future work, we also plan to explore other neural network based learning models, such as Recurrent Neural Networks (RNN) and gated feedback RNN for sentiment analysis.

CHAPTER 4

EMOTIONAL AWARENESS BASED CLASSIFICATION MODEL FOR TWITTER SENTIMENT ANALYSIS USING A DEEP NEURAL NETWORK[1]

4.1. Introduction

Social media platforms such as Twitter [2] allow people to express and share their thoughts and opinions on all kinds of topics and events. This type of environment is beneficial for marketing [4] and social studies [5].

Whereas standard texts with many words help gather enough statistics, the texts (tweets) in Twitter consist only of a few characters. Moreover, tweets are more likely to have abbreviations or acronyms that appear infrequently in conventional documents. Therefore, applying traditional methods to such settings will not provide us with acceptable performance.

In this paper, we propose an approach based on a deep learning model that is going beyond the content of a target document (tweet). It takes into account, besides the target tweet, the emotional state of a user who wrote it. The experimental results reported in this paper show the effectiveness of the proposed approach in comparison with other models.

The proposed model classifies tweets into three sentiment classes: positive, negative, or neutral. This is done with the assistance of five other deep learning models that extract the emotional state of the tweet's writer from their Twitter timeline. The emotional states these models try to identify are joy, anger, disgust, sadness, and optimism.

The main motivation behind our approach is the intuition that the association between the written document (tweet) and the emotional state of the writer can provide a model

[1]The work in this chapter is published in [2].
[2]http://twitter.com/

with useful indicators to boost its performance.

The remainder of this paper is structured as follows. After giving background information in Section 4.2, Section 4.3 presents a description of our proposed system. Section 4.4 shows the experiments and results. Finally, we conclude our work and discuss future work in Section 4.5.

4.2. Related Work

Most existing sentiment classification methods for social media focus on document level classification, such as the ConSent "Context-based Sentiment Analysis" algorithm by [50]. The algorithm has two phases, learning and detection. A set of key terms and context terms from a training set are produced in the learning phase. The detection phase is where the classification takes place. All tweets are searched for the key and context terms, and the classifier is used to identify sentiments in the document. It is clear that the ConSent algorithm focuses mainly on the tweet content.

However, there are other proposed platforms that use some assisting features such as the emotional state of a tweet's writer and relationships among the users. The main goal of such platform is clustering users. In [51], the authors propose an emotional aware clustering approach to group tweets based on eight primary emotions. These emotions are acceptance, fear, anger, joy, anticipation, sadness, disgust and surprise. Their proposed model relies on using an existing dictionary (WordNet). However, [53] point out that WordNet is not a very reliable source since it introduces too much noise.

Going beyond the content of a document benefits sentiment classification because it provides the classifier with a deep understanding of the task. To investigate its usefulness, [54] develop a multidimensional framework in order to analyze the spatial, temporal and sentiment aspects of tweets discussing the same topic. The authors find that the combination of the sentiment aspects with the temporal and spatial dimensions leads to interesting insights about the topics.

4.3. Proposed Approach

Our approach introduces the power of utilizing the emotional state of users, who wrote the tweets, into the sentiment analysis task. The proposed approach consists of two main steps: (1) the emotional analysis step, in which a user's emotional state is recognized, and (2) the sentiment analysis step, where each document (tweet) is classified.

The emotional analysis step: The task of this step is to classify a given tweet as one (or more) of five emotions (anger, disgust, joy, optimism, and sadness) that best represent the emotional state of the tweet's writer. To perform this task, we first prepare the training datasets. The provided dataset (SemEval-2018) used in this step is labeled in such a way that a tweet could be labeled with one or more emotion categories. In order to treat the task as a binary classification problem, we created five datasets with the same tweets but each with one emotion. For instance, the tweets in the anger dataset are labeled with 1 (emotion exists) or 0 (emotion does not exist) based on the provided dataset.

We built and trained five deep learning models for the emotions. Each model is a Bidirectional RNN network, based on a GRU (Gated Recurrent Unit). We will refer to the overall model as a Bidirectional GRU-Emotional State Model (BiGRU-ESM). The main components include an input layer, an embedding layer, a spatial dropout, a Bi-GRU layer, a max and an average pooling layer, and an output layer. At the input layer, the pre-processed tweet is treated as a sequence of words, $W = (w_1, w_2, \ldots, w_n)$. These are given by a one-hot vector, which has the length of the size of the vocabulary. Because of the inconsistency in the length of tweets, the shorter tweets need to be padded. We normalize the tweet length by zero padding.

At the embedding layer, we apply a pre-trained GloVe[3] word embedding [63] on each word in the vocabulary list. We adopt 200-dimensional GloVe vectors of 27 billion tokens, which are trained on 2 billion tweets from twitter. To reduce overfitting, the spatial dropout layer is applied to the embedding layer. The Bi-GRU layer contains 100 neurons. The temporal information of the tweet sequence is captured in this layer in both directions, forward and reverse. Concurrently, the output of this layer is fed to two pooling layers, max and average. The outputs of both layers are concatenated. Finally, the network output is converted to probabilities by applying a sigmoid activation function.

The sentiment analysis step: The task of this step is: given a tweet and a set of probabilities of the five emotional states of a user who has posted the tweet, predict whether it is of positive, negative, or neutral sentiment.

To perform this task, we built a model similar to the one in the emotional analysis step with an extra input layer. We refer to this model as the Emotional Awareness based Classification Model (EACM). The input layer of the EACM consists of an embedding with a dimension of five. It represents the emotional state of the user inferred by her or his tweet history and classified by our BiGRU-ESM model. This representation of the user emotion provides the EACM model with useful signals that increase its classification accuracy.

4.4. Experiments and Results

4.4.1. Data Sources

The training, development and testing datasets used to train and test the five emotions models are provided by the SemEval-2018 Task 1, Emotion Classification (E-C) subtask [92]. An overview of the datasets is provided in Table 4.1.

[3]https://nlp.stanford.edu/projects/glove/

Table 4.1: The number of tweets for each dataset.

Dataset	No. of tweets
Train	6,838
Dev	886
Test	3,259
Total	**10,983**

Table 4.2: The number of tweets for each SemEval dataset.

Dataset	No. of tweets
Train	6,000
Dev	1,998
Test	20,632
Total	**28,630**

The tweets are annotated by emotional categories. They contain eleven emotions: anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, and trust. Figure 4.1 shows the percentage of tweets in each emotional category. The emotions of joy, anger, disgust, sadness, and optimism get a high percentage of tweets; therefore, these five emotions are considered in this paper and the rest of the emotions are eliminated.

We train and test our main model on the benchmark datasets provided by the SemEval challenge. The datasets are classified based on a three-point scale: positive, negative, or neutral sentiment. However, due to deletion or changed privacy settings, the provided datasets include only Twitter status IDs along with associated labels. We obtained 95% of the entire set of tweets of the SemEval data using the Twitter API. An overview of the data is provided in Table 4.2.

To discover the emotions of a user (who tweets), we need to collect some tweets from the

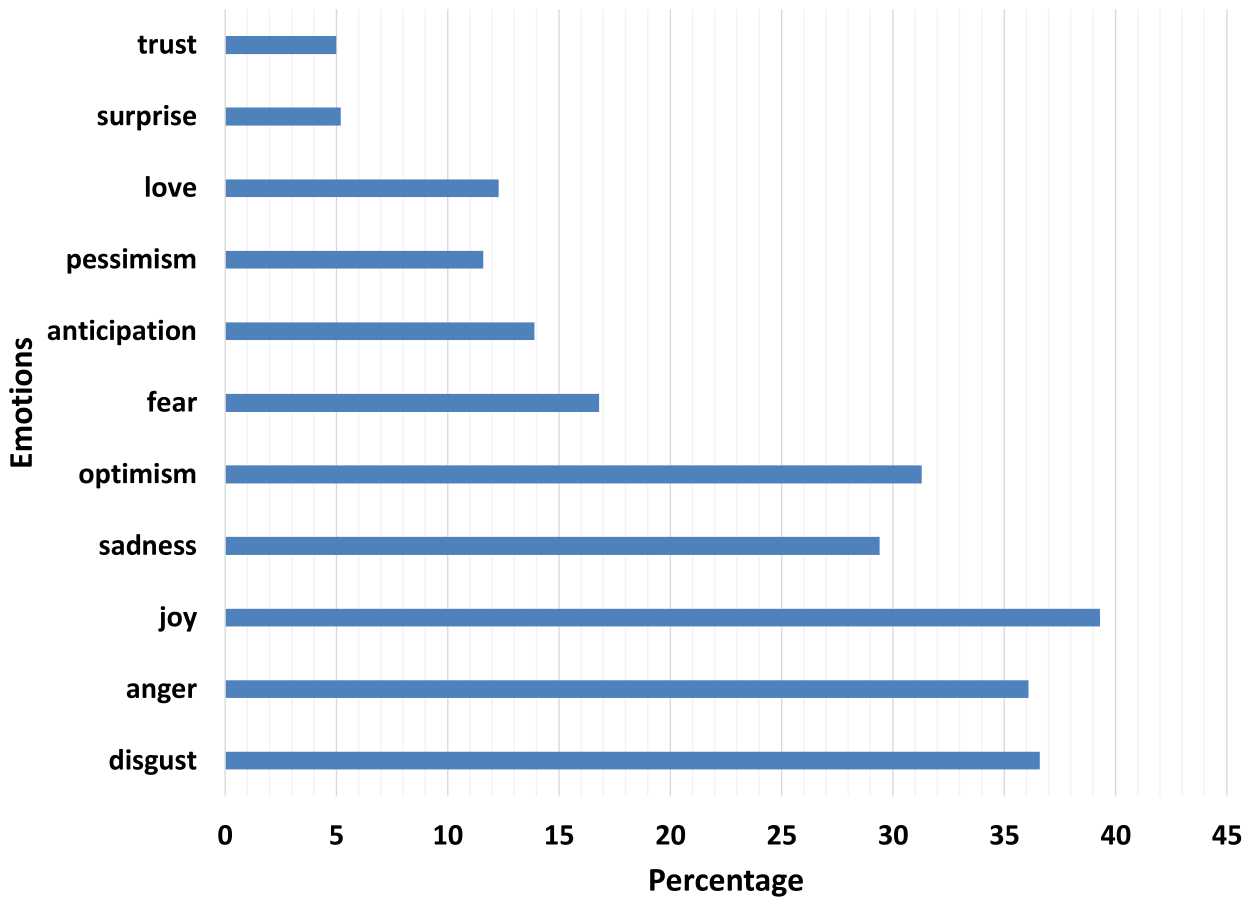


Figure 4.1: Percentage of tweets in each emotional category.

user timeline at different times of the day. However, the provided datasets do not have this information. To achieve the goal we conduct a series of steps. First, user IDs are collected corresponding to their tweet in the SemEval. Second, a user timeline is obtained by the Twitter API using the user ID. Twitter only allows access to a user's most recent 3,240 tweets. Then each user's tweets are kept in a file, the filename of which contains the user ID to allow easy access. Each line is formatted as a tweet followed by its creation date and its language. The total number of users we extracted for the datasets is 5,739. This provides us with 15,628,294 tweets in total.

4.4.2. Preprocessing

The syntax of tweets is commonly not well-structured. Thus we applied the following preprocessing steps to the raw data. Each tweet is converted to lowercase. All Twitter handles (URLs, mentions '@', and hashtags '#') are replaced by placeholders <url>, <user>, and <hashtags>, respectively. Since numbers, punctuation, and special characters do not provide useful sentiment information, they are eliminated from the data. All words with length less than 3 characters are also removed.

Elongated words that have one letter repeated more than two times are tagged and substituted with the same words but where at most two consecutive occurrences are kept. Emoticons are replaced with tags reflecting their meaning. For example, the ':)' emoticon is replaced by a <smile> tag.

Our proposed models have pre-trained embeddings with GloVe algorithm (see Section 4.3). Therefore, for optimal benefit, we also apply the same preprocessing steps mentioned in the GloVe's website, which are used to create GloVe word embeddings on Twitter data. Finally, for tokenizing we utilize the tokenizer provided by the Keras[4] library.

4.4.3. Results

We trained and validated our models on the training and development datasets, and tested their performance on the test dataset (see Section 4.4.1). In order to illustrate the performance of our approach, we compare the results with the following baseline models:

CM: is a variant of our EACM model without utilizing the emotional state of the writers.

Ensemble-CNN-LSTM: is an ensemble of three classifiers with a soft voting method. Two of these classifiers are a convolutional neural network (CNN) and a Long Short Term

[4]https://keras.io/

Memory (LSTM). It is introduced by [93].

CNN-GloVe: is a one-layer convolutional neural network model using pre-trained GloVe embedding [94].

Ensemble-CNN: is a CNN model comprising three CNN models where each one is trained using different embeddings. These are lexical, part-of-speech, and sentiment embeddings. This model is proposed by [95].

The results for accuracy, averaged F1 across the positive, negative and neutral classes, and averaged precision and recall across all three classes are reported and obtained for all models.

Table 4.3: Performance comparison of models. Best scores are in bold.

Model	Accuracy (%)	Average F1 (%)
CM	62.00	58.00
EACM	**70.86**	**69.20**
Ensemble-CNN-LSTM	61.60	61.70
CNN-GloVe	63.50	59.30
Ensemble-CNN	61.70	63.00

Table 4.3 shows the accuracy and the average F1 of all classification methods. It emerges that our EACM model outperforms other baseline models and achieves a significant result with 70.86% in accuracy and 69.20% in average F1. We observe that EACM shows an improvement of 8.86% and 11.2% in accuracy and average F1, respectively, compared to the CM model that does not take user emotional state into account.

Our approach of enriching the model with the user's emotional state improves the classification performance of our proposed EACM model, compared with other models using text only, by nearly 7-9.5% in accuracy and 6-9% in average F1.

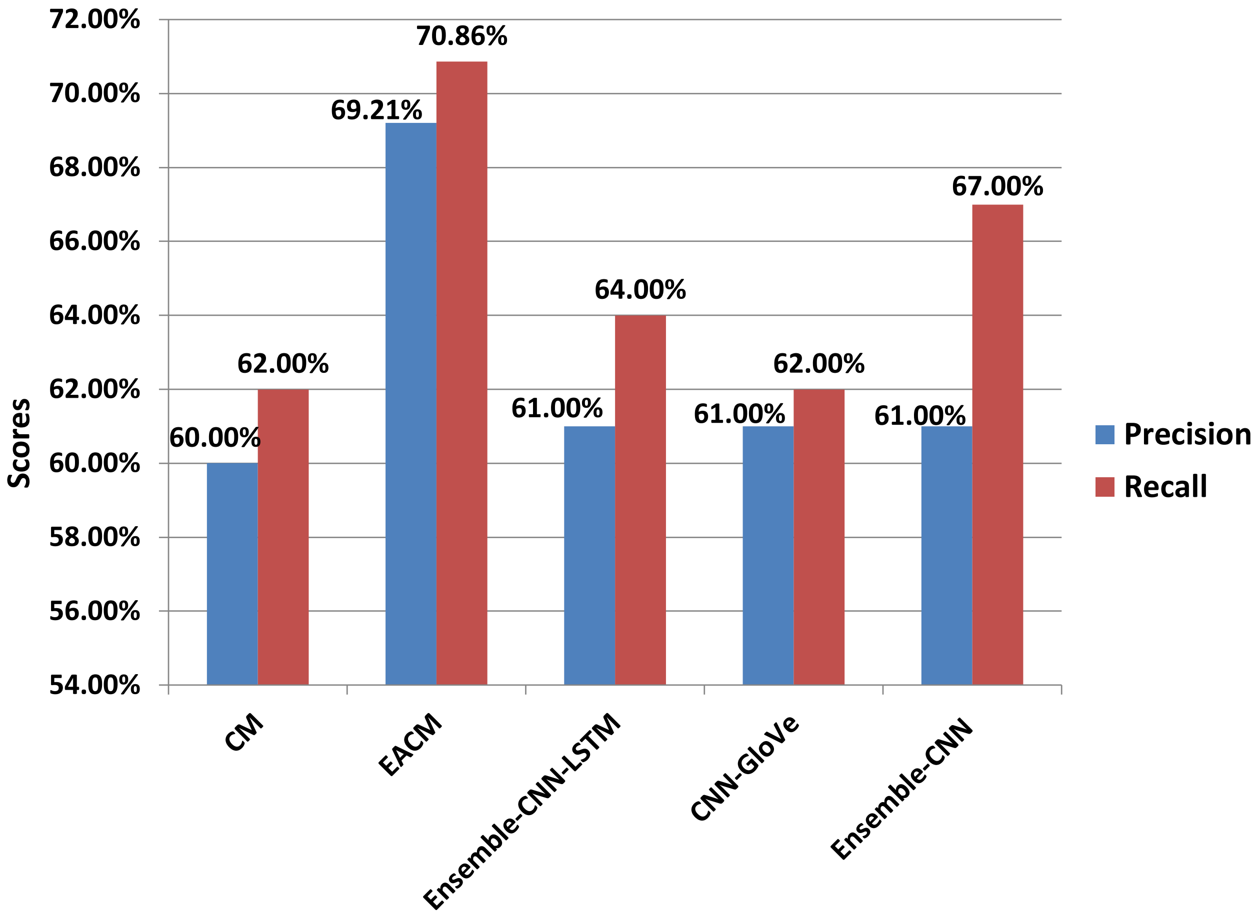


Figure 4.2: Averaged precision and recalls of the models.

The average precision and recalls of the models across the positive, negative and neutral classes are illustrated in Figure 4.2. We notice from Figure 4.2 that the precision and the recall produced by our EACM model are about 69.21% and 70.86%, respectively, which are higher by nearly 8-9% in precision and 4-8.5% in recall compared to other models.

These observations show that using an extra source of information can be beneficial for improving the sentiment analysis task. In our case, the extra source of information is the emotional state of the tweet writer.

4.5. Conclusions

In this paper we illustrate our approach of going beyond the textual content of a tweet, by taking into account not only the text but also the emotional state of the user who wrote that tweet. We propose the Emotional Awareness based Classification Model (EACM) model, using a bidirectional RNN network structure based on a gated recurrent unit. We built and trained five deep neural network models, collectively named BiGRU-ESM (Bidirectional GRU-Emotional State Model), one model for each of the emotion categories of anger, disgust, joy, optimism, and sadness. The task of these five models is to provide the main model (EACM) with the emotional state of the users (writers) as extracted from their tweet history.

The experimental results demonstrate the effectiveness of the proposed approach in comparison with other baseline models that utilize the textual content only. Specifically, the results show a considerable improvement in performance and accuracy of the sentiment classification tasks with the new approach.

This work suggests interesting directions for future work. For example, it would be interesting to investigate extracting the emotional state of the users not only from their timeline in Twitter but also across other social media platforms such as Facebook and Instagram. We also plan to contribute to data sets by collecting and annotating tweets that have emotions, since there is currently a lack of such data.

CHAPTER 5

INCORPORATING SOCIAL RELATIONS BETWEEN USERS To IMPROVE THE PERFORMANCE OF TWITTER SENTIMENT ANALYSIS

The research question we aim to address in this chapter is:

RQ3: Can incorporating social relations between users improve the performance of Twitter sentiment analysis?

We introduce our methodology of incorporating implicit social contexts of microblogging users in the sentiment analysis task. The implicit social contexts are derived by measuring the similarity among users based on three levels: a profile, a timeline, and a content level. The aim of these levels is to reveal possible relationships between any two users. Based on the proposed methodology, we create a framework, named a Social Interaction Aware-based Approach (SIAA), combining three deep learning models.

This chapter is organized as follows. Section 5.1 presents a description of our proposed methodology. In Section 5.2, we introduce the SIAA framework in detail. Section 5.3 shows the experiments and results. A description of the datasets used in this chapter is given in Section 5.3.1. Section 5.3.2 discusses how the data was pre-processed. Section 5.3.3 presents empirical evaluation results to verify the effectiveness of using a topical context to create a user embedding. In Section 5.3.4, we describe our experimental setup and analyze the results.

5.1. Methodology (Proposed Approach)

Various types of metadata are provided by social media platforms, such as Twitter. The metadata consists of useful information about users, for example, temporal, geolocation, and relational. These can be used to improve the sentiment analysis task in social media.

Our focus in this part of the research is on social relations (called social contexts) of microblogging users, and exploiting them in the sentiment analysis task. There are two kinds of social contexts: explicit and implicit [96]. The explicit social contexts are derived from direct relationships between users. An example of the explicit social contexts in Twitter is the relation between follower and followee formed by users [97]. On the contrary, the implicit social contexts can be defined as indirect relationships between users that are unobserved through direct social relationships. It refers to the relations of users who tend to write on similar topics [11]. The aim of implicit social contexts is to extract a latent connection between users who do not necessarily have a direct connection that can be leveraged to improve the accuracy of the sentiment analysis task.

Throughout the literature, there has been more attention paid to the influence of the explicit social context on sentiment analysis. In contrast, studying the usefulness of the implicit social context and ways of applying it to the sentiment analysis task in microblogging is ignored. Therefore, in this part of the research, we incorporate implicit social contexts of microblogging users in the sentiment analysis task, which enables our proposed models to effectively use the latent information extracted from the indirect relationships among users. The implicit social context features can be derived by measuring the similarity among users based on three levels: a profile, a timeline, and a content level. Each level is designed to reveal possible relationships between two given users. The following presents definitions and notations for each level.

5.1.1. Profile Level Similarity (PLS)

The aim of this level is to find a similarity between users based on their publicly available profiles information, inspired by Collaborative Filtering (CF) methods that are used by recommender systems. The purpose of recommender systems is to suggest items that users are likely to find interesting based on some historical user data [98]. With the availability of user profile information, the concept of CF can be adapted to sentiment analysis tasks. Not unlike CF, our goal in this level is to list all users whose profile details are similar to a target user. Then, the users in the created list are ranked based on a calculated score. These two steps, listing and ranking users, are performed as follows.

Let $U = \{u_1, u_2, ..., u_n\}$ be the set of users[1] , where n is the number of users, and users each have their own profile $p_{u_i} \in P = \{p_{u_1}, p_{u_2}, ..., p_{u_n}\}$. Each profile is composed of elements $e \in E = \{e_1, e_2, ..., e_m\}$, where m is the number of elements. Some examples of these elements that we take into our consideration in this level include: a number of posts a user has, the number of their friends and followers, location, geographic information, whether they keep the default profile image or change it, and a profile description. Unlike the explicit social relationships between users that are static, the implicit social relationships have dynamic characteristics. That means the explicit social relationships are going to be the same between two users regardless of the context that one person choses to follow (befriend) another person. In the implicit social relationships, the relations among users are determined by the context; therefore, it is called indirect relationships. A user, A, could be related to user B in one context and not related to B in another context. The context could be anything. It could be a topic, an item, or an event. In this research, we assume the context is a list of topics (see Section 5.3.1). Accordingly, we need a list of topics defined as $T = \{t_1, t_2, ..., t_d\}$ where d is the number of distinct topics.

[1] In this part of the research, we use the terms user and author interchangeably.

The first step is to list all users in U who had posts (tweets) regarding a specific topic in T. This topic is called a target topic. This is done by the following formula:

$$ULT\left(u_i, t_j\right) = \begin{cases} 1, & u_i \in U \ \wedge \ t_j \ \in \ T \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

The function $ULT(.)$ takes a user $u_i \in U$ and topic $t_j \in T$, to reference a list of users who wrote one or more posts using topics in T. This creates a lookup table stored in a $n \times d$ matrix A, where n represents the number of users in U and d represents the number of topics in T. We call this matrix a user-topic matrix, where $A_{i,j} = 1$ if user u_i is posted a tweet in a topic t_j, and $A_{i,j} = 0$ otherwise.

The second step is to compute the profile similarities between a target user who wrote a post in a topic and all other users listed in that topic. The similarity measure used in our work is the cosine similarity. Therefore, given a target user x, a topic t, and a list of users L in t corresponding to the matrix A, the user profile similarity denoted $UPS(x, t, L)$ is defined as a list:

$$UPS(x, t, L) = cos_t(x, y) \text{ for } y \in L \tag{5.2}$$

where $cos_t(.)$ is a cosine similarity function computed by:

$$cos_t\left(\vec{x}, \vec{y}\right) = \frac{\vec{x} \ . \ \vec{y}}{\|\vec{x}\| \ \times \ \|\vec{y}\|} = \frac{\sum_{e=1}^{m} x_e \ y_e}{\sqrt{\sum_{e=1}^{m} x_e^2} \ \sqrt{\sum_{e=1}^{m} y_e^2}} \tag{5.3}$$

where $\vec{x}$ and $(\vec{y} \in L)$ are vectors created by the profile elements, e is an element in E that forms the user profiles p_x and p_y of users x and y, $x_e \in p_x$ and $y_e \in p_y$ for $e = 1, 2, ..., m$.

The output of the computed cosine similarity step is a set of lists of the form $UPS(x, t, L)$

associated with each user x in U. As mentioned above, these lists are created based on a given topical context. Each list has a group of users along with scores, computed by Eq. (5.2), that represent their similarity with a given user and topic. Therefore, the number of lists a user could have is $|T|$.

The last step is to rank these lists of users of a target user. The best choice to accomplish this task is the K-Nearest Neighbors algorithm (K-NN). This algorithm finds the k most similar users of a given user and his/her list of users. The value of k is determined by the topic. The output of this step consists of the k nearest neighbors (users) who, based on the input values, are most similar to the given target user. Then, the sentiment ratio of each label (positive and negative) of the k neighbors is calculated. This provides us with two vectors of length k, one for each label. The average of each vector is computed and assigned to the target user. Therefore, the target user has two sentiment values.

5.1.2. Timeline Level Similarity (TLS)

This level aims to find social similarity between users based on their historical tweets (we refer to these tweets as a user Twitter timeline). In the TLS level, we utilize the target user's timeline and his/her contextual users' timelines similarity to build the implicit social relations. That is, this level is going deeper into discovering the indirect social relations between users and employ these to improve the accuracy of the sentiment analysis task. In particular, we introduce a contextual similarity to social contexts.

We replace the author profile set P of the previous section by the author timeline (historical) set H. Therefore, $h_i \in H = \{h_1, h_2, ..., h_n\}$ represents $u_i \in U$ for all tweets in his/her timeline. We use the Twitter API to obtain a Twitter timeline for each user in U. The API returns the most recent 3,200 tweets in their timelines.

Unlike PLS, TLS measures the similarity in textual settings. First and foremost, the format of the given data needs to be converted into a measurable format. One way to do this is to represent each tweet in an author timeline as vectors of features, then compare them by measuring the distance between these features. The literature offers several techniques to convert tweets into vectors [17], such as Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF) and word embeddings. Here, we decided to use the word embeddings. The reasons for this choice are that the word embeddings are useful for recognizing contextual content [99] and perform very well on sentiment analysis tasks [100]. Accordingly, we apply the pre-trained GloVe word embeddings [63]. These embeddings are obtained via training on a corpus of 2 billion tweets from Twitter [101]. The resulting vectors are then used to compute the similarities.

Given a target user x timeline, a topic t, and a list of users L in t that are derived from the matrix A along with their timelines, the author timeline similarity denoted $ATS(x,t,L)$ is defined as:

$$ATS(x,t,L) = cos_t(x,y) \text{ for } y \in L \tag{5.4}$$

where $cos_t(.)$ is a modified version of the cosine similarity function in Eq. (5.3),

$$cos_t\left(\vec{x},\vec{y}\right) = \frac{\vec{x}\ .\ \vec{y}}{\|\vec{x}\|\ \times\ \|\vec{y}\|} = \frac{\sum\limits_{j=1}^{m} x_j\ y_j}{\sqrt{\sum\limits_{j=1}^{m} x_j^2}\sqrt{\sum\limits_{j=1}^{m} y_j^2}} \tag{5.5}$$

where $\vec{x}$ and $(\vec{y} \in L)$ here are vectors created by Twitter timelines of users x and y, j refers to a feature in the created vectors, m is the number of features users x and y have in their timeline tweets, and h_x and h_y are users' x and y Twitter timelines, $x_j \in h_x$ and $y_j \in h_y$ for $j = 1, 2, ..., m$.

The output of the above equation (Eq. 5.4) is a set of lists of users who have at least one tweet in a given topic t with a target user. Each of these users has a score representing the degree of similarity between that user and the target user. The list of users is sorted in descending order. The sentiment ratio of each label (positive and negative) of the top n users on the list is calculated. This provides us with two vectors of length n, one for each label. The average of each vector is computed and assigned to the target user. Therefore, the target user has two sentiment values.

5.1.3. Content Level Similarity (CLS)

The target of this level is to find social similarity between authors based on their tweets in a given dataset. The CLS level is an alternative version of the TLS. Here, we assume that historical tweets of authors are not available for any reason.

To address the lack of historical tweets, we discover the implicit social relation between users based on their tweets that exist in the dataset only. For each user in U, a list of users that is created based on a given topical context is formed. The number of lists of each user equals $|T|$ the number of topics in T. Similar to the TLS, we apply the pre-trained GloVe word embeddings in order to represent the content as vectors. The resulting vectors are then used to compute the similarities.

Accordingly, given a target user x tweet, a topic t, and a list of users L in t that are derived from the matrix A along with their tweets in a given dataset, the author content similarity denoted $ACS(x, t, L)$ is defined as the list

$$ACS(x, t, L) = cos_t(x, y) \text{ for } y \in L \tag{5.6}$$

The $cos_t(.)$ function is calculated similarly to Eq. (5.4), except that the vectors here are created at the content level. Like the TLS, the sentiment ratio of each label (positive and negative) of the n top users in the target user list is calculated. This provides us with two vectors of length n, one for each label. The average of each vector is computed and assigned to the target user.

5.2. Proposed Models

Based on our approach explained in Section 5.1, we proposed a framework, named a Social Interaction Aware-based Approach (SIAA), combining three deep learning models, one for each level. These models are PLS-based Model (PLSM), TLS-based Model (TLSM), and CLS-based Model (CLSM).

Each model has a similar architecture with different levels of analysis. We use a deep neural network based on two architectures: Gated Recurrent Units (GRU) and Conventional Neural Networks (CNN). The GRU architectures are useful for the sentiment classification task because of their ability to overcome vanishing and exploding gradient issues [102], while the CNN architectures are able to recognize local features from textual content [103].

The inputs of the models are the target tweet, along with the tweet's author, and the contextual topic (i.e., the topic appears in the target tweet). The target tweet is fed into an embedding layer. The target tweet should be converted to a low-dimensional embedding to be acceptable as an input in this layer. Therefore, a word embedding, which is a distributional feature representation, is associated with each token (word) of the target tweet. For the purposes of this work, we utilize the publicly available GloVe embeddings [63], pre-trained on 2 billion tweets from Twitter with a dimensionality of 200. Since not all tweets have the same length (number of tokens), a padding is performed. The tweet length is normalized by padding with zeros. The padding size is determined by the maximum number of tokens a tweet in a dataset can have. To avoid overfitting, the spatial dropout layer is applied to the embedding layer. Then, the output of this layer is fed into a bi-directional GRU layer of dimension 200. Concurrently, a max-pooling layer and an average-pooling layer are applied to the output of the last step.

The target tweet's author and the topic inputs are used to extract implicit relationships. The implicit social context representation is selected based on the model target level. Also, the set of users contributing to the target topic are formed (see Section 5.1). The output of this step is then fed into a dense layer to represent implicit social context features of the target author

The above part of the model deals with a general representation (i.e., at the level of a given dataset) of a tweet. This type of representation lacks in local representation of a tweet within its topical context. Therefore, based on the assumption that tweets with the same topic should have embeddings close to each other [104], we use the (target) topic in the target tweet to create a topic-specific representation. To do this, a character-based convolutional neural network model is proposed. The reason of working at the character level instead of the word-level is that the number of tweets in a topic is less than the number of tweets in the whole dataset, which means fewer tokens (words) can be extracted at the word level for learning tweet representation compared to the character level extraction. That is, the character level representation can help the model to identify tokens (i.e., as a combination of characters) that are most related to the target topic.

Accordingly, each topic in T has its own character n-grams space constructed in the input layer. The number of consecutive characters ranges from 1 to 4 n-grams. In the embedding layer of this model, the tweet representation is created. Each created character is replaced by its TF-IDF score. Then, the representation is fed into a conventional layer to extract local features in each token window. A max-pooling layer follows, which is used to select the most important features from the feature pool.

The last outputs of each part of the framework are concatenated and fed to another dropout layer. At the final layer, the resulting output is run through a fully-connected layer where the number of neurons equals the number of labels in the task. The RMSprop optimizer [105] and a cross-entropy loss function are used in this layer. A continuous value representing the sentiment polarity of the target tweet is extracted by this final layer. The final prediction of the label of a given tweet is chosen as the (postive or negative) label with the higher probability.

5.3. Experiments and Results

5.3.1. Dataset Sources

We evaluate our models on two Twitter sentiment analysis benchmark datasets: SemEval and HCR. These two datasets are used by researchers to assess the efficiency of their proposed models in the sentiment analysis task.

Health Care Reform (HCR): This dataset was collected by [106]. It has tweets debating America's health care reform in March 2010. The HCR dataset is divided into three sets: training, development, and test dataset. Here, we deal with all three subsets as one full dataset for analysis. The dataset contains five types of labels: positive, negative, neutral, irrelevant, and unsure. In this work, our focus is on identifying the polarity of tweets. Therefore, tweets with positive and negative labels are used in our experiments and neutral tweets are filtered. The labels are assigned to nine different topics (targets): Health Care Reform, Obama, Democrats, Liberals, Conservatives, Republicans, Tea Party, Stupak, and Other.

SemEval: This dataset was created for the Twitter sentiment analysis tasks by [107] in the Semantic Evaluation of Systems challenge (SemEval). Each year, the organizers of this challenge provide a dataset for the sentiment analysis task of that year. The tweets are manually labeled by five Amazon Mechanical Turk (AMT) workers. The provided dataset can be created specifically for that year or from previous years. In this part of the research, we conduct our experiments using a dataset collected from the 2015 and 2016 SemEval Twitter challenge datasets. Each tweet in the dataset comprises the tweet ID, the author ID, the topic of the tweet, and the sentiment label (positive or negative). The number of topics in the SemEval dataset is 274. The Twitter API is used to download all tweets. Detailed statistics of these datasets are provided in Table 5.1

Table 5.1: Statistics of datasets

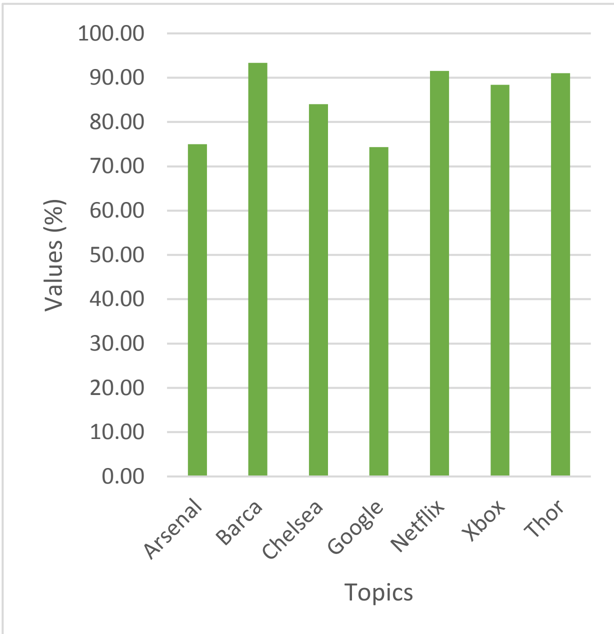
	SemEval	HCR
No. of Positive Tweets	10,993	448
No. of Negative Tweets	4,568	1,125
Total	15,561	1,573
No. of Topics	274	9
No. of Users	14,814	920

5.3.2. Preprocessing

The preprocessing steps mentioned in Section 4.4.2 are used for this research question. Topics that have less than 30 tweets are removed along with their tweets. Any duplicated tweets are eliminated.

5.3.3. Usefulness of Topic Context

Here, we present empirical evaluation results to verify the effectiveness of our method of using topical context to create a user list that is used later on as user representation (i.e., user embedding) for inferring implicit social relationships of that user. We apply the proposed method on the two datasets introduced in Section 5.3.1.



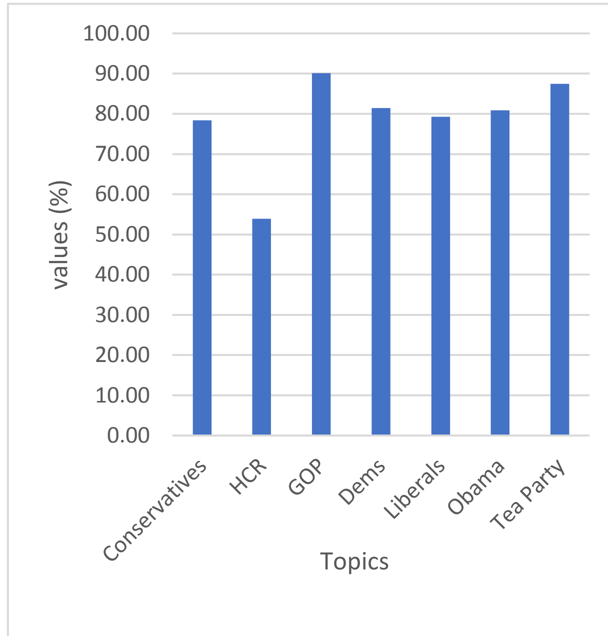


Figure 5.1: Percentage of agreement in opinions among authors, based on a given topic in the datasets HCR (on the right) and SemEval (on the left).

Figure 5.1 depicts the percentage of agreement in opinions among authors, based on a given topic in the datasets HCR (on the right) and SemEval (on the left). It shows the seven most common targets (topics) in both datasets. The HCR dataset topics are Health Care Reform (HCR), Conservatives, Obama, Democrats (Dems), Tea Party, Liberals, and Republican (GOP), while the SemEval dataset topics are Arsenal, Barca, Chelsea, Google, Netflix, Xbox, and Thor.

The values in Figure 5.1 indicate that there is a relatively strong correspondence between user opinions and topics on which the users like to share their thoughts. For example, the Thor and Barca topics obtain about 90% and 93%, respectively, of agreement among the users who wrote about these topics. It is worth noticing that the opinion agreements on non-general topics such as Thor and GOP are higher than general topics such as Google and HCR. Overall, using the topical context to create user embeddings yields a sufficiently accurate representation.

5.3.4. Results

We trained and validated our models on the training and development datasets and tested their performance on the test dataset (see Section 5.3.1). In order to illustrate the performance of our approach, we compare the results with several baseline models as described below:

Naive Bayes (NB): Naive Bayes works well on text categorization [85]. This is a probabilistic algorithm that uses Bayes' rule. Eq. (5.7) represents Bayes' rule,

$$P(cd) = \frac{P(c)P(d|c)}{P(d)} \tag{5.7}$$

where c is the class and d is the document (tweet) under consideration.

Support Vector Machine (SVM): Another popular classification technique relies on the Support Vector Machine. According to [87], SVM has been shown to be highly effective at text categorization.

Convolutional Neural Network (CNN): This is a typical neural network architecture for sentiment analysis tasks [48], which leads to very competitive results on sentiment classification [108].

Multilayer Perceptron (MLP): MLP is a type of feedforward Artificial Neural Network (ANN) [109]. Units are structured in layers consisting of an input layer, one or more hidden layers, and an output layer. Layers are arranged consecutively. Data passes through layers starting from the input layer and ending at the output layer. MLP has been used successfully in many fields [40].

Long Short-Term Memory (LSTM): The LSTM models have achieved impressive performance on the sentiment classification task [91]. They have the ability to handle long-range dependencies [68].

Tables 5.2 and 5.3 show the results obtained from the baseline models and our models in accuracy, precision, recall, and F1 measures, along with the averages of precision, recall, and F1 across all datasets. Among the baseline models, the highest accuracy, 88.69%, is achieved on the SemEval dataset using LSTM, and 77.78% on the HCR dataset using SVM. On the other hand, MLP provides the lowest performance in accuracy on both datasets (SemEval: 73.04% and HCR: 72.84%).

These results show that our proposed models: PLSM, TLSM, and CLSM, outperform all models by a substantial margin. It is also worth noting that our framework provides a stable performance across the classes.

As for per-class sentiment classification, we observe that all baseline models perform relatively poorly on recalling negative tweets on the SemEval dataset, and positive tweets on the HCR dataset. For example, the recall of the baseline models for negative tweets ranges from 12.83% to 76.57% on the SemEval dataset. Similarly, the recall of the baseline models for positive tweets ranges from 27.03% to 61.90% on the HCR dataset. This may be due to the imbalanced sentiment class distribution in the datasets. This is a common issue with most social media datasets, which has a huge negative effect on classifiers created for sentiment analysis tasks.

However, our evaluation shows that our models based on the proposed approaches are robust with respect to the imbalanced dataset issue. They produce a consistent and superior performance compared to the baseline models on both datasets. The accuracy of our proposed models ranges from 86.80% to 90.10% on the SemEval dataset and from 83.75% to 90.96% on the HCR dataset. Furthermore, it can be clearly seen in Figure 5.2 that our

Table 5.2: Experimental results of our models, compared with baselines on the SemEval dataset

			Positive			Negative			Average		
	Model	Accuracy	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Baseline Models	CNN	87.35%	89.70%	93.04%	91.34%	80.56%	76.57%	76.57%	87.11%	87.36%	87.16%
	LSTM	88.69%	90.68%	93.87%	92.25%	82.97%	75.59%	79.11%	88.50%	88.69%	88.53%
	NB	86.19%	87.19%	94.38%	90.64%	82.92%	66.30%	73.68%	85.94%	86.19%	85.70%
	SVM	84.93%	89.40%	89.36%	89.38%	74.03%	74.12%	74.07%	84.94%	84.93%	84.94%
	MLP	73.04%	73.24%	97.70%	83.72%	69.57%	12.83%	21.66%	72.17%	73.04%	65.69%
Our Approach (SIAA)	PLSM	87.50%	92.88%	92.70%	92.78%	82.02%	82.40%	82.21%	87.45%	87.55%	87.50%
	TLSM	86.80%	91.73%	93.34%	92.53%	82.94%	79.20%	81.03%	87.34%	86.27%	86.78%
	CLSM	90.10%	93.24%	92.81%	93.02%	82.48%	83.41%	82.94%	90.13%	90.10%	90.11%

Table 5.3: Experimental results of our models, compared with baselines on the HCR dataset

			Positive			Negative			Average		
	Model	Accuracy	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Baseline Models	CNN	76.14%	60.76%	43.24%	50.53%	80.00%	89.05%	84.28%	74.58%	76.14%	74.77%
	LSTM	76.14%	62.69%	37.84%	47.19%	78.90%	91.17%	84.59%	74.33%	76.14%	74.05%
	NB	77.21%	76.19%	34.04%	47.06%	77.37%	95.50%	85.48%	77.02%	77.22%	74.05%
	SVM	77.78%	57.78%	61.90%	59.77%	85.78%	83.55%	84.65%	78.31%	77.78%	78.01%
	MLP	72.84%	53.57%	27.03%	35.93%	76.04%	90.81%	82.77%	69.71%	72.84%	69.57%
Our Approach (SIAA)	PLSM	83.87%	91.35%	61.46%	73.48%	87.07%	97.76%	92.11%	89.21%	79.61%	82.79%
	TLSM	90.96%	97.87%	76.03%	85.58%	91.52%	99.43%	95.31%	94.70%	87.73%	90.45%
	CLSM	83.75%	84.19%	52.85%	64.94%	83.65%	96.05%	89.42%	83.80%	83.75%	82.45%

proposed models produce better values in positive recall and precision than the baseline models. This shows the effectiveness of our approach compared to the other models.

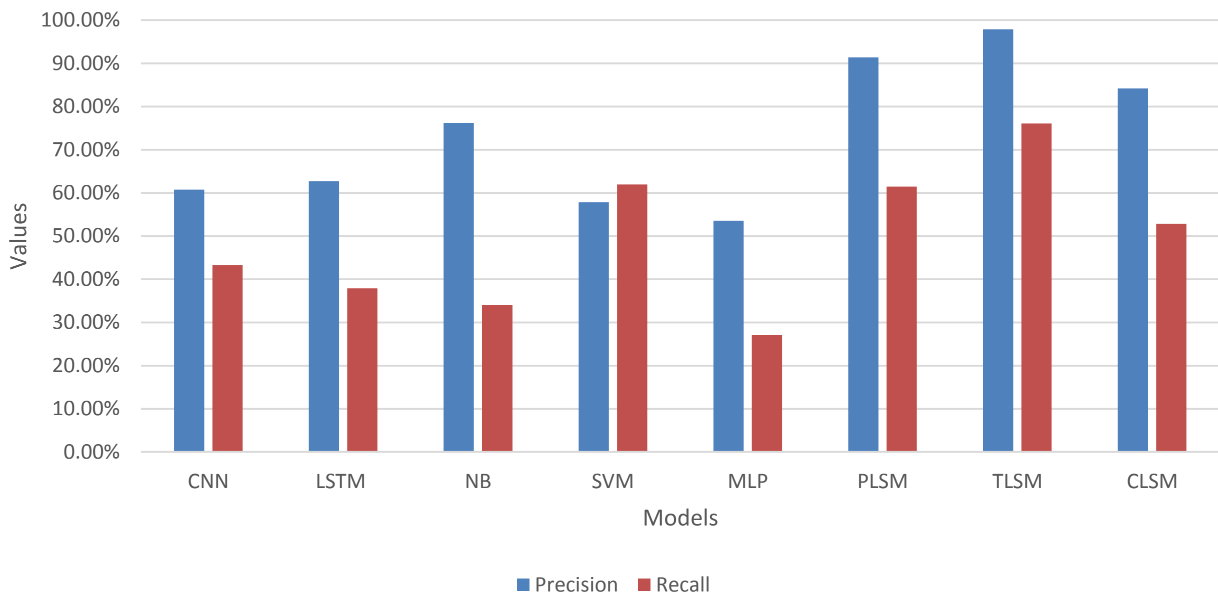


Figure 5.2: Recall and precision values comparison of the positive label on the HCR dataset.

One difficulty in the sentiment analysis task is the lack of labeled training data [98]. It can be clearly observed from the results that this issue has a huge negative impact on the performance of the baseline models. This includes traditional machine learning (SVM and NB) and deep learning-based models (LSTM, CNN, and MLP). The performance of these models on the HCR dataset is lower compared to their performance on the SemEval dataset. For instance, the accuracy of SVM and NB on the SemEval dataset is 84.93% and 86.19%, respectively, while their accuracy on the HCR dataset is 77.78% and 77.21%, respectively. On the SemEval dataset, LSTM and CNN attain 88.69% and 87.35% in accuracy, respectively. Conversely, they produce a low accuracy score, 76.14%, on the HCR dataset.

Our proposed approaches and the three designed models overcome the issue of the need for a large amount of labeled data in order to have a better performance. The results show that our models outperform baseline models for identifying both negative and positive sentiments on both datasets regardless of data size. The accuracy of PLSM, TLSM, and CLSM on the SemEval dataset, which has a large number of training data, is 87.50%, 86.80%, and 90.10% respectively. These models provide consistent performance on the HCR dataset, which has a lack of labeled training data. The accuracy and average precision are 83.87% and 89.21% for PLSM, 90.96% and 94.70% for TLSM, and 83.75% and 83.80% for CLSM.

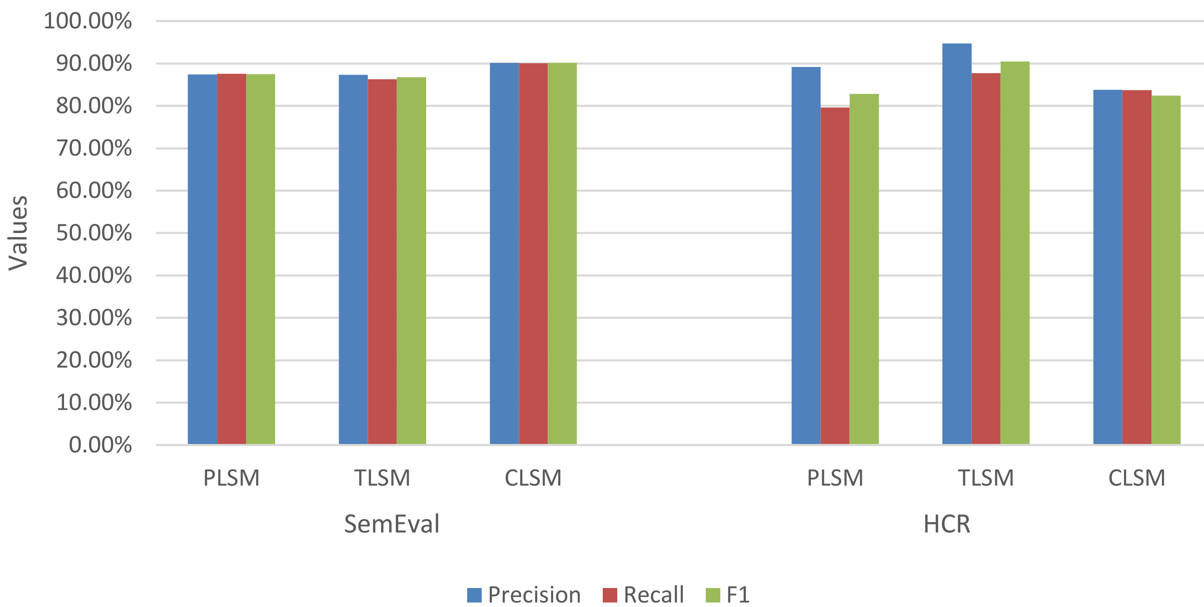


Figure 5.3: Comparing average precision, recall, and F1 values of our models for the SemEval and HCR datasets.

Figure 5.3 shows the average precision, recall, and F1 comparison of our proposed models (PLSM, TLSM, and CLSM) on the SemEval and HCR datasets. As mentioned in Section 5.2, each model deals with its designated level to extract the implicit social context among users. PLSM works at the profile level, TLSM at the timeline level, and CLSM at content level.

Figure 5.3 demonstrates that, by exploiting social relations, our proposed models outperform the text-based methods significantly.

Even though the performance of the proposed models is similar across the datasets, some observations are worth noticing. The TLSM model has higher average precision and F1 on the HCR dataset (94.70%, 90.45%, respectively) compared to the SemEval dataset (87.34%, 86.78%, respectively). This shows that extracting social contexts at the timeline level can leverage the sentiment analysis task, especially in case of a dataset, such as HCR, with an insufficient amount of data. As another observation, comparing the performance of PLSM on the SemEval dataset with the HCR dataset, we can see that PLSM attains a relatively high score in precision (89.21%) on the HCR dataset. For CLSM, the obtained results on the HCR dataset are slightly worse compared to the SemEval dataset. This could be explained by the fact that the method extracts the social contexts at this level. At the content level, the social context among users is created based on their tweets in a given dataset (see Section 5.1). Accordingly, the size of the given dataset can affect the performance of that model.

In general, the models based on our proposed approaches produce relatively balanced results on both datasets. The results of our proposed models demonstrate the effectiveness of enriching models with the additional sources of information, by going beyond the textual content of a given document (tweet). Here, the sources of information are obtained from social contexts. Furthermore, the results show that our methods can be used to overcome the issue resulting from a lack of training data.

CHAPTER 6

THE EFFECT OF DETECTING POLARITY AT THE TOPIC LEVEL ON SENTIMENT CLASSIFICATION PERFORMANCE

In this chapter, we address the fourth research question of this thesis. Our goal is to explore the effect of detecting polarity at the topic level on sentiment classification performance. Discovering the polarity at this level is more appealing in some cases, for example, a given tweet without enough words to infer its polarity. To achieve this goal, we introduce the power of utilizing the information at the topic level into the sentiment analysis task. Therefore, we propose a framework for Twitter sentiment analysis based on a deep learning approach utilizing awareness of topic-level information.

This chapter is organized as follows. Section 6.1 presents a description of our proposed methodology. Section 6.2 shows the experiments and results. The description of the datasets used in this chapter is shown in Section 6.2.1. Section 6.2.2 discusses how the data was pre-processed. In Section 6.2.3, we describe our experimental setup and analyze the results.

6.1. Methodology (Proposed Approach)

In this part of the research, we have a corpus C in which $c_i \in C$ is composed of a triplet (d_i, t_i, l_i), where $d \in D = \{d_1, d_2, ..., d_n\}$ is a document of n tweets, $t \in T = \{t_1, t_2, ..., t_m\}$ is one of m distinct topics, and $l \in L$ is a sentiment label. Each topic in T is associated with a set of tweets $t_i = \{t_{i1}, t_{i2}, ..., t_{ig}\}$, where g is the number of tweets in the topic. Therefore, given a corpus $\acute{C}$ of documents with their associated topics T and corresponding sentiment labels L, the task is to predict a sentiment label for a given tweet based on a topic where the tweet has appeared.

Our key idea is to introduce the power of utilizing the information at the topic level into the sentiment analysis task. Users post tweets on different topics on Twitter to express their opinions and thoughts. While sentiment analysis at the tweet level provides very useful information, addressing a sentiment tendency towards topics is more appealing in some cases. For example, for a given tweet without enough words to allow a classifier to infer the tweet polarity, incorporating the topic-level polarity of the tweet can provide the classifier with useful indicators, which in turn can enhance the Twitter sentiment analysis performance.

Accordingly, we propose a framework for Twitter sentiment analysis based on a deep learning approach utilizing awareness of topic-level information. The framework consists of four main components and each one of these deals with the task at a different level. Figure 6.1 provides a high-level overview of our framework.

The first component of our framework targets the sentiment analysis task at the tweet-level. Basically, we built and trained a tweet-level sentiment classification model based on a deep learning approach. The model consists of a Bidirectional RNN network, based on a GRU (Gated Recurrent Unit). We will refer to this model as a Bidirectional GRU-Tweet-Level Model (BiGRU-TLM).

The BiGRU-TLM architecture includes an input layer, an embedding layer, a spatial dropout, a Bi-GRU layer, and a max and an average pooling layer. At the input layer, the pre-processed tweet d_i is treated as a sequence of words, $W_{d \in D} = (w_1, w_2, ..., w_n)$. These are given by a one-hot vector, which has the length of the size of the vocabulary. Because of the inconsistency in the length of tweets, the shorter tweets need to be padded. We normalize the tweet length by padding with zeros.

At the embedding layer, we apply a pre-trained GloVe word embedding on each word in the vocabulary list. 200-dimensional GloVe vectors are adapted with 27 billion tokens (words), which are trained on 2 billion tweets from Twitter. To reduce overfitting, the

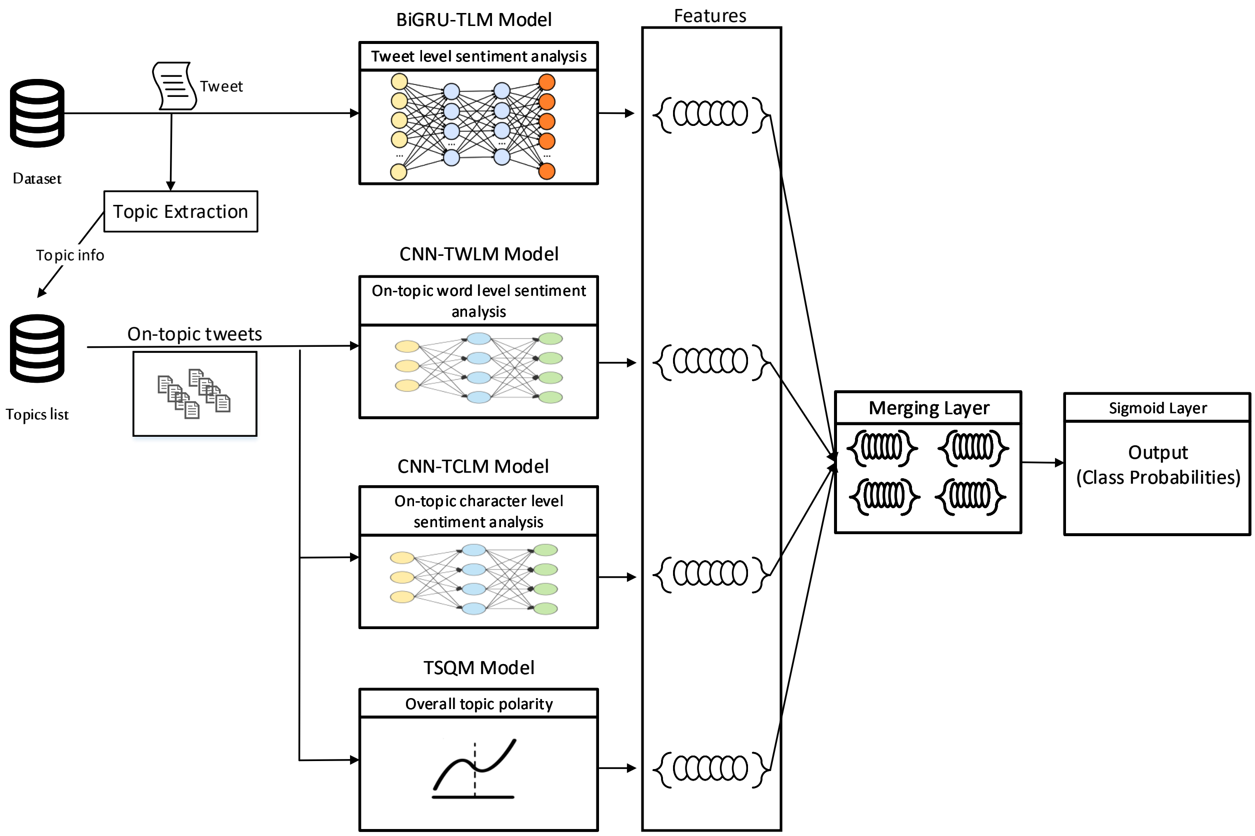


Figure 6.1: High-level overview of our framework.

spatial dropout layer is applied to the embedding layer. The Bi-GRU layer contains 100 neurons. The temporal information of the tweet sequence is captured in this layer in both directions, forward and in reverse. Concurrently, the output of this layer is fed to two pooling layers, max and average. The outputs of both layers are concatenated and then appended to other outputs in a merging layer. The aim of the merging layer in the proposed framework is to combine the outputs of all components in the framework into one final vector.

The second component of our proposed framework deals with the sentiment analysis task at the word-level within the topical context. Not all words (tokens) have the same level of impact on a topic where they appear. Therefore, it is reasonable not to treat all words at the same level of importance at the topic perspective. Based on this assumption, we seek to infer word importance for a given topic and use this as an indicator of the topic polarity.

To do this, we propose a deep learning model using Convolutional Neural Networks (CNN), that works directly on tweets of a specific topic to improve the sentiment classification. We will refer to this model as a CNN-Topic and Word Level Model (CNN-TWLM).

The CNN-TWLM architecture includes an input layer, an embedding layer, a one-dimensional convolutional neural network layer, and an average pooling layer. The input layer has two tasks: preparing the required data for each topic in T and creating a representation of each word. Since the main corpus C has all tweets along with topics stored in one main set, subsets of C are created. This could be defined as follow:

$$S_{t_i} = \{(d, t_i, l) \subset \acute{C} | \text{ d appears only in } t_i \in \text{T}\}.$$

Each S_{t_i} is a subset of C that contains only tweets from D occurring in one specific topic t_i. This is the first task of the input layer. The second task of the input layer consists of taking the S_{t_i} and forming a vector representation of words associated only with the topic t_i. Each topic in T has its own vector representation of its own words. Therefore, the number of vector representations constructed is $|T|$.

The vector representation of each word in a given topic in the CNN-TWLM input layer is built using a Bag-of-N-Grams (BoNG) approach. The n-gram in BoNG is set up for two gram types: unigrams and bigrams. In the BoNG approach, each word is represented as a one-hot vector. The length of each vector is equal to the size of the vocabulary of a given topic. That is, only one dimension is 1, with all others being 0. However, the one-hot vector alone is not sufficient to determine the importance of words toward a topic. We need a weighting scheme that allows to associate each word with a score reflecting its importance toward a topic where it occurred. To achieve that, we employ a Term Frequency-Inverse Document

Frequency (TF-IDF) scheme. It is a weighting scheme that represents the importance of a word w to a document d [110]. it is calculated by

$$\mathrm{W}_{i,j} = \mathrm{TF}_{i,j} \times \log \frac{N}{\mathrm{DF}_i} \tag{6.1}$$

where $\mathrm{TF}_{i,j}$ is the term frequency of i in j, DF_i (document frequency) is the number of documents contains i, and N is the total number of documents. Therefore, the TF-IDF score, calculated as

$$\mathrm{WS}(S_{t_i} \subset \acute{C}, w) = \mathrm{TF}(S_{t_i}, w) \times \log \frac{|S_{t_i}|}{|S_{t_i}, w|} \tag{6.2}$$

is a weighting scheme defined for a subset S_{t_i} of $\acute{C}$ containing only tweets that occurs in a topic t_i, and a word w. Here $TF(S_{t_i}, w)$ is term frequency, i.e., the number of occurrences of the word w in the tweets of S_{t_i}, $|S_{t_i}|$ is the number of tweets associated with the topic t_i, and $|S_{t_i}, w| = |\{d \in D \mid (d, t_i, l) \in S_{t_i}$ and d contains the word w$\}|$ is the number of tweets in S_{t_i} that contain the word w.

In the embedding layer of CNN-TWLM, the tweet representation is created. Each word of tweets in S_{t_i} is replaced by its TF-IDF score calculated by Eq. (6.2). The size of the embedding layer is set to 61, which is the maximum numbers of words (terms) in the tweets across all topics in T for the considered dataset. The resulting representation is fed to the convolutional neural network layer where the local features are extracted. Then, a pooling function in the average pool layer is applied to the output of the convolutional neural network layer. Following this, the output of CNN-TWLM is concatenated with the outputs of other models in the proposed framework.

The third component of our proposed framework operates on the character level within the topical context. Most linguistic issues exist in social media settings (in particular, Twitter), for example, spelling mistakes, the extensive use of slangs and abbreviations, and presence of content with mixed languages [58]. Dealing with the sentiment analysis task at the character level reduces the effect of these issues. Therefore, we propose CNN-TCLM (CNN-Topic and Character Level Model), a character-based convolutional neural network model. As main advantage of this model, it provides our proposed framework with the ability to learn in a language-independent manner.

The CNN-TCLM model architecture is similar to CNN-TWLM, except that it works on character level instead of word level. Each topic in T has its own character n-grams space constructed in the input layer. The number of consecutive characters ranges from 1 to 4 n-grams. In the embedding layer of this model, the tweet representation is created. Each created character of the tweets in S_{t_i} is replaced by its TF-IDF score (which is a modified version of Eq (6.2)), as follows:

$$\mathrm{WS}(S_{t_i} \subset \acute{C}, ch) = \mathrm{TF}(S_{t_i}, ch) \times \log \frac{|S_{t_i}|}{|S_{t_i}, ch|} \tag{6.3}$$

is a weighting scheme defined for a subset S_{t_i} of $\acute{C}$ containing only tweets that occurs in a topic t_i, and a character ch. Here $TF(S_{t_i}, ch)$ is term frequency, i.e., the number occurrences of the character ch in the tweets of S_{t_i}, $|S_{t_i}|$ is the number of tweets associated with the topic t_i, and $|S_{t_i}, ch| = |\{d \in D \mid (d, t_i, l) \in S_{t_i}$ and d contains the character ch $\}|$ is the number of tweets in S_{t_i} that contain the character ch. The output of this model is concatenated in a merging layer with the outputs of other models in the proposed framework.

The fourth component of our proposed framework is designed to provide the framework with the overall sentiment polarity of all topics in T across the different classes. This can be done by creating a model based on a quantification approach. We will refer to this model as a Topic Sentiment Quantification Model (TSQM). In this model, an estimation of a tweet distribution on a given topic is provided for each class.

Each topic in T has its own TSQM model, built and trained on its own dataset, $S_{t_i} \subset C$. The number of created models is $|T|$. A logistic regression classifier is used to create TSQM. It is trained with hybrid features: word n-gram (1-2) and character n-gram (1-4). The estimation given by

$$\mathrm{TP}(l, t_i) = \frac{|S_{t_i}, l|}{|S_{t_i}|} \tag{6.4}$$

where $\mathrm{TP}(l, t_i)$ is a function that computes the topic probability of t_i in T being labeled l, $|S_{t_i}, l|$ is the number of tweets in S_{t_i} labeled l, and $|S_{t_i}|$ is the number of all tweets in the topic t_i. The resulting values are appended to other outputs in the merging layer of the framework.

Concurrently, the outputs derived from all the above components are concatenated to form a single feature vector, and the dropout layer is applied to the feature vector. This vector is then fed to a fully connected sigmoid layer. In this layer, a continuous value representing the sentiment polarity (as a probability) of a given tweet occurring in a specific topic is calculated. The highest value is chosen for the final prediction of the sentiment of the tweet.

6.2. Experiments and Results

6.2.1. Dataset Sources

We evaluate our models on the benchmark datasets provided by the SemEval challenge runs in 2015 for subtasks B and D, and 2016 for subtasks B, D, C, and E [111], [112]. Each dataset is divided into training, development and testing set. The datasets are labeled in a two-point scale (positive and negative classes), and a five-point scale (-2, -1, 0, 1, and 2); corresponding to strongly negative, negative, neutral, positive, and strongly positive).

In all the datasets, Twitter status IDs and topics which appeared in them are given along with annotations. The Twitter API is used to download all tweets. An overview of the statistics of the used datasets is provided in Table 6.1 and 6.2.

Table 6.1: SemEval-(2015-2016)-BD datasets statistics

Dataset		No. of Topics	Positive	Negative	Total
SemEval-2015-BD	Train	137	870	1,516	2,386
	Test	44	142	344	486
SemEval-2016-BD	Train	60	3,591	755	4,346
	Dev	40	2,139	603	2,742
	Test	100	8,212	2,339	10,551

Table 6.2: SemEval-2016-CE dataset statistics

	SemEval-2016-CE		
	Train	Dev	Test
No. of Topic	60	40	100
Strongly positive	437	201	382
Positive	3,154	1,938	7,830
Neutral	1,654	1,258	10,081
Negative	668	529	2,201
Strongly negative	87	74	138
Total	6,000	4,000	20,632

The datasets used in this part of the research are different in scales. To get the most benefit of the datasets, they need to be unified. Since we deal with the problem as a two-point scale sentiment classification task, we convert the five-point scale dataset into a two-point scale based on the following equation:

$$\mathrm{LC}(t_i, l_i) = \begin{cases} negative & \text{if } \; l_i = \{-2, -1, 0\} \\ positive & \text{if } \; l_i = \{1, 2\} \end{cases} \tag{6.5}$$

where $\mathrm{LC}(t_i, l_i)$ is a Label Converter function that receives a tweet (t_i) and its original label (l_i) and does the proper label conversion. If l_i equals either -2, -1, or 0, the label of the t_i becomes negative. Similarly, if l_i equals either 1 or 2, the label of the t_i becomes positive. According to the dataset provider, SemEval, tweets labeled neutral (i.e., 0) can be treated as negative ones. Therefore, the label 0 is associated with a negative label in Eq. (6.5). After applying the LC, the total number of tweets used in this part of the research is 49,596: 28,893 positive tweets and 20,703 negative tweets.

6.2.2. Preprocessing

The syntax of tweets is commonly not well-structured. Thus, we applied the following preprocessing steps to the raw data. Each tweet is converted to lowercase. All Twitter handles (URLs, mentions '@', and hashtags '#') are replaced by placeholders <url>, <user>, and <hashtags>, respectively. Since numbers, punctuation, and special characters do not provide useful sentiment information, they are eliminated from the data. All words of length less than three characters are also removed. Topics that have less than 30 tweets are removed along with their tweets. Any duplicated tweets are eliminated.

Elongated words that have one letter repeated more than twice are tagged and substituted with the same words but where at most two consecutive occurrences are kept. For example, "goooooood" becomes "good". Emoticons are replaced with tags reflecting their meaning. For example, the ':)' emoticon is replaced by a <smile> tag.

Our proposed models have pre-trained embeddings with the GloVe algorithm (see Section 4.3). Therefore, for optimal benefit, we also apply the same preprocessing steps mentioned in the GloVe website, which are used to create GloVe word embeddings on Twitter data. Finally, for tokenizing, we utilize the tokenizer provided by the Keras[1] library.

6.2.3. Results

We trained and validated our models on the training and development datasets, and tested their performance on the test dataset (see Section 6.2.1). In order to illustrate the performance of our approach, we compare the results with several baseline models as described below.

Model 1 (M1): This model is proposed by [113]. It is based on a majority vote scheme. It combines several supervised machine learning algorithms: Ridge, Logistic Regression, Stochastic Gradient Descent, Nearest Centroid, Bernoulli Naïve Bayes, Linear SVC, and Passive-Aggressive.

Model 2 (M2): The authors of [114] build a model for a topic-based sentiment classification using a simple convolutional neural network. It consists of six layers: an input layer, a convolutional layer, a max-pooling layer, a topic embedding layer, a concatenate layer, and an output layer.

Model 3 (M3): This model applies a quantification approach with a deep learning based model for sentiment analysis tasks on Twitter. A Long Short-Term Memory (LSTM)

[1]https://keras.io/

is used as a model structure, which is an extension of a Recurrent Neural Network (RNN) model. This model is presented in [115].

Model 4 (M4): In [116], a deep multi-layer convolutional neural network is proposed. This model operates at the character level. It takes character embeddings as an input. Then, a series of three convolutional filters is applied to the embeddings. In this model, three non-linear layers are used with a variety of activation functions: linear rectification, hyperbolic tangent, and sigmoid.

Model 5 (M5): This model is presented in [117]. The authors combine a Support Vector Machine (SVM) model with Word2Vec vectors generated by training tweets.

Model 6 (M6): [118] applies a Multi-Kernel Gaussian Process model for sentiment analysis on Twitter. It is based on non-parametric Bayesian modeling. The model uses a fixed rule multi-kernel learning method and vectors constructed by a Bag-of-Words (BOW) approach.

Model 7 (M7): This model presented in [119] is built by combining a deep learning model (Gated Recurrent Units) and SVM. The model first extracts features using GRU trained on pre-trained word embeddings (obtained by GloVe). Then, the resulting word embeddings feed into a SVM classifier to achieve the sentiment analysis task.

Table 6.3: Experimental results of our model, compared with baselines

		Positive			Negative			Average		
Model	Accuracy	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
M1	60.70%	50.40%	93.30%	65.40%	89.80%	39.20%	54.60%	70.10%	66.30%	60.00%
M2	77.70%	67.30%	85.50%	75.30%	88.30%	72.50%	79.60%	77.80%	79.00%	77.50%
M3	79.00%	69.70%	83.80%	76.10%	87.60%	75.90%	81.30%	78.65%	79.80%	78.70%
M4	71.20%	83.40%	78.70%	81.00%	37.60%	45.00%	40.90%	60.50%	61.80%	61.00%
M5	80.90%	71.80%	85.90%	78.20%	89.30%	77.60%	83.00%	80.55%	81.80%	80.60%
M6	51.80%	44.90%	92.20%	60.40%	82.90%	25.00%	38.50%	63.90%	58.60%	49.40%
M7	49.90%	41.10%	59.60%	48.70%	61.90%	43.50%	51.10%	51.50%	51.60%	49.90%
TACM	89.20%	88.00%	84.60%	86.27%	90.70%	93.40%	92.03%	89.35%	89.00%	89.15%

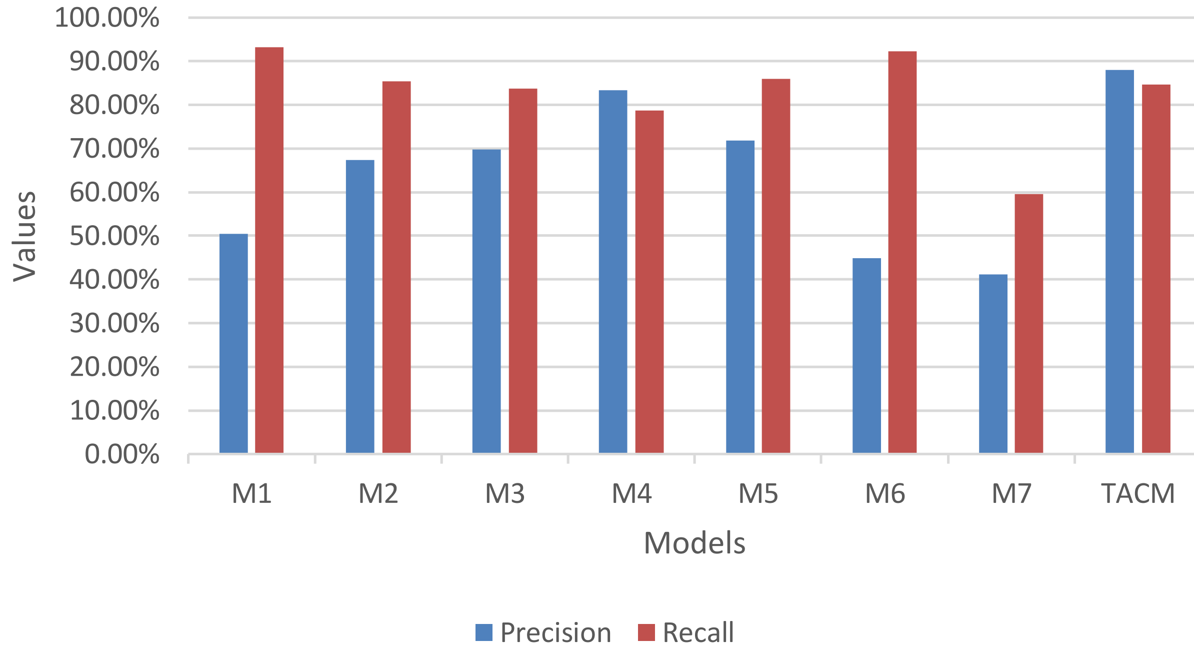


Figure 6.2: Recall and precision values comparison of the positive label.

The results in accuracy, precision, recall, and F1 metrics along with the averages of precision, recall, and F1 across all classes are reported and obtained for all models. These metrics values are presented in Table 6.3.

The experimental results demonstrate that our proposed framework, TACM, outperforms other baseline models and achieves a significant result with 89.20% in accuracy. Similarly, when we look at the average precision, recall, and F1, our proposed framework produces the highest performance among all models, with an average F1 of 89.15%, precision of 89.35%, and recall of 89.00%. These results show that TACM outperforms the other models by a substantial margin. It is also worth noting that our framework provides a stable performance across the (positive and negative) classes.

We see a low performance of models that are only using texts (tweets) compared to

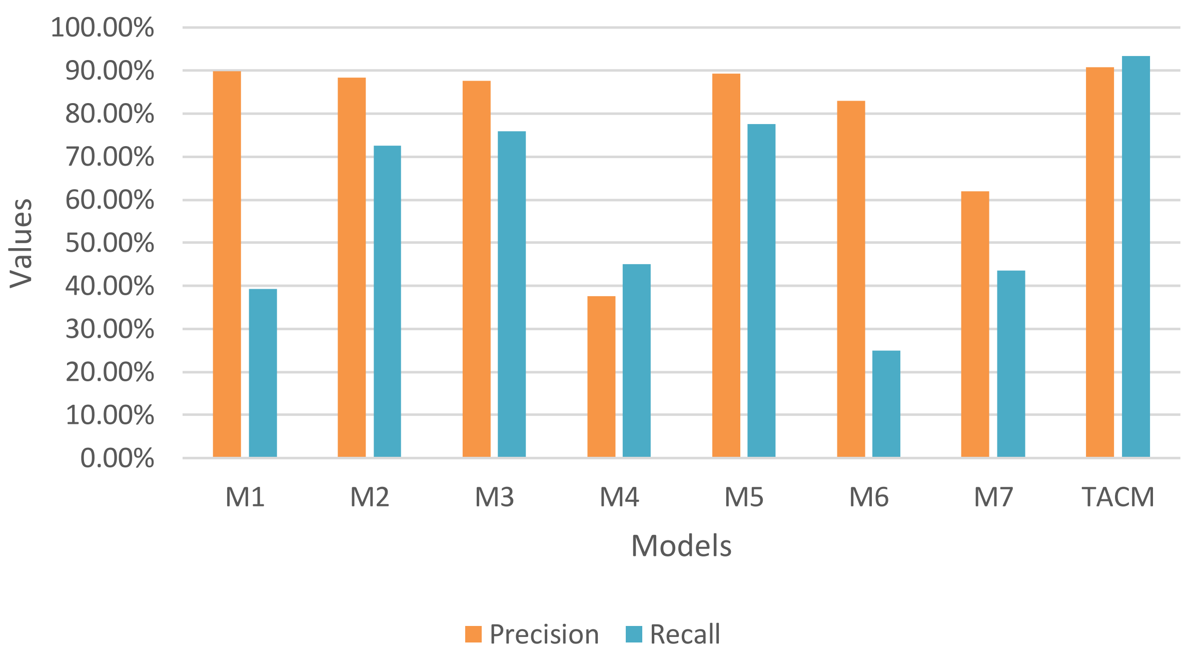


Figure 6.3: Recall and precision values comparison of the negative label.

methods using topical context. Models M6 and M7 attain 51.80% and 49.90% in accuracy, respectively, whereas M2 attains 77.70% in accuracy. These results show that incorporating topical context information in a Twitter sentiment classifier may indeed lead to an improvement.

Figure 6.2 depicts the positive class precision and recall of the models, while Figure 6.3 shows the precision and recall in the negative class of all models. While the accuracy gives an overall evaluation of the model performance, the precision and recall values are equally important, since they reveal much more information about the classification property. For example, as for per-class sentiment classification, the recall values of the positive class produced by some models (M1 and M6; 93.30% and 92.20%, respectively) are better than our framework, which achieves 84.60%. However, at 88% our proposed framework outperforms the M1 and M6 models by 37.60 - 43.10% in the precision values of the same class (see Table

6.3 and Figure 6.2). As for the negative class, most baseline models show extremely low recall values compared to their precision values. This is shown in Figure 6.3. Such varying performance might be due to the imbalanced distribution in the number of positive and negative instances (see Section 6.2.1).

From the above, we notice that the imbalanced dataset issue has a significant impact on the baseline models. However, looking at the recall and precision values across the classes, our proposed framework provide us with a good and stable performance. This means that our proposed framework overcomes this issue. It also demonstrates the effectiveness of enriching the model with additional sources of information, going beyond the textual content of a given document (tweet). Here, the additional information is obtained from the topical context.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

The main objective of this thesis was to improve the performance of the sentiment analysis task in microblogs. Twitter was used as an example case study of microblogging platforms. We investigated the role of applying fine-grained methods to the Twitter sentiment analysis task, which overcomes some difficulties faced by traditional machine learning based methods. Unlike standard texts where many words help gather enough statistics, tweets in Twitter consist only of a few characters. Moreover, tweets are more likely to have abbreviations or acronyms that appear infrequently in conventional documents. Therefore, applying traditional methods to such settings will not provide us with acceptable performance. Here, we looked at the problem, not only from a linguistic perspective, but also from several perspectives that went beyond the content of a document (tweet). To this end, we examined the following four research questions:

- **RQ1:** Is there a relationship between the user's behavior and his/her posts? And if such a relationship exists, can it be used to enhance sentiment analysis performance?
- **RQ2:** What emotion (mood) did the author express prior to the tweet that is to be classified? Can this information enhance the model performance?
- **RQ3:** Can incorporating social relations between users improve the performance of Twitter sentiment analysis?
- **RQ4:** What is the effect of detecting polarity at the topic level on sentiment classification performance?

To address the above research questions, we proposed multiple approaches and frameworks based on deep neural networks.

For the first research question, we presented a sentiment analysis model developed by combining a list of features. We proposed the architecture of a Convolutional Neural Network (CNN) that takes into account not only the text (user tweets) but also user behavior. Our evaluation demonstrated the efficiency of the model in a social media setting.

Our model outperformed the baseline methods in accuracy, recall, precision, and F1. In addition, the proposed model was affected less by unbalanced dataset issues. Moreover, the approach overcame the issue of needing a large dataset to train deep learning models such as CNN and LSTM. We introduced this method in Chapter 3.

To answer the second research question, we applied our approach of going beyond the textual content of a tweet, by taking into account not only the text but also the emotional state of the user who wrote that tweet. We proposed the Emotional Awareness based Classification Model (EACM), using a bidirectional RNN network structure based on a gated recurrent unit. We built and trained five deep neural network models, collectively named BiGRU-ESM (Bidirectional GRU-Emotional State Model), one model for each of the emotion categories of anger, disgust, joy, optimism, and sadness. The task of these five models was to provide the main model (EACM) with the emotional state of the users (writers) as extracted from their tweet history.

The experimental results demonstrated the effectiveness of the proposed approach in comparison with other baseline models that utilize the textual content only. Specifically, the results showed a considerable improvement in performance and accuracy of the sentiment classification tasks with the new approach. This method was discussed in Chapter 4.

In Chapter 5, we introduced our proposed approach to address the third research question. We explained our notion of incorporating implicit social contexts of microblogging users in the sentiment analysis task. The implicit social contexts were derived by measuring the similarity among users based on three levels: a profile, a timeline, and a content level. The aim of these levels is to reveal possible relationships between users.

Based on this approach, we proposed a framework named the Social Interaction Aware-based Approach (SIAA) combining three deep learning models. The models are the PLS-based model (PLSM), TLS-based model (TLSM), and CLS-based model (CLSM). Each model targets a specific level. Gated Recurrent Units (GRU) and Conventional Neural Networks (CNN) were used to build these models.

Overall, our findings provide further support for our hypothesis that incorporating the social relations of microblogging users helps strengthen the learning of the personalized sentiment classifier. Our results indicate the effectiveness of enriching models with social contexts.

In Chapter 6, we addressed the fourth research question of this thesis. Our goal was to explore the effect of detecting polarity at the topic level on sentiment classification performance. Discovering the polarity at this level is more appealing in some cases, for example, a given tweet without enough words to infer its polarity. To achieve this goal, we employed the power of utilizing the information at the topic level into the sentiment analysis task, and proposed a framework for Twitter sentiment analysis based on a deep learning approach utilizing awareness of topic-level information.

The framework consists of four main components, each one dealing with the task at a different level. The first component targets the sentiment analysis task at the tweet level. The second deals with the task at the word level within the topical context. The third component operates on the character level within the topical context. The last component provides the framework with the overall sentiment polarity of all topics in a given dataset across the different classes. Our results indicate that incorporating topical context information in a

Twitter sentiment classifier indeed leads to an improvement compared to baseline models dealing only with the textual content of tweets.

Some limitations we encountered in the current study point out interesting directions for future work. Our approaches were intended to work in a static (off-line) environment. This means Twitter sentiment analysis is done on fixed datasets. However, Twitter as one of the social media platforms that works in a dynamic (streaming) environment, where tweets are created, posted, and modified in a timely manner. This gives motivation to adapt our approaches to deal with streaming environment as a potential future direction of this research.

Another direction that would be interesting to investigate would be the contributions of the produced list of features for non-binary sentiment classification tasks. In future work, we also plan to explore other neural network based learning architectures and apply them to sentiment analysis tasks. In addition, we would like to implement our approaches for other natural languages, such as Arabic.

As future work, we also aim to investigate extracting the emotional state of the users, not only from their timeline in Twitter, but also across other social media platforms such as Facebook and Instagram. We further plan to contribute to data sets by collecting and annotating tweets that have emotions, since there is currently a lack of such data.

BIBLIOGRAPHY

[1] A. S. M. Alharbi and E. de Doncker, "Enhance a Deep Neural Network Model for Twitter Sentiment Analysis by Incorporating User Behavioral Information", in, ser. Lecture Notes in Computer Science, D.-S. Huang, V. Bevilacqua, P. Premaratne, and P. Gupta, Eds., vol. 10361, Cham: Springer International Publishing, 2017, pp. 81–88, ISBN: 978-3-319-63308-4. DOI: `10.1007/978-3-319-63309-1_8`. [Online]. Available: `http://link.springer.com/10.1007/978-3-319-63309-1http://link.springer.com/10.1007/978-3-319-63309-1{\_}8`.

[2] A. S. M. Alharbi and E. de Doncker, "Emotional Awareness based Classification Model for Twitter Sentiment Analysis using a Deep Neural Network", in *Proceedings of the 21st International Conference on Artificial Intelligence ICAI'19*, CSREA Press, 2019, pp. 142–145, ISBN: 1-60132-501-0.

[3] A. S. M. Alharbi and E. de Doncker, "Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information", *Cognitive Systems Research*, vol. 54, pp. 50–61, 2019, ISSN: 13890417. DOI: `10.1016/j.cogsys.2018.10.001`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S1389041718300482`.

[4] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent Twitter Sentiment Classification", *Computational Linguistics*, pp. 151–160, 2011. [Online]. Available: `http://www.aclweb.org/anthology/P11-1016`.

[5] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment in Twitter events", *Journal of the American Society for Information Science and Technology*, vol. 62, no. 2, pp. 406–418, 2011, ISSN: 15322882. DOI: `10.1002/asi.21462`. arXiv: `0912.3882`.

[6] J. Lin, W. Mao, and D. D. Zeng, "Personality-based refinement for sentiment classification in microblog", *Knowledge-Based Systems*, vol. 132, pp. 204–214, 2017, ISSN: 09507051. DOI: `10.1016/j.knosys.2017.06.031`. [Online]. Available: `http://dx.doi.org/10.1016/j.knosys.2017.06.031http://linkinghub.elsevier.com/retrieve/pii/S0950705117303118`.

[7] J. W. Pennebaker, M. R. Mehl, and K. G. Niederhoffer, "Psychological Aspects of Natural Language Use: Our Words, Our Selves", *Annual Review of Psychology*, vol. 54, no. 1, pp. 547–577, 2003, ISSN: 0066-4308. DOI: `10.1146/annurev.psych.54.101601.145041`. [Online]. Available: `http://www.annualreviews.org.ezproxy.library.ubc.ca/doi/abs/10.1146/annurev.psych.54.101601.145041http://www.annualreviews.org/doi/10.1146/annurev.psych.54.101601.145041`.

[8] M. McPherson, L. Smith-Lovin, and J. M. Cook, “Birds of a Feather: Homophily in Social Networks”, *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001, ISSN: 0360-0572. DOI: 10.1146/annurev.soc.27.1.415. arXiv: 9702163 [cond-mat]. [Online]. Available: http://www.jstor.org/stable/2678628http://www.annualreviews.org/doi/10.1146/annurev.soc.27.1.415.

[9] A. D. I. Kramer, J. E. Guillory, and J. T. Hancock, “Experimental evidence of massive-scale emotional contagion through social networks”, *Proceedings of the National Academy of Sciences*, vol. 111, no. 24, pp. 8788–8790, 2014, ISSN: 0027-8424. DOI: 10.1073/pnas.1320040111. [Online]. Available: http://www.pnas.org/cgi/doi/10.1073/pnas.1320040111.

[10] X. Hu, L. Tang, J. Tang, and H. Liu, “Exploiting social relations for sentiment analysis in microblogging”, in *Proceedings of the sixth ACM international conference on Web search and data mining - WSDM ’13*, vol. 1, New York, New York, USA: ACM Press, 2013, p. 537, ISBN: 9781450318693. DOI: 10.1145/2433396.2433465. [Online]. Available: http://dl.acm.org/citation.cfm?id=2433396.2433465http://dl.acm.org/citation.cfm?doid=2433396.2433465.

[11] K. Yongyos and K. Shirai, “Incorporating an Implicit and Explicit Similarity Network for User-Level Sentiment Classification of Microblogging”, in *PRICAI 2016: Trends in Artificial Intelligence: 14th Pacific Rim International Conference on Artificial Intelligence*, 2016, pp. 180–192, ISBN: 978-3-319-42910-6. DOI: 10.1007/978-3-319-42911-3. [Online]. Available: http://link.springer.com/10.1007/978-3-319-42911-3.

[12] J. Read, “Using emoticons to reduce dependency in machine learning techniques for sentiment classification”, in *Proceedings of the ACL Student Research Workshop on - ACL ’05*, Morristown, NJ, USA: Association for Computational Linguistics, 2005, p. 43, ISBN: 1932432515. DOI: 10.3115/1628960.1628969. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1628960.1628969.

[13] L. Barbosa and J. Feng, “Robust Sentiment Detection on Twitter from Biased and Noisy Data”, in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, Beijing, China, 2010, pp. 36–44, ISBN: 1012341234. DOI: 10.1016/j.sedgeo.2006.07.004. arXiv: 1702.08388. [Online]. Available: http://dl.acm.org/citation.cfm?id=1944566.1944571.

[14] A. Bifet and E. Frank, “Sentiment Knowledge Discovery in Twitter Streaming Data”, in *Proceedings of the 13th international conference on Discovery science*, Canberra, Australia, 2010, pp. 1–15, ISBN: 978-3-642-16183-4. DOI: 10.1007/978-3-642-16183-4. [Online]. Available: http://link.springer.com/10.1007/978-3-642-16184-1{_}1.

[15] A. Pak and P. Paroubek, “Twitter as a Corpus for Sentiment Analysis and Opinion Mining”, in *Seventh Conference on International Language Resources and Evaluation*, 2010, pp. 1320–1326.

[16] J. Cigarrán, Á. Castellanos, and A. García-Serrano, "A step forward for Topic Detection in Twitter: An FCA-based Approach", *Expert Systems with Applications*, vol. 57, pp. 21–36, 2016, ISSN: 09574174. DOI: 10.1016/j.eswa.2016.03.011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0957417416301038.

[17] A. Giachanou and F. Crestani, "Like It or Not: A Survey of Twitter Sentiment Analysis Methods", *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–41, 2016, ISSN: 0360-0300. DOI: 10.1145/2938640. [Online]. Available: http://dx.doi.org/10.1145/2938640.

[18] C. Baziotis, N. Pelekis, and C. Doulkeridis, "DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis", in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 747–754. [Online]. Available: http://www.aclweb.org/anthology/S17-2126.

[19] A. Bella, C. Ferri, J. Hernandez-Orallo, and M. J. Ramirez-Quintana, "Quantification via Probability Estimators", in *2010 IEEE International Conference on Data Mining*, IEEE, 2010, pp. 737–742, ISBN: 978-1-4244-9131-5. DOI: 10.1109/ICDM.2010.75. [Online]. Available: http://ieeexplore.ieee.org/document/5694031/.

[20] G. Paltoglou and A. Giachanou, "Opinion Retrieval: Searching for Opinions in Social Media", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8830, 2014, pp. 193–214, ISBN: 10.1007/978-3-319-12511-4_10. DOI: 10.1007/978-3-319-12511-4_10. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84921407090{\&}partnerID=40{\&}md5=ab1923e2f9e3d1e308d0f2cdd664a480http://link.springer.com/10.1007/978-3-319-12511-4{_}10.

[21] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis", *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008, ISSN: 1554-0669. DOI: 10.1561/1500000001. [Online]. Available: http://www.nowpublishers.com/article/Details/INR-001.

[22] B. Liu, *Sentiment Analysis Mining Opinions, Sentiment, and Emotions*. Cambridge University Press, 2015, p. 367, ISBN: 978-1-107-01789-4.

[23] D. Das and S. Bandyopadhyay, "Emotion Analysis on Social Media: Natural Language Processing Approaches and Applications", *Spring-Verlag Wien*, pp. 19–37, 2014. DOI: 10.1007/978-3-7091-1340-0. [Online]. Available: http://link.springer.com/10.1007/978-3-7091-1340-0.

[24] D. Maynard and A. Funk, "Automatic detection of political opinions in tweets", *CEUR Workshop Proceedings*, vol. 718, pp. 81–92, 2011, ISSN: 16130073. DOI: 10.1007/978-3-642-25953-1_8.

[25] S.-M. Kim, S.-M. Kim, E. Hovy, and E. Hovy, "Determining the sentiment of opinions", *Proceedings of the 20th international conference on Computational Linguistics - COLING '04*, 1367–es, 2004. DOI: 10.3115/1220355.1220555. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1220355.1220555.

[26] A. Muhammad, N. Wiratunga, and R. Lothian, "A Hybrid Sentiment Lexicon for Social Media Mining", in *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, IEEE, 2014, pp. 461–468, ISBN: 978-1-4799-6572-4. DOI: 10.1109/ICTAI.2014.76. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6984512.

[27] A. B. Goldberg and X. Zhu, "Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization", in *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, Morristown, NJ, USA: Association for Computational Linguistics, 2006, pp. 45–52. DOI: 10.3115/1654758.1654769. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1654758.1654769.

[28] B. Pang and L. Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts", in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*, Barcelona, Spain: Association for Computational Linguistics, 2004, pp. 271–278, ISBN: 9780140447972. DOI: 10.3115/1218955.1218990. arXiv: 0409058 [cs]. [Online]. Available: http://arxiv.org/abs/cs/0409058http://portal.acm.org/citation.cfm?doid=1218955.1218990http://www.ncbi.nlm.nih.gov/pubmed/25246403http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4249520http://aac.asm.org/lookup/doi/10.1128/AAC.03728-14.

[29] P. Santidhanyaroj, T. A. Khan, C. M. Gelowitz, and L. Benedicenti, "A sentiment analysis prototype system for social network data", in *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, IEEE, 2014, pp. 1–5, ISBN: 978-1-4799-3101-9. DOI: 10.1109/CCECE.2014.6900951. [Online]. Available: http://ieeexplore.ieee.org/document/6900951/.

[30] H. Isah, P. Trundle, and D. Neagu, "Social Media Analysis for Product Safety using Text Mining and Sentiment Analysis", in *14th UK Workshop on Computational Intelligence (UKCI)*, Bradford, UK, 2014, pp. 1–7, ISBN: 9781479955381.

[31] X. Zhou, X. Tao, J. Yong, and Z. Yang, "Sentiment Analysis on Tweets for Social Events", in *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, IEEE, 2013, pp. 557–562, ISBN: 978-1-4673-6085-2. DOI: 10.1109/CSCWD.2013.6581022. [Online]. Available: http://ieeexplore.ieee.org/document/6581022/.

[32] P. Stone, D. Dunphy, and M. Smith, *The General Inquirer: A Computer Approach to Content Analysis*. The MIT Press, 1966.

[33] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining", *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, vol. 10, pp. 2200–2204, 2010. DOI: citeulike-article-id:9238846.

[34] C. Strapparava and A. Valitutti, "WordNet-Affect: an affective extension of WordNet", in *Proceedings of the 4th International Conference on Language Resources and Evaluation*, vol. 4, 2004, pp. 1083–1086, ISBN: 2-9517408-1-6. [Online]. Available: http://scholar.google.com/scholar?hl=en{\&}btnG=Search{\&}q=intitle:WordNet-Affect:+an+Affective+Extension+of+WordNet{\#}0{\%}5Cnhttp://wndomains.fbk.eu/publications/lrec2004.pdf.

[35] G. Paltoglou and M. Thelwall, "Twitter, MySpace, Digg", *ACM Transactions on Intelligent Systems and Technology*, vol. 3, no. 4, pp. 1–19, 2012, ISSN: 21576904. DOI: 10.1145/2337542.2337551.

[36] G. Paltoglou, S. Gobron, M. Skowron, M. Thelwall, and D Thalmann, "Sentiment analysis of informal textual communication in cyberspace", *In Proc Engage 2010 Springer LNCS StateoftheArt Survey*, pp. 13–25, 2010. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.173.3880{\&}rep=rep1{\&}type=pdf.

[37] E. Cambria, B. B. Schuller, Y. Xia, and C. Havasi, "New Avenues in Opinion Mining and Sentiment Analysis", *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 15–21, 2013, ISSN: 1541-1672. DOI: 10.1109/MIS.2013.30. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6468032.

[38] A. Jurek, M. D. Mulvenna, and Y. Bi, "Improved lexicon-based sentiment analysis for social media analytics", *Security Informatics*, vol. 4, no. 1, p. 9, 2015, ISSN: 2190-8532. DOI: 10.1186/s13388-015-0024-x. [Online]. Available: http://www.security-informatics.com/content/4/1/9.

[39] Z. Hu, J. Hu, W. Ding, and X. Zheng, "Review Sentiment Analysis Based on Deep Learning", in *IEEE 12th International Conference on e-Business Engineering*, 2015, pp. 87–94, ISBN: 978-1-4673-8002-7. DOI: 10.1109/ICEBE.2015.24. [Online]. Available: http://www.computer.org/csdl/proceedings/icebe/2015/8002/00/8002a087-abs.html.

[40] D.-S. Huang, "A Constructive Approach for Finding Arbitrary Roots of Polynomials by Neural Networks", *IEEE Transactions on Neural Networks*, vol. 15, no. 2, pp. 477–491, 2004, ISSN: 1045-9227. DOI: 10.1109/TNN.2004.824424. [Online]. Available: http://ieeexplore.ieee.org/document/1288251/.

[41] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification", in *Association for Computational Linguistics*, 2014, pp. 1555–1565, ISBN: 9781937284725.

[42] S. M. Mohammad, S. Kiritchenko, and X. Zhu, “NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets”, in *Proceedings of the Seventh International Workshop on Semantic Evaluation, SemEval-2013*, vol. 2, 2013, pp. 321–327. arXiv: `arXiv:1308.6242v1`.

[43] A. Severyn and A. Moschitti, “Twitter Sentiment Analysis with Deep Convolutional Neural Networks”, in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR ’15*, 2015, pp. 959–962, ISBN: 9781450336215. DOI: `10.1145/2766462.2767830`. [Online]. Available: `http://dl.acm.org/citation.cfm?doid=2766462.2767830`.

[44] Q. You, J. Luo, H. Jin, and J. Yang, “Joint Visual-Textual Sentiment Analysis with Deep Neural Networks”, in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1071–1074, ISBN: 9781450334594. DOI: `10.1145/2733373.2806284`. arXiv: `arXiv:1509.06041v1`.

[45] C. J. Hutto and E. Gilbert, “Vader: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text”, in *The 8th International AAAI Conference on Weblogs and Social Media*, 2014, pp. 216–225, ISBN: 9781577356578. [Online]. Available: `http://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8109{\%}5Cnhttp://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf`.

[46] R. Paulus, R. Socher, and C. D. Manning, “Global Belief Recursive Neural Networks”, in *Advances in Neural Information Processing Systems*, vol. 2, 2014, pp. 2888–2896. [Online]. Available: `http://papers.nips.cc/paper/5275-global-belief-recursive-neural-networks`.

[47] W. De Mulder, S. Bethard, and M.-F. Moens, “A survey on the application of recurrent neural networks to statistical language modeling”, *Computer Speech & Language*, vol. 30, no. 1, pp. 61–98, 2015, ISSN: 08852308. DOI: `10.1016/j.csl.2014.09.005`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S088523081400093Xhttp://linkinghub.elsevier.com/retrieve/pii/S088523081400093X`.

[48] M. Huang, Y. Cao, and C. Dong, “Modeling Rich Contexts for Sentiment Classification with LSTM”, *arXiv:1605.01478*, 2016. arXiv: `1605.01478`. [Online]. Available: `http://arxiv.org/abs/1605.01478`.

[49] F. H. Khan, U. Qamar, and S. Bashir, “SentiMI: Introducing point-wise mutual information with SentiWordNet to improve sentiment polarity detection”, *Applied Soft Computing Journal*, vol. 39, pp. 140–153, 2016, ISSN: 15684946. DOI: `10.1016/j.asoc.2015.11.016`. [Online]. Available: `http://dx.doi.org/10.1016/j.asoc.2015.11.016`.

[50] G. Katz, N. Ofek, and B. Shapira, "ConSent: Context-based sentiment analysis", *Knowledge-Based Systems*, vol. 84, pp. 162–178, 2015, ISSN: 09507051. DOI: 10.1016/j.knosys.2015.04.009. [Online]. Available: http://dx.doi.org/10.1016/j.knosys.2015.04.009.

[51] K. Tsagkalidou, V. Koutsonikola, A. Vakali, and K. Kafetsios, "Emotional Aware Clustering on Micro-blogging Sources", in *Affective Computing and Intelligent Interaction: 4th International Conference, ACII*, PART 1, vol. 6974, Memphis, TN, USA, 2011, pp. 387–396, ISBN: 9783642245992. DOI: 10.1007/978-3-642-24600-5_42. [Online]. Available: http://link.springer.com/10.1007/978-3-642-24600-5{_}42.

[52] A. Gill, R. French, D. Gergle, and J. Oberlander, "Identifying Emotional Characteristics from Short Blog Texts", in *30th Annual Conference of the Cognitive Science Society*, Washington DC, 2008, pp. 2237–2242, ISBN: 9781605580111. [Online]. Available: http://www.cogsci.rpi.edu/CSJarchive/proceedings/2008/pdfs/p2237.pdf.

[53] A. Mudinas, D. Zhang, and M. Levene, "Combining lexicon and learning based approaches for concept-level sentiment analysis", in *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining - WISDOM '12*, 2012, pp. 1–8, ISBN: 9781450315432. DOI: 10.1145/2346676.2346681. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2346676.2346681.

[54] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced Sentiment Learning Using Twitter Hashtags and Smileys", in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, 2010, pp. 241–249. [Online]. Available: http://dl.acm.org/citation.cfm?id=1944566.1944594.

[55] D. Tang, F. Wei, B. Qin, M. Zhou, and T. Liu, "Building Large-Scale Twitter-Specific Sentiment Lexicon: a Representation Learning Approach", in *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, 2014, pp. 172–182, ISBN: 9781941643266.

[56] 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, 2016, ISSN: 1041-4347. DOI: 10.1109/TKDE.2015.2489653. [Online]. Available: http://ieeexplore.ieee.org/document/7296633/.

[57] L. Gui, Y. Zhou, R. Xu, Y. He, and Q. Lu, "Learning representations from heterogeneous network for sentiment classification of product reviews", *Knowledge-Based Systems*, vol. 124, pp. 34–45, 2017, ISSN: 09507051. DOI: 10.1016/j.knosys.2017.02.030. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0950705117301144.

[58] S. Zhang, X. Zhang, and J. Chan, "Language-Independent Twitter Classification Using Character-Based Convolutional Networks", in *Advanced Data Mining and Applications*, ser. Lecture Notes in Computer Science, G. Cong, W.-C. Peng, W. E. Zhang, C. Li, and A. Sun, Eds., vol. 10604, Cham: Springer International Publishing, 2017, pp. 413–425, ISBN: 978-3-319-69178-7. DOI: 10.1007/978-3-319-69179-4_29. [Online]. Available: http://link.springer.com/10.1007/978-3-319-69179-4http://link.springer.com/10.1007/978-3-319-69179-4{_}29.

[59] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality", in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, vol. 2, 2013, pp. 3111–3119, ISBN: 2150-8097. DOI: 10.1162/jmlr.2003.3.4-5.951. arXiv: 1310.4546. [Online]. Available: http://www.crossref.org/deleted{_}DOI.htmlhttp://arxiv.org/abs/1310.4546.

[60] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors", in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 238–247, ISBN: 9781937284725. DOI: 10.3115/v1/P14-1023. arXiv: arXiv:1011.1669v3. [Online]. Available: http://aclweb.org/anthology/P14-1023.

[61] S. Lai, K. Liu, S. He, and J. Zhao, "How to Generate a Good Word Embedding", *IEEE Intelligent Systems*, vol. 31, no. 6, pp. 5–14, 2016, ISSN: 1541-1672. DOI: 10.1109/MIS.2016.45. [Online]. Available: https://arxiv.org/pdf/1507.05523.pdfhttp://ieeexplore.ieee.org/document/7478417/.

[62] M. Li, Q. Lu, Y. Long, and L. Gui, "Inferring Affective Meanings of Words from Word Embedding", *IEEE Transactions on Affective Computing*, vol. 8, no. 4, pp. 443–456, 2017, ISSN: 1949-3045. DOI: 10.1109/TAFFC.2017.2723012. [Online]. Available: http://ieeexplore.ieee.org/document/7968355/.

[63] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation", in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 1532–1543, ISBN: 9781937284961. DOI: 10.3115/v1/D14-1162. arXiv: 1504.06654. [Online]. Available: http://aclweb.org/anthology/D14-1162.

[64] O. Levy, Y. Goldberg, and I. Dagan, "Improving Distributional Similarity with Lessons Learned from Word Embeddings", *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 211–225, 2015, ISSN: 1472-6947. [Online]. Available: https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/570http://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/1472-6947-15-S2-S2http://arxiv.org/abs/1103.0398http://www.ncbi.nlm.nih.gov/pubmed/26099735http://www.pubmedcentral.nih..

[65] S. Poria, H. Peng, A. Hussain, N. Howard, and E. Cambria, "Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis", *Neurocomputing*, vol. 261, pp. 217–230, 2017, ISSN: 09252312. DOI: 10.1016/j.neucom.2016.09.117. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0925231217302023.

[66] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis", in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2011, pp. 142–150, ISBN: 9781932432879. DOI: 978-1-932432-87-9.

[67] C. N. dos Santos and M. Gatti, "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts", in *The 25th International Conference on Computational Linguistics*, 2014, pp. 69–78.

[68] S. Vosoughi, P. Vijayaraghavan, and D. Roy, "Tweet2Vec : Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder", in *The 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 1041–1044, ISBN: 9781450340694.

[69] J. Wehrmann, W. Becker, H. E. L. Cagnini, and R. C. Barros, "A Character-based Convolutional Neural Network for Language-Agnostic Twitter Sentiment Analysis", in *2017 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2017, pp. 2384–2391, ISBN: 978-1-5090-6182-2. DOI: 10.1109/IJCNN.2017.7966145. [Online]. Available: http://ieeexplore.ieee.org/document/7966145/.

[70] I. Mozetič, M. Grčar, and J. Smailović, "Multilingual Twitter Sentiment Classification: The Role of Human Annotators", *PLOS ONE*, vol. 11, no. 5, M. Perc, Ed., e0155036, 2016, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0155036. arXiv: 1602.07563. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0155036.

[71] A. C. E. Lima and L. N. de Castro, "A multi-label, semi-supervised classification approach applied to personality prediction in social media", *Neural Networks*, vol. 58, pp. 122–130, 2014, ISSN: 08936080. DOI: 10.1016/j.neunet.2014.05.020. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2014.05.020http://linkinghub.elsevier.com/retrieve/pii/S0893608014001282.

[72] A. C. E. S. Lima and L. N. de Castro, "Predicting Temperament from Twitter Data", in *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, IEEE, 2016, pp. 599–604, ISBN: 978-1-4673-8985-3. DOI: 10.1109/IIAI-AAI.2016.239. [Online]. Available: http://ieeexplore.ieee.org/document/7557682/.

[73] M. Roshanaei and S. Mishra, "Studying the attributes of users in Twitter considering their emotional states", *Social Network Analysis and Mining*, vol. 5, no. 1, p. 34, 2015, ISSN: 1869-5450. DOI: 10.1007/s13278-015-0278-9. [Online]. Available: http://link.springer.com/10.1007/s13278-015-0278-9.

[74] J. Tang, Y. Zhang, J. Sun, and J. Rao, "Quantitative study of individual emotional states in social networks", *Affective Computing*, vol. 3, no. 2, pp. 132–144, 2012, ISSN: 1949-3045. DOI: 10.1109/T-AFFC.2011.23. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=5959157.

[75] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines", in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814, ISBN: 9781605589077.

[76] 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, ISSN: 15322882. DOI: 10.1002/asi.21416. [Online]. Available: http://doi.wiley.com/10.1002/asi.21416.

[77] V. Hangya and R. Farkas, "A comparative empirical study on social media sentiment analysis over various genres and languages", *Artificial Intelligence Review*, pp. 1–21, 2016, ISSN: 15737462. DOI: 10.1007/s10462-016-9489-3.

[78] S. Huang, Z. Niu, and C. Shi, "Automatic construction of domain-specific sentiment lexicon based on constrained label propagation", *Knowledge-Based Systems*, vol. 56, pp. 191–200, 2014, ISSN: 09507051. DOI: 10.1016/j.knosys.2013.11.009. [Online]. Available: http://dx.doi.org/10.1016/j.knosys.2013.11.009.

[79] J. Bross and H. Ehrig, "Automatic construction of domain and aspect specific sentiment lexicons for customer review mining", *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*, pp. 1077–1086, 2013. DOI: 10.1145/2505515.2505574. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2505515.2505574.

[80] S. Sedhai and A. Sun, "HSpam14 : A Collection of 14 Million Tweets for Hashtag-Oriented Spam Research", pp. 223–232, 2015. DOI: 10.1145/2766462.2767701.

[81] E. MARTÍNEZ-CÁMARA, M. T. MARTÍN-VALDIVIA, L. A. UREÑA-LÓPEZ, and A. R. MONTEJO-RÁEZ, "Sentiment analysis in Twitter", *Natural Language Engineering*, vol. 20, no. 01, pp. 1–28, 2014, ISSN: 1351-3249. DOI: 10.1017/S1351324912000332. [Online]. Available: http://www.journals.cambridge.org/abstract{_}S1351324912000332.

[82] P. Chikersal, S. Poria, and E. Cambria, "SeNTU : Sentiment Analysis of Tweets by Combining a Rule-based Classifier with Supervised Learning", *SemEval2015*, no. SemEval, pp. 647–651, 2015.

[83] B. Liu, *Sentiment Analysis and Opinion Mining*, 1. 2012, vol. 5, pp. 1–167, ISBN: 9781608458844. DOI: 10.2200/S00416ED1V01Y201204HLT016.

[84] S. Guha, "Sentibase : Sentiment Analysis in Twitter on a Budget", in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 590–594.

[85] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision", *Processing*, vol. 150, no. 12, pp. 1–6, 2009, ISSN: 00370738. DOI: 10.1016/j.sedgeo.2006.07.004. [Online]. Available: http://www.stanford.edu/{~}alecmgo/papers/TwitterDistantSupervision09.pdf.

[86] X. Ji, S. A. Chun, Z. Wei, and J. Geller, "Twitter sentiment classification for measuring public health concerns", *Social Network Analysis and Mining*, vol. 5, no. 1, 2015, ISSN: 1869-5450. DOI: 10.1007/s13278-015-0253-5. [Online]. Available: http://link.springer.com/10.1007/s13278-015-0253-5.

[87] B. Pang, L. Lee, H. Rd, and S. Jose, "Thumbs up ? Sentiment Classification using Machine Learning Techniques", *EMNLP '02 Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, no. July, pp. 79–86, 2002, ISSN: 1554-0669. DOI: 10.3115/1118693.1118704. arXiv: 0205070 [cs].

[88] J. R. Quinlan, "Induction of decision trees", *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986, ISSN: 0885-6125. DOI: 10.1007/BF00116251. [Online]. Available: http://link.springer.com/10.1007/BF00116251.

[89] P. Gonçalves, M. Araújo, F. Benevenuto, and M. Cha, "Comparing and Combining Sentiment Analysis Methods", in *Proceedings of the first ACM conference on Online social networks - COSN '13*, New York, New York, USA: ACM Press, 2013, pp. 27–38, ISBN: 9781450320849. DOI: 10.1145/2512938.2512951. arXiv: arXiv:1406.0032v1. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2512938.2512951.

[90] 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, ISSN: 09507051. DOI: 10.1016/j.knosys.2015.06.015. [Online]. Available: http://dx.doi.org/10.1016/j.knosys.2015.06.015http://linkinghub.elsevier.com/retrieve/pii/S0950705115002336.

[91] X. Wang, Y. Liu, C. SUN, B. Wang, and X. Wang, "Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory", in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 1343–1353, ISBN: 9781941643723. [Online]. Available: http://www.aclweb.org/anthology/P15-1130.

[92] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "SemEval-2018 Task 1: Affect in Tweets", in *Proceedings of The 12th International Workshop on Semantic Evaluation*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2018, pp. 1–17. DOI: 10.18653/v1/S18-1001. [Online]. Available: http://aclweb.org/anthology/S18-1001.

[93] S. Xu, H. Liang, and T. Baldwin, “UNIMELB at SemEval-2016 Tasks 4A and 4B: An Ensemble of Neural Networks and a Word2Vec Based Model for Sentiment Classification”, in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 183–189, ISBN: 9781941643952. DOI: `10.18653/v1/S16-1027`. [Online]. Available: `http://aclweb.org/anthology/S16-1027`.

[94] S. Ruder, P. Ghaffari, and J. G. Breslin, “INSIGHT-1 at SemEval-2016 Task 4: Convolutional Neural Networks for Sentiment Classification and Quantification”, in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 178–182, ISBN: 9781941643952. DOI: `10.18653/v1/S16-1026`. arXiv: `1609.02746`. [Online]. Available: `http://arxiv.org/abs/1609.02746http://aclweb.org/anthology/S16-1026`.

[95] M. Rouvier and B. Favre, “SENSEI-LIF at SemEval-2016 Task 4: Polarity embedding fusion for robust sentiment analysis”, in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 202–208, ISBN: 9781941643952. DOI: `10.18653/v1/S16-1030`. [Online]. Available: `http://www.aclweb.org/anthology/S16-1030http://aclweb.org/anthology/S16-1030`.

[96] N. Phan, D. Dou, B. Piniewski, and D. Kil, “A deep learning approach for human behavior prediction with explanations in health social networks: social restricted Boltzmann machine (SRBM+)”, *Social Network Analysis and Mining*, vol. 6, no. 1, p. 79, 2016, ISSN: 1869-5450. DOI: `10.1007/s13278-016-0379-0`. [Online]. Available: `http://link.springer.com/10.1007/s13278-016-0379-0`.

[97] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li, “User-level sentiment analysis incorporating social networks”, in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’11*, New York, New York, USA: ACM Press, 2011, p. 1397, ISBN: 9781450308137. DOI: `10.1145/2020408.2020614`. arXiv: `1109.6018`. [Online]. Available: `http://doi.acm.org/10.1145/2020408.2020614{\%}5Cnhttp://dl.acm.org/citation.cfm?doid=2020408.2020614http://dl.acm.org/citation.cfm?doid=2020408.2020614`.

[98] A. Sapountzi and K. E. Psannis, “Social networking data analysis tools & challenges”, *Future Generation Computer Systems*, vol. 86, pp. 893–913, 2018, ISSN: 0167739X. DOI: `10.1016/j.future.2016.10.019`. [Online]. Available: `http://dx.doi.org/10.1016/j.future.2016.10.019https://linkinghub.elsevier.com/retrieve/pii/S0167739X1630423X`.

[99] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis: A survey”, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, no. January,

e1253, 2018, ISSN: 19424787. DOI: 10.1002/widm.1253. arXiv: 1801.07883. [Online]. Available: http://doi.wiley.com/10.1002/widm.1253.

[100] H. Li, X. Li, D. Caragea, and C. Caragea, "Comparison of Word Embeddings and Sentence Encodings as Generalized Representations for Crisis Tweet Classification Tasks", in *Proceedings of the ISCRAM Asian Pacific 2018 Conference (ISCRAM Asian Pacific 2018)*, 2018, pp. 1–13. [Online]. Available: http://aidr.qcri.org/.

[101] Z. Zhang, Q. Zhou, and H. Wu, "NLPZZX at SemEval-2018 Task 1: Using Ensemble Method for Emotion and Sentiment Intensity Determination", in *Proceedings of The 12th International Workshop on Semantic Evaluation*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2018, pp. 116–122. DOI: 10.18653/v1/S18-1015. [Online]. Available: http://aclweb.org/anthology/S18-1015.

[102] M Zhang, Y Zhang, and D. Vo, "Gated Neural Networks for Targeted Sentiment Analysis", in *30th Conference on Artificial Intelligence (AAAI 2016)*, 2016, pp. 3087–3093, ISBN: 9781577357605. [Online]. Available: http://zhangmeishan.github.io/targeted-sentiment.pdf.

[103] X. Wang, W. Jiang, and Z. Luo, "Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts", in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2428–2437.

[104] B. Dhingra, Z. Zhou, D. Fitzpatrick, M. Muehl, and W. W. Cohen, "Tweet2Vec: Character-Based Distributed Representations for Social Media", in *The 54th Annual Meeting of the Association for Computational Linguistics*, vol. 2, 2016, pp. 1–6. arXiv: 1605.03481. [Online]. Available: http://arxiv.org/abs/1605.03481.

[105] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude", *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[106] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge, "Twitter Polarity Classification with Label Propagation over Lexical Links and the Follower Graph", in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 53–63, ISBN: 9781937284138.

[107] H. Saif, M. Fernandez, Y. He, and H. Alani, "Evaluation Datasets for Twitter Sentiment Analysis: A survey and a new dataset, the STS-Gold", *Proceedings of the First International Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives from AI (ESSEM 2013)*, pp. 9–21, 2013, ISSN: 16130073. [Online]. Available: http://www.researchgate.net/publication/259261310{_}Evaluation{_}Datasets{_}for{_}Twitter{_}Sentiment{_}Analysis.{_}A{_}survey{_}and{_}a{_}new{_}dataset{_}the{_}STS-Gold/file/72e7e52a9c1e790a77.pdf.

[108] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences", in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 655–665, ISBN: 9781937284725. DOI: 10.3115/v1/P14-1062. [Online]. Available: http://aclweb.org/anthology/P14-1062.

[109] C. C. Aggarwal, *Data Classification: Algorithms and Applications*. 2014, p. 707, ISBN: 1466586745. [Online]. Available: https://books.google.com/books?id=gJhBBAAAQBAJ{\&}pgis=1.

[110] C. D. Manning, P. Raghavan, and H. Schutze, "Scoring, term weighting, and the vector space model", in *Introduction to Information Retrieval*, c, Cambridge: Cambridge University Press, 2009, pp. 100–123, ISBN: 978-0-521-86571-5. DOI: 10.1017/CBO9780511809071.007. [Online]. Available: http://nlp.stanford.edu/IR-book/pdf/06vect.pdfhttps://www.cambridge.org/core/product/identifier/CBO9780511809071A040/type/book{_}part.

[111] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "SemEval-2016 Task 4: Sentiment Analysis in Twitter", in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 1–18, ISBN: 0361-803X (Print)\r0361-803X (Linking). DOI: 10.18653/v1/S16-1001. arXiv: arXiv:0906.0612. [Online]. Available: http://aclweb.org/anthology/S17-2088http://aclweb.org/anthology/S16-1001.

[112] S. Rosenthal, P. Nakov, S. Kiritchenko, S. M. Mohammad, A. Ritter, and V. Stoyanov, "SemEval-2015 Task 10: Sentiment Analysis in Twitter", in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 451–463. [Online]. Available: http://alt.qcri.org/semeval2015/cdrom/pdf/SemEval078.pdf.

[113] S. Symeonidis, D. Effrosynidis, J. Kordonis, and A. Arampatzis, "DUTH at SemEval-2017 Task 4: A Voting Classification Approach for Twitter Sentiment Analysis", in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2017, pp. 704–708. DOI: 10.18653/v1/S17-2117. [Online]. Available: http://aclweb.org/anthology/S17-2117.

[114] W. Maoquan, C. Shiyun, X. Yufei, and Z. Lu, "EICA at SemEval-2017 Task 4: A Simple Convolutional Neural Network for Topic-based Sentiment Classification", in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2017, pp. 737–740. DOI: 10.18653/v1/S17-2124. [Online]. Available: http://aclweb.org/anthology/S17-2124.

[115] N. Karpov, "NRU-HSE at SemEval-2017 Task 4: Tweet Quantification Using Deep Learning Architecture", in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2017, pp. 683–688. DOI: `10.18653/v1/S17-2113`. [Online]. Available: `http://aclweb.org/anthology/S17-2113`.

[116] U. Sidarenka, "PotTS at SemEval-2016 Task 4: Sentiment Analysis of Twitter Using Character-level Convolutional Neural Networks.", in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, vol. 2003, Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 230–237. DOI: `10.18653/v1/S16-1035`. [Online]. Available: `https://github.com/WladimirSidorenko/` `http://aclweb.org/anthology/S16-1035`.

[117] S. M. Jiménez-Zafra, A. Montejo-Ráez, M. Martin, and L. A. Urena Lopez, "SINAI at SemEval-2017 Task 4: User based classification", in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2017, pp. 634–639. DOI: `10.18653/v1/S17-2104`. [Online]. Available: `http://aclweb.org/anthology/S17-2104`.

[118] S. Rajendram and T. T. Mirnalinee, "SSN_MLRG1 at SemEval-2017 Task 4: Sentiment Analysis in Twitter Using Multi-Kernel Gaussian Process Classifier", in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2017, pp. 709–712. DOI: `10.18653/v1/S17-2118`. [Online]. Available: `http://aclweb.org/anthology/S17-2118`.

[119] M. Wang, B. Chu, Q. Liu, and X. Zhou, "YNUDLG at SemEval-2017 Task 4: A GRU-SVM Model for Sentiment Classification and Quantification in Twitter", in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2017, pp. 713–717, ISBN: 1147192499. DOI: `10.18653/v1/S17-2119`. [Online]. Available: `http://aclweb.org/anthology/S17-2119`.

Appendix

A. Some Source Codes Used in this Work

Code of collecting tweets using Twitter API.

```
cache = {}
for line in args.dist:
  columns = line.strip().split('\t')
  sid = columns[0]
  UserID = columns[1]

  while not sid in cache:
    try:
      TheTweet = t.statuses.show(_id=sid)
      ## The tweet
      text=TheTweet['text'].replace('\n', ' ').replace('\r', ' ')
      lang_T = TheTweet['lang']
      retweet_count = TheTweet['retweet_count']
      favourites_count = TheTweet['favorite_count']
      Tweet_created_at = TheTweet['created_at']

      ## Hashtags
      hashtags = TheTweet['entities']['hashtags']
      hashtag_count= len(hashtags)
      TheHashtagsAre=str(hashtag_count)
      if hashtag_count!=0:
        for hashtag in hashtags:
          TheHashtagsAre = ",".join([TheHashtagsAre] + [hashtag['
   text']] )

      ## About the user
      screen_name = TheTweet['user']['screen_name']
      name = TheTweet['user']['name']
      User_id_str = TheTweet['user']['id_str']
      NumberOfTweets = TheTweet['user']['statuses_count']
      friends_count = TheTweet['user']['friends_count']
      followers_count = TheTweet['user']['followers_count']
      Verified = TheTweet['user']['verified']

      cache[sid] = text.encode('utf-8')

    except TwitterError as e:
      if e.e.code == 429:
        rate = t.application.rate_limit_status()
```

```
      reset = rate['resources']['statuses']['/statuses/show/:id'][
 'reset']
      now = datetime.datetime.today()
      future = datetime.datetime.fromtimestamp(reset)
      seconds = (future-now).seconds+1
      if seconds < 10000:
        sys.stderr.write("Rate limit exceeded, sleeping for %s
 seconds until %s\n" % (seconds, future))
        time.sleep(seconds)
    else:
      cache[sid] = 'Not Available'

  text = cache[sid]

  args.output.write("\t".join(columns + [text]+[Tweet_created_at] +
 [screen_name]+[name]+ [User_id_str]+[str(NumberOfTweets)] + [str(
 followers_count)]+[str(friends_count)] + [str(Verified)]+[lang_T
 ]+[str(retweet_count)] +[str(favourites_count)]+ [TheHashtagsAre
 ]) + '\n')
```

Gathering users Twitter timeline.

```
def get_all_tweets(user_id):
  #Twitter only allows access to a users most recent 3240 tweets
     with this method
  auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
  auth.set_access_token(access_key, access_secret)
  api = tweepy.API(auth)

  #initialize a list to save all returned tweets
  retrievedTweet = []

  #Send a request to Twitter API
  new_tweets = api.user_timeline(user_id = user_id,count=200)
  if api.rate_limit_status()['resources']['statuses']["/statuses/
     user_timeline"]["remaining"]<=25:
    print api.rate_limit_status()['resources']['statuses']["/
       statuses/user_timeline"]
    time.sleep(960)
    print api.rate_limit_status()['resources']['statuses']["/
       statuses/user_timeline"]

  retrievedTweet.extend(new_tweets)
```

```
  oldTweets = retrievedTweet[-1].id - 1

  while len(new_tweets) > 0:
    new_tweets = api.user_timeline(user_id = user_id,count=200,
       max_id=oldTweets)
    retrievedTweet.extend(new_tweets)
    oldTweets = retrievedTweet[-1].id - 1

  outtweets = [[tweet.text.encode("utf-8").replace('\n', ' ').
     replace('\r', ' '), tweet.created_at,tweet.lang , tweet.id_str]
      for tweet in retrievedTweet]

  #write the collected tweets into txt file
  with open('UsersTweets\%s.txt' % user_id, 'wb') as f:
    writer = csv.writer(f,delimiter='\t')
    writer.writerows(outtweets)
```

Deep learing model used to train each emotion.

```
embedding_dim = GLOVE_DIM
inp = Input(shape=(MAX_LEN, ))
embeddingLayer = Embedding(NB_WORDS, embedding_dim, weights=[
   emb_matrix], input_length=MAX_LEN, trainable=True)(inp)
Spat_DropoutLayer = SpatialDropout1D(0.5)(embeddingLayer)
Bi_directionalLayer= Bidirectional(GRU(100, return_sequences=True))(
   Spat_DropoutLayer)
avgPool = GlobalAveragePooling1D()(Bi_directionalLayer)
maxPool = GlobalMaxPooling1D()(Bi_directionalLayer)
conc_Layer = concatenate([avgPool, maxPool])
outp = Dense(1, activation="sigmoid")(conc_Layer)

model = Model(inputs=inp, outputs=outp)
model.compile(loss='binary_crossentropy',
       optimizer='adam',
       metrics=['accuracy'])
```

Annotation code for users with five emotions.

```
def get_annotating(fileName):
#   print fileName
  UserTimeLine = pd.read_csv(pathToUsersTimline+'/'+fileName,
                 sep="\t",
                 header=None,
                 names=['Tweet', 'CreationDate','Lang','TweetId'])
```

```
#   Removing Hashtags
  UserTimeLine['cleaned_tweet'] = np.vectorize(remove_pattern)(
   UserTimeLine['Tweet'], "\\#\\b[\\w\\-\\_]+\\b")
#   Removing Punctuations, Numbers, and Special Characters
  UserTimeLine['cleaned_tweet'] = UserTimeLine['cleaned_tweet'].str.
   replace("[^a-zA-Z#]", " ")
#   Removing Short Words_
  UserTimeLine['cleaned_tweet'] = UserTimeLine['cleaned_tweet'].
   apply(lambda x: ' '.join([w for w in x.split() if len(w)>3]))
#   preprocess_twitter.py
  UserTimeLine['cleaned_tweet']=UserTimeLine['cleaned_tweet'].apply(
   preprocess_twitter.tokenize)

#   Converting words to numbers
  X_test_seq = tk.texts_to_sequences(UserTimeLine['cleaned_tweet'])
#   Hear we padding the sequences
  X_test_seq_trunc = pad_sequences(X_test_seq, maxlen=MAX_LEN,
   padding='post')

#   Prediction
  # Optimism_Model
  y_pred_Before_Converted_Optimism=Optimism_Model.predict(
   X_test_seq_trunc,batch_size=2048, verbose=0)
  UserTimeLine['Optimism'] =y_pred_Before_Converted_Optimism
  UserTimeLine['Optimism'] = UserTimeLine['Optimism'].map(lambda p:
   1 if p >= 0.5 else 0)

  # Anger_Model
  y_pred_Before_Converted_Anger=Anger_Model.predict(X_test_seq_trunc
   ,batch_size=2048, verbose=0)
  UserTimeLine['Anger'] =y_pred_Before_Converted_Anger
  UserTimeLine['Anger'] = UserTimeLine['Anger'].map(lambda p: 1 if p
    >= 0.5 else 0)

  # Disgust_Model
  y_pred_Before_Converted_Disgust=Disgust_Model.predict(
   X_test_seq_trunc,batch_size=2048, verbose=0)
  UserTimeLine['Disgust'] =y_pred_Before_Converted_Disgust
  UserTimeLine['Disgust'] = UserTimeLine['Disgust'].map(lambda p: 1
   if p >= 0.5 else 0)

  # Joy_Model
```

```
    y_pred_Before_Converted_Joy=Joy_Model.predict(X_test_seq_trunc,
     batch_size=2048, verbose=0)
    UserTimeLine['Joy'] =y_pred_Before_Converted_Joy
    UserTimeLine['Joy'] = UserTimeLine['Joy'].map(lambda p: 1 if p >=
     0.5 else 0)

    # Sadness_Model
    y_pred_Before_Converted_Sadness=Sadness_Model.predict(
     X_test_seq_trunc,batch_size=2048, verbose=0)
    UserTimeLine['Sadness'] =y_pred_Before_Converted_Sadness
    UserTimeLine['Sadness'] = UserTimeLine['Sadness'].map(lambda p: 1
     if p >= 0.5 else 0)

#  Save the results
    UserTimeLine.to_csv(PathToSaveTheAnnotatedTweets+'/'+fileName ,sep
     ='\t',index=False)
```

The Emotional Awareness based Modle (EACM).

```
embedding_dim = GLOVE_DIM
# First input - the target tweet
inp1 = Input(shape=(MAX_LEN, ))
embeddingLayer = Embedding(NB_WORDS, embedding_dim, weights=[
   emb_matrix], input_length=MAX_LEN, trainable=True)(inp1)
s_DropoutLayer = SpatialDropout1D(0.2)(embedding_layer)
BidirectionalLayer = Bidirectional(GRU(100, return_sequences=True))(
   s_DropoutLayer)
avg_poolLayer = GlobalAveragePooling1D()(BidirectionalLayer)
max_poolLayer = GlobalMaxPooling1D()(BidirectionalLayer)
# Second input - the five emotional states of a user
inp2 = Input(shape=(5, ), name='Emotional_inputs')
concLayer = concatenate([avg_poolLayer, max_poolLayer,inp2])

outp = Dense(3, activation="softmax", name='output')(concLayer)

model = Model(inputs=[inp1,inp2], outputs=outp)
model.compile(loss='categorical_crossentropy',
       optimizer='rmsprop',
       metrics=['accuracy'])
```

Deep learning model based on the Social Interaction Aware-based Approach (SIAA).

```
embedding_dim = GLOVE_DIM
filter_sizes = [2, 3, 5]
```

```
num_filters = 256
drop = 0.3
# First input - the target tweet
inp1 = Input(shape=(MAX_LEN, ),name='Target_Tweet')
embedding_inp1= Embedding(NB_WORDS, embedding_dim, weights=[
   emb_matrix], input_length=MAX_LEN, trainable=False)(inp1)
Spatial_DropoutLayer = SpatialDropout1D(0.2)(embedding_inp1)
Bi_directionalLayer = Bidirectional(GRU(100, return_sequences=True))
   (Spatial_DropoutLayer)
avg_pool_inp1 = GlobalAveragePooling1D()(Bi_directionalLayer)
max_pool_inp1 = GlobalMaxPooling1D()(Bi_directionalLayer)

# Second input - the topic-specific representation
inp2 = Input(shape=(MaxLen_char, ), name='
   TFIDFBasedOnTopic_input_char')

embedding_inp2 = Embedding(input_dim=MaxLen_char_vocabulary_size,
           output_dim=embedding_dim,
           input_length=MaxLen_char)(inp2)
y1 = Conv1D(64, kernel_size = 2, padding = "valid",
   kernel_initializer = "he_uniform")(embedding_inp2)
avg_pool_inp2 = GlobalAveragePooling1D()(y1)
max_pool_inp2 = GlobalMaxPooling1D()(y1)

# Third input - the two vector of length 2 that represent the
   sentiment ratio of each label of the target user.
inp3 = Input(shape=(2, ), name="User_Sentiment_Raito_PLS_TLS_CLS")
y3= Dense(10,activation='tanh')(inp3)
conc_ALL = Concatenate(axis=1)([max_pool_inp1,avg_pool_inp1,
   avg_pool_inp2,max_pool_inp2,y3])
dropout = Dropout(drop)(conc_ALL)

outp = Dense(2, activation="sigmoid", name='output')((dropout))

model = Model(inputs=[inp1,inp2,inp3], outputs=outp)
model.compile(loss='categorical_crossentropy',
       optimizer='rmsprop',
       metrics=['accuracy'])
```