DISSERTATION

On

Twitter Sentiment Analysis Using Deep Learning

Thesis Submitted in the partial fulfillment of the requirements for the degree of

Master of Engineering

In

Computer Science & Engineering

Submitted By

Sudeshna Nath

Examination Roll number: M4CSE22013

Registration number: 154137 of 2020-2021

Under Guidance of

Prof. (Dr.) Kamal Sarkar

Jadavpur University

Dept. of Computer Science & Engineering

Faculty Council of Engineering and Technology

JADAVPUR UNIVERSITY

KOLKATA – 700032

2020 – 2021

Department of Computer Science & Engineering

Faculty Council of Engineering and Technology

JADAVPUR UNIVERSITY, KOLKATA – 700032

Certificate of Recommendation

This is to certify that Sudeshna Nath (Examination Roll number:M4CSE22013) has completed her dissertation entitled "Twitter Sentiment Analysis using Deep Learning", under the supervision and guidance of Prof. (Dr.) Kamal Sarkar, Jadavpur University, Kolkata. We are satisfied with his work, which is being presented for the partial fulfillment of the degree of Master of Engineering in Computer Science & Engineering, Jadavpur University, Kolkata - 700032.

Prof. Kamal Sarkar
Teacher in Charge of Thesis
Professor, Dept. of Computer Science & Engineering
Jadavpur University, Kolkata – 700 032

Prof. Anupam Sinha
HOD, Dept. of Computer Science & Engineering
Jadavpur University, Kolkata – 700 032

Prof. Chandan Mazumdar
Dean, Faculty Council of Engineering and Technology
Jadavpur University, Kolkata – 700 032

Faculty Council of Engineering and Technology

JADAVPUR UNIVERSITY, KOLKATA – 700032

*Certificate of Approval**

The foregoing thesis entitled "Twitter Sentiment Analysis using Deep Learning," is hereby approved as a creditable study of Master of Engineering in Computer Science & Engineering and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion therein but approve this thesis only for the purpose for which it is submitted.

Final Examination for Evaluation of the Thesis

____

____

Signature of Examiners

* *Only in case the thesis is approved.*

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as part of her Master of Engineering in Computer Science & Engineering.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name:	Sudeshna Nath
Exam Roll Number:	M4CSE22013
Registration number:	154137 of 2020-2021
Thesis Title:	Twitter Sentiment Analysis using Deep Learning

Signature with date:

Acknowledgements

This dissertation would not have been possible without the support of many people. First and foremost, I would like to thank my supervisor, Dr. Kamal Sarkar, Professor, Department of Computer Science and Engineering, Jadavpur University, Kolkata. He has provided me with the perfect balance of guidance and freedom. He is the first person who introduced me to the world of deep learning and text analysis and provided me with the guidance that was essential in making this dissertation. I also want to thank him for his perspective and helping me pursue and define projects with more impact.

I would also like to thank my colleagues in the DST-SERB lab for their support and company without whom this dissertation would not have been possible.

Finally, I would like to thank Jadavpur University for providing me with the opportunity to work in such a productive environment. I would also like to thank all the other professors and research scholars of the department who have extended their helping hands whenever I needed help.

Date:

Place:

Sudeshna Nath,

M.E in CSE,

Exam Roll: M4CSE22013

Contents

Abstract

Sentiment Analysis, also known as opinion mining, is a natural language processing (NLP) technique that helps in analyzing pieces of texts, or an entire text to determine the emotional attitude of the author behind writing the particular message, review, or tweet. It is, basically, a text classification method that helps in identifying whether an online writing carries a positive, negative, or neutral connotation to it. Sentiment analysis finds its significance in a lot of domains these days, the most popular among them being brand and social media monitoring. Most businesses are mindful of their customers' opinions on their products, thereby working on their strengths, and enhancing customer experience, which in turn, benefits them. It even helps in finance and stock monitoring with the correct analysis of customer sentiments for a more beneficial investment into it. Market research and analyzing competitors in it is also a noticeably big advantage that can be achieved via sentiment analysis. In this article, we propose various methods to mine the twitter data, and eventually classify it as positive, negative, or neutral. Amongst the traditional machine learning methods, we have implemented the Support Vector Machine (SVM) algorithm, the Multinomial Naive Bayes, and the Gaussian Naive Bayes methods. For the deep learning methods, we have implemented a Multichannel CNN method, a combination of LSTM and CNN algorithm. We have also used BERT in combination with other deep learning algorithms. Among the lexicon-based approaches, we have used a polarity dictionary in combination with BERT to get superior results on our approach. To implement these models, the datasets used were the US airline twitter dataset which gave the best result at an accuracy of 80.12%, and the SemEval 2013 task A dataset which gave the best result at an accuracy of 65.54%.

Chapter 1:
Introduction

Introduction

Background

Looking into neuroscience literature, emotions are just as important as cognition. In today's world, cognition is given more value than emotions. However, to make a reasonable decision, there needs to be a marker which are the emotions that need to differentiate between "this is a right decision" and "this is not." Emotions were earlier considered to be a psychological construct. Then there came neuroscientists like Antonio Damasio who showed that emotions can be used as real science to be studied. A feeling or an opinion, particularly one that is founded on emotions, is referred to as sentiment. In the subject of text mining, sentiment analysis (SA) is an active area of study. The algorithmic treatment of text's opinions, sentiments, and subjectivity is referred to as SA. It can also refer to a feeling-driven attitude, subjective impression, thinking, or judgement.

The process of recognizing human emotion is known as emotion recognition. Happiness, sorrow, fear, rage, surprise, and other emotions are formed from individuals' personal (subjective) experiences as well as their interactions with their surroundings (audio/visual signals).

People's ability to recognize other people's emotions varies greatly. The use of technology to assist individuals in recognizing emotions is a relatively new study field. In general, technology works best when various modalities are used in conjunction. Automating the recognition of facial expressions from video, spoken expressions from audio, written expressions from text, and physiology, as measured by wearables, has received the most attention to date.

Deep Learning is a subset of Machine Learning that learns to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts. We use multiple hidden layers of neurons between the input and output layers, each of them feeding data into another. These hidden layers, in turn, execute nonlinear transformations of the inputs entered into the network. In a deep learning model, the problem-solving is done end-to-end, whereas, in a traditional machine learning approach, accurate feature identification has to be done by the user. Since our dataset is quite large (the US Airline Twitter dataset consists of 14640 tweets to be processed while the SemEval 2013 Task A dataset consists of 13,220 such tweets), deep learning has proved to work better over traditional machine learning models.

Deep learning is making noteworthy progress in solving problems that have eluded the artificial intelligence field for years. It has proven to be extremely effective at detecting subtle structures in high-dimensional data, making it useful for a wide range of scientific, business, and government applications. It has beaten other machine-learning techniques at predicting the activity of potential drug molecules [34] analyzing particle accelerator data [35, 36] reconstructing brain circuits11, and predicting the effects of mutations in non-coding DNA on gene expression and disease[37,38] in addition to breaking records in image recognition [27, 28, 29, 30] and speech recognition[31, 32, 33]. Deep learning has shown to be exceedingly promising for a variety of tasks in natural language understanding [39], including subject classification, sentiment analysis, question answering [40], and language translation [41, 42].

Applications

Since its inception and rebirth, natural language processing (NLP), which uses computational and linguistics techniques to help computers understand and occasionally produce human languages in the form of texts and speech/voice, has made a significant contribution to offering practical solutions to pressing societal and human issues.Natural Language Processing (NLP), which combines approaches in linguistics and computations to enable computers to comprehend and produce texts and speech/voice in the form of human languages, is used in text-based emotion detection utilizing artificial intelligence. Among the various uses, some of them are listed below:

- Helps in the translation of content from one language to another without any sort of human effort [1].
- Sentiment analysis has advantages that go beyond assisting human agents. If there is a chatbot on a site, sentiment analysis can help it as well. It can teach the chatbot to recognize and respond to customer moods [15].
- To use reviews for accurate suggestions in recommendation systems, Shen et al. [16] presented a novel technique called Sentiment Based Matrix Factorization with Reliability (SBMF+R).
- Sentiment Analysis has found its significance in the healthcare and medical domain as well. Patients can use social media as a good avenue to communicate their wants and concerns. As a result, applying sentiment analysis to analyses patient-generated data on social media can help determine patient healthcare coverage and treatment needs.

Speech recognition systems [2], sentiment analysis [3], text classification [4], questions and answers [5], text summarization [7], are other applications of sentiment analysis in the present-day world. Sentiment analysis is extended to include emotion recognition. Emotion detection is the process of extracting fine-grained emotions such as anger, happiness, sadness, anxiety, depression, and so on, and using this information to inform future decisions. Human-computer interaction [8], education [9], data mining [10], psychology, E-learning [9], software engineering [11], website customization [12], information filtering systems [13], gaming[14], and other fields will benefit from this study of text-based emotion detection using artificial intelligence methods.

Motivation

Eighty percent of the data in the globe is unstructured, according to a survey. Regardless of whether the data is in the form of emails, texts, documents, articles, or anything else, it needs to be examined and organized. Sentiment Analysis is necessary since it stores data in a productive, economical manner. Sentiment analysis helps resolve all real-world problems and real-world circumstances.

Implementation Framework

In this project, we have implemented numerous methods in deep learning to evaluate our models. A multichannel CNN has been used with an activation function that helps to run the model in high-dimensional for the processing of the neurons. A model combining LSTM and CNN has been used in order to combine the concept of dimensionality for the dataset with a huge number of parameters, along with LSTM's capability of being able to capture long-term dependencies between sequences of words. The usage of pre-trained models such as BERT is something that we have done here, as well. The reasons for using BERT are pretty obvious, in the sense that, due to being trained on huge sets of data, it gives good accuracy on most data. Previously, no matter how a word was used, it would return the same vector, whereas BERT returns distinct vectors for the same word based on the words around it. A model that combines BERT with the external knowledge of polarity wordnet has been used. The usage of a polarity wordnet maximizes the linguistic information, thereby adding a boost to our code.

The proposed models were implemented using the Python programming language and using an open-source machine learning library for Python, called TensorFlow.
Python allows wiring clear, concise, and readable code and is extensively used for developing machine learning algorithms. TensorFlow is a free and open-source software library for

machine learning and artificial intelligence. Although it can be applied to many different tasks, deep neural network training and inference are given special attention. One of the best features of TensorFlow is how easy it is to construct the custom operations that are performed during the training and testing of the neural networks. TensorFlow allows programmers to easily declare tensors (n-dimensional vectors) and perform basic operations using them on GPUs.

To assess the performance of any deep learning architecture, it is crucial to have a proper dataset. The performance often depends on how well the data is curated in the first place. Both the models were evaluated on two standard tweet datasets that are used for verifying the performance of deep learning models:

The first dataset is the standard English language dataset often used for sentiment analysis – the US airline twitter dataset This dataset contains a list of 14640 tweets classified as either positive, negative, or neutral. Each row in the dataset has fifteen features attached to it.

The second dataset is the SemEval 2013 Task A dataset. The entire dataset is divided into three parts- training, testing and a validation set. Each row of the dataset contains an ID, the corresponding tweet and a sentiment attached to it.

Organization of Thesis Work

Chapter 1 introduces the thesis. This section provides an overview about deep learning and sentiment analysis, and how they have been used. This section also further discusses how the experiments has been implemented.

Chapter 2 provides a detailed literature survey. The extensive literature survey introduces the concept of sentiment analysis in the machine learning and deep learning domain and discusses in details previously studied models.

Chapter 3 introduces the methodologies that have been proposed. Overall, three different methods have been proposed in the thesis – Machine learning, Deep Learning and Hybrid Approaches.

Chapter 4 discusses the details of the parameters used in order to achieve the best results.

Chapter 5 explains the datasets used in detail and their sources.

Chapter 6 discusses the experiments and results. A total of thirteen experiments are performed using three different approaches and two datasets.

Chapter 7 concludes the thesis and discusses future scope of the project.

Chapter 2:

Literature Survey

Literature Survey

The purpose of the research is to figure out whether a tweet in Indian language is favorable, negative, or neutral. We used three strategies in this case: (1) an SVM-based methodology (2) a few deep learning-based techniques, and (3) a hybrid approach. Although sentiment analysis is an active research field with a large body of work, the majority of existing sentiment analysis research has concentrated on sentiment analysis of products [63] or movie reviews [64] written in the English language. Recently, there has been an increase in study interest in Twitter sentiment analysis in non-English languages.

We found that the first academic studies assessing public opinion were conducted before and after World War II, and they were driven primarily by political considerations [66,67]. Modern sentiment analysis only became popular in the middle of the 2000s, and it initially focused on online product reviews, such as those seen at [65]. Since then, the application of sentiment analysis has expanded to various other fields, including financial market forecasting [68] and responses to terrorist acts [69]. Additionally, several issues that affect the applicability of sentiment analysis, like irony detection [70] and multi-lingual support [71], have been solved in research that combines sentiment analysis and natural language processing. Additionally, regarding emotions, attempts are progressing from straightforward polarity identification to more nuanced distinctions of emotions and differentiating between negative emotions such as rage and grief [72].

Year	Model	Key Features
2017	Topic Adaptive Sentiment Classification using SVM [Lavanya et al. 2017] [74]	- Although applied to static data, the proposed algorithm can be applied to dynamic tweets for a given timeline. - Uses Point-wise Mutual Information and Information Retrieval (PMI-IR) to calculate feature values for each word.
2017	Multichannel LSTM-CNN model for Vietnamese sentiment analysis [Nguyen et al. 2017] [73]	- Built a Vietnamese sentiment (VS) corpus containing 17,500 reviews from Vietnamese e-commercial sites - Proposed a multi-channel LSTM-CNN model for Vietnamese sentiment analysis that outperformed the individual models
2020	BERT based pipeline for Italian twitter sentiment analysis [Chiorrini et al. 2020] [76]	-Uses AlBERTo, a model based on BERT, specifically trained on a large unlabeled Italian tweet corpus

Year	Model	Key Features
2021	SVM, LSTM, CNN models combined with word embeddings, [Moreno-Garcia et al. 2021] [77]	- Uses BERT and Word2vec word embeddings to be combined with the models - Proved deep learning models combined gave better results than using an individual model
2022	Tweet analysis using Composite SVM kernels, BiLSTM and CNN [75]	- Developed Indian language Twitter dataset for topic-oriented sentiment analysis - Developed single and composite kernels and deep learning based-classifiers
2017	Sentiment Analysis using SVM [Ahmad et al. 2017] [78]	-Performs SVM on two pre-classified datasets of tweets. - Uses Weka tool for performance analysis
2017	Twitter sentiment analysis using deep learning methods [Ramadhani et al. 2017] [79]	- Uses feedforward neural network with ReLU and sigmoid function activation - Used the one thousand datasets of each positive and negative for training and testing
2019	Stock market sentiment analysis using BERT [Gomes Sousa et al. 2019] [80]	-Corpus of 582 financial news manually labeled with sentiment from leading news websites - Fine-tuned the pre-trained BERT model with an additional output layer - Data analysis highlighting the relation between the Dow Jones Industrial index and the developed BERT sentiment classifier.
2019	Sentiment analysis using BiLSTM [Xu et al. 2019] [81]	- According to the deficiency of the word representation method in the current research, the sentiment information contribution degree is integrated into the TF-IDF algorithm of the term weight computation, and a new representation method of word vector based on the improved term weight computation is proposed. - Context information is fully understood to implement the BiLSTM model
2020	BERT-BiLSTM to analyze sentiments about investors and consumers in energy market. [Cai et al. 2020] [82]	- The result reveals the advantages of the combination forecasting model, BERT-BiLSTM on sentiment analysis. - This can accurately predict the sentiment orientation of Internet users during the major events so as to provide technical support for the decision-making of energy market.
2014	Ranked wordnet graph for sentiment polarity classification in twitter [Montejo-Ráez et al. 2014] [83]	- Combining the results of a random walk analysis of the concepts found in the text over the WordNet graph with SentiWordNet scores. - Provides a solution to the disadvantages associated with supervised models like SVM
2015	Q-WordNet: Extracting Polarity from WordNet Senses [Agerri et al. 2015] [84]	- The resource builds Q-WordNet as a subset of WordNet synsets with annotations for positive and negative polarity. -Obtains better results on Q-WordNet than WordNet 2.0

Chapter 3:
Proposed Methodologies

Proposed Methodologies

For implementing the proposed sentiment classification method, we have used two types of classifiers-(1) the support vector machine (SVM) classifier (the one vs rest classifier) and (2) deep learning classifiers. In this section, at first, the SVM-based sentiment classification model is described and then the deep learning-based sentiment classification model is described.

1. Traditional Machine Learning based Approach

The machine learning approach allows systems to automatically learn and improve as a result of their experiences. Text is classified into several emotion classes using machine learning algorithms. Social networking sites make their data available on the internet in an easy and unrestricted manner. This abundance of data piques the attention of young researchers who want to pursue a career in sentiment analysis. On social media discussion boards, people express their emotions and perspectives [18]. Machine learning approaches have increased sentiment analysis accuracy and sped up autonomous data evaluation in recent years.

The framework for the SVM-based method consists of preprocessing, feature extraction, and data splitting into train and test sets and classification stages. The set of labelled vectors is obtained after processing the training data, and a classifier is formed using the labelled vectors. The classifier is then evaluated on the test data after it has been generated. We used the one vs. all technique to construct a multiclass SVM model that uses three different binary SVM classifiers, one for each sentiment class because the sentiment classification problem is a multiclass problem with three classes (positive, negative, and neutral). The data is created by labelling all samples in the class y_i as either "positive," "negative," or "neutral" to train the binary SVM classifier for the sentiment class y_i (in our example, i=1 to 3). The preprocessing step of a binary SVM model involves removing punctuation, hashtags, hyperlinks, and other irrelevant characters from the input tweets dataset. Stop word removal is another preprocessing step that is performed. Stop words, such as conjunctions and prepositions, are frequently encountered in texts without being tied to a specific topic. With appropriate text processing, sentiment analysis accuracies using SVM may be improved greatly [19].

Feature extraction is the process of converting raw data into numerical features that may be processed while maintaining the original data set's content. Compared to directly using machine learning on raw data, it yields better results. On the processed data, the experiment applied several term-weighting schemes, consisting of Count Vectorizer and Term Frequency Inverse Document Frequency (TFIDF), for each n-gram scheme to create the word vectors. Countvectorizer is a text-to-numerical data conversion method. An example to demonstrate how it works is given below:

`Text = ["The sun rises in the east," "The blazing sun mocked me"]`

There are two text inputs, and each one is preprocessed, tokenized, and represented as sparse matrices. Countvectorizer employs word-level tokenization and turns the text to lowercase by default.

The text is transformed into a sparse matrix as shown below.

The	sun	rises	in	east	blazing	mocked	me
3	2	1	1	1	1	1	1

Term Frequency — Inverse Document Frequency (TF-IDF) is a statistic that attempts to better identify the importance of a word in a document while also considering its relationship to other documents in the same corpus. This is done by counting the number of times a term appears in a document as well as the number of times the same word appears in other documents in the corpus. There is a quite simple formula that summarizes the TF-IDF measure.

The term frequency (TF) can be calculated as:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

Next, calculating the Inverse Document Frequency (IDF) as:

IDF = log (Number of the documents in the corpus) / (Number of documents where the specific term t appears).

Compiling them, The TF-IDF of a term is calculated by multiplying TF and IDF scores.

TF - IDF = T F * IDF.

The data is then split into two parts: 80% training set and 20% testing set. Python's scikit-learn library has been used to implement these steps.

Support Vector Machine (SVM) has been chosen for the classification in the experiment. The support vector machine algorithm's goal is to find a hyperplane in an N-dimensional space (N — the number of features) that distinguishes between data points. Because of its advantages, such as its ability to handle huge features, SVM performs well for text classification. SVM is also robust in the presence of a sparse set of instances and since the majority of problems are linearly separable [20]. Previous sentiment analysis research [21] [22] [9] has yielded promising results using Support Vector Machines.

2. Deep Learning Based Approach

Deep learning is a machine learning and artificial intelligence (AI) technique, inspired by the structure and functioning of the brain, that mimics how humans acquire knowledge. Deep learning allows computational models with several processing layers to learn multiple degrees of abstraction for data representations. These techniques have vastly enhanced the state-of-the-art in speech recognition, visual object recognition, object detection, and a variety of other fields like drug development and genomics. Deep learning reveals intricate structures in massive data sets by using the backpropagation technique to demonstrate how a machine should modify its internal parameters that are used to compute the representation in each layer from the representation in the previous layer.

Multichannel CNN

A word embedding layer and a one-dimensional convolutional neural network are used in a conventional deep learning model for text categorization and sentiment analysis.
Multiple concurrent convolutional neural networks that read the source content with varied kernel sizes can be used to expand the model. As a result, a multichannel convolutional neural network for text is created, which can read the text of various n-gram sizes (groups of words). The entire process consists of data preparation and developing the model. For data preparation, the sentiment part of the raw data is encoded using a label encoder and converted to categorical values, while the tweets are loaded and cleaned to remove punctuations, numerical, stop words and other irrelevant characters. The data is then split into train and test sets with a ratio of 4:1. The next step of the process involves developing a multichannel convolutional neural network for the sentiment prediction problem. The first step involves loading the cleaned training dataset. The training dataset must then be fitted with a Keras Tokenizer. The tokenizer is utilized to define the Embedding layer's vocabulary as well as encode the review documents as integers. Given a list of documents, a Tokenizer is imported from Keras to create a Tokenizer and then fitted on texts. An Embedding layer is used as the input of a conventional model for document classification, followed by a one-dimensional convolutional neural network, a pooling layer, and finally a prediction output layer. The convolutional layer's kernel size determines the number of words to examine while the convolution is applied to the input text document, acting as a grouping parameter.

Multiple variants of the standard model with varying-sized kernels are used in a multi-channel convolutional neural network for document classification. This enables the document to be processed at various resolutions or n-grams (word groupings) at the same time, while the model learns how to effectively integrate these interpretations. Yoon Kim initially detailed this method in an article published in 2014[25]. Kim worked with both static and dynamic models in his paper (updated). The functional API of Keras can be used to define a multiple-input model. For analyzing 1-gram, 2-grams, and 3-grams of the dataset, we will create a model with three input channels. The following components make up each channel:

- The length of input sequences is defined by the input layer.
- 100-dimensional real-valued representations and an embedding layer set to the vocabulary's size.
- A one-dimensional convolutional layer with 32 filters and a kernel size equal to the number of words to read simultaneously.
- To consolidate the output from the convolutional layer, use the Max Pooling layer.
- To simplify concatenation, flatten the layer to reduce the three-dimensional output to two dimensions.

The three channels' outputs are combined into a single vector, which is then processed by a few Dense layers and an output SoftMax layer.

A combination of LSTM with CNN

Long Short-Term Memory

Long Short-Term Memory (LSTM) is a more advanced variant of recurrent neural network (RNN) architecture that was created to more precisely reflect chronological sequences and their long-range relationships than regular RNNs. The inner design of a basic LSTM cell, the changes included in the LSTM architecture, and a few applications of LSTMs that are in great demand are among the highlights. It also compares and contrasts LSTMs and GRUs. The essay finishes with a list of LSTM network drawbacks and a quick overview of the future attention-based models that are rapidly replacing LSTMs in real-world applications. LSTM networks are an extension of recurrent neural networks (RNNs) mainly introduced to handle situations where RNNs fail. When we talk about RNN, it is a network that operates on the current input while taking into account the prior output (feedback) and temporarily storing it in memory (short-term memory). The most well-liked uses of this technology are in the areas of non-Markovian control, speech processing, and musical composition. RNNs do have some shortcomings, though. In the beginning, it is unable to keep data for a longer period of time. To predict the current output, it may occasionally be necessary to resort to data that was saved a long time ago. However, handling such "long-term dependencies" is completely beyond the capabilities of RNNs. Second, there is no finer control over which part of the context needs to be carried forward and how much of the past needs to be 'forgotten.' Exploding and vanishing gradients, which happen when a network is being trained via backtracking, are another problem with RNNs (more on this later). Long Short-Term Memory (LSTM) was introduced as a result. The training model is left unmodified, and the vanishing gradient problem has been nearly entirely eliminated. With LSTMs, which also deal with noise, distributed representations, and continuous values, certain problems with long lags can be solved. Unlike the hidden Markov model, which requires keeping a limited number of prior states, LSTMs do not have this requirement (HMM). We can choose from a wide range of LSTM parameters, including learning rates and input and output biases. Thus, there is no need for precise modifications. The complexity to update each weight is reduced to O(1) with LSTMs, similar to that of Back Propagation Through Time (BPTT), which is an advantage. Exploding and vanishing gradients, which happen when a network is being trained via backtracking, are another problem with RNNs (more on this later). Long Short-Term Memory (LSTM) was introduced as a result. The training model is left unmodified, and the vanishing gradient problem has been nearly entirely eliminated. With LSTMs, which also deal with noise, distributed representations, and continuous values, certain problems with long lags can be solved. Unlike the hidden Markov model, which requires keeping a limited number of prior states, LSTMs do not have this requirement (HMM). We can choose from a wide range of LSTM parameters, including learning rates and input and output biases. Thus, there is no need for precise modifications. The output is usually in the range of 0-1 where '0' means 'reject all' and '1' means 'include all'.

The output of an LSTM unit at the current state is dependent on the results of past states, making it a type of recurrent neural network (RNN) appropriate for sequence learning. LSTM mixes the current input and the output from the previous time step at each time step. LSTM was created to address the shortcomings of RNN, which can be difficult to train when the input sequence is extensive. RNN is capable of capturing temporal dependencies. The LSTM network attempts to address the exploding or vanishing gradient problem that plagues the vanilla RNN with the use of three gates: the input gate, output gate, and forget gate. Each LSTM cell has three inputs, and two outputs. For a given time t, is the hidden state, is the cell state or memory, is the current data point or input.

The first sigmoid layer has two inputs–and where is the hidden state of the previous cell. It is known as the forget gate as its output selects the amount of information of the previous cell to be included. The output is a number in [0,1] which is multiplied (pointwise) by the previous cell state.

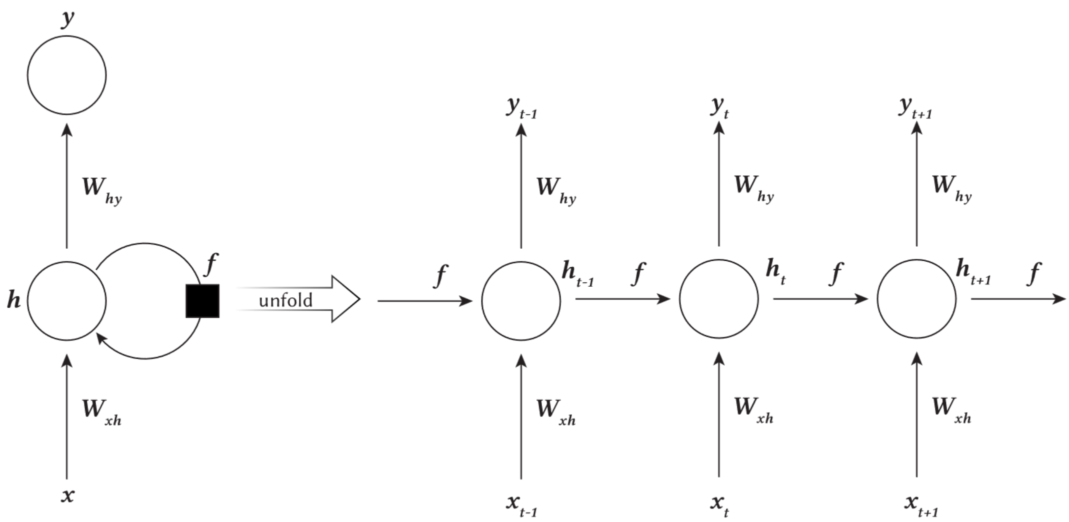


Figure 1. Architecture of a Long short term memory module.

Convolutional Neural Network

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning system that can take an input image, assign relevance (learnable weights and biases) to various aspects/objects in the image, and distinguish between them. A ConvNet requires substantially less pre-processing compared to other classification techniques. ConvNets can learn these filters and properties with enough training, unlike simple techniques that require hand-engineering of filters. A ConvNet's design is modelled after the way the visual cortex is set up and resembles the way neurons are connected in the human brain. Only a small portion of the visual field, known as the Receptive Field, allows individual neurons to respond to stimuli. A group of similar fields converge to cover the visual area completely. The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Three layers make up the Convolution Neural Network model that was implemented. The basic functionality of a convolution neural network is similar to that of an animal's visual cortex. Convolution neural networks perform well in text categorization tasks. Text classification criteria are identical to image classification criteria, with the exception that instead of pixel values, we use a matrix of word vectors. The proposed model is written in Python and uses the TensorFlow library. The convolutional neural network is a multi-layer neural network that improves on the error back propagation network. It excels at dealing with image-related machine learning challenges, particularly huge photos. Yann Lecun was the first to suggest CNN, which he used to recognize handwritten characters [26]. The input layer, convolutional network layer, LSTM, or its variants layer, and softmax classifier layer make up the text categorization model based on CNN and LSTM or its variants. Figure 2 depicts the model's structure. After reading into the data, the text is preprocessed first, following the previous methods of preprocessing to remove the whitespaces, special characters, and stop words. The text is then tokenized into words to be passed into the CNN model, while character-wise tokenization is done for it to be passed into the LSTM layer. The model is then built having a non-sequential LSTM layer, along with a CNN model with two channels to analyze 1-gram and 2-gram of the data. The outputs are then concatenated and passed through a dense layer and a SoftMax layer.

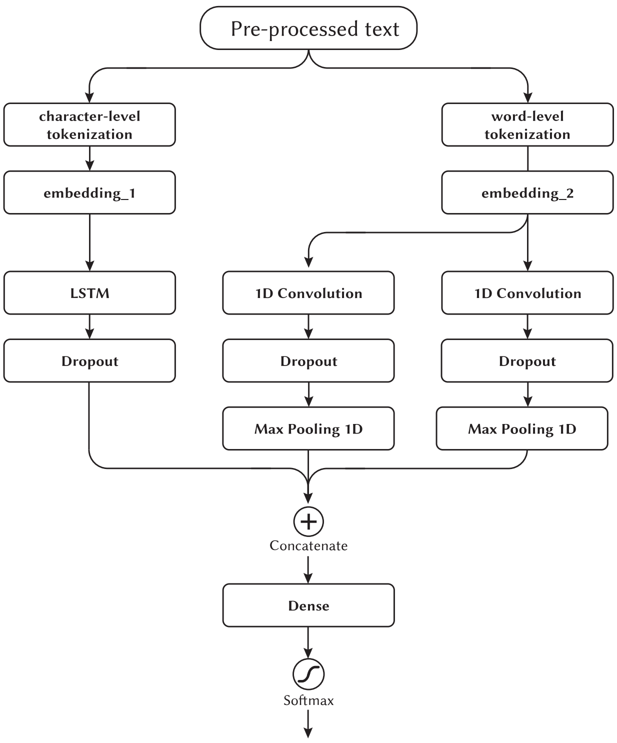


Figure 2. Architecture of a LSTM+CNN.

BERT

In recent years, the BERT model has become a popular state-of-the-art model. It can handle NLP tasks such as supervised text classification without the need for human intervention. This approach is widely popular in academics and industry because of its versatility in dealing with any corpus while producing excellent results. Other tactics, though, have proved successful in the past. We begin with a discussion of BERT and a review of traditional NLP techniques. The behavior of BERT is then empirically tested in combination with other deep learning algorithms and also a lexicon-based dictionary. The goal of this study is to contribute empirical evidence to the case for using BERT as a default on NLP tasks. Experiments demonstrate BERT's superiority and independence from NLP problem aspects such as text language, providing empirical proof for using BERT as a default strategy in NLP situations.

For our model, we use BERT's sentence embedding techniques instead of the word embeddings. Sentence embedding approaches use vectors to encode whole sentences and their semantic content. This aids the machine's comprehension of the text's context, intent, and other nuances. Sentence Embedding, like Word Embedding, is a popular study subject with some extremely fascinating ways of assisting machines in understanding our language. Suppose we come across a sentence like 'Don't run down the street, that's dangerous!', and a few sentences later, we read 'I intend to run for President four years from now.'. Sentence Embedding helps the machine draw the inference between a 'run' down the street and a 'run' for the presidency.

Regarding the BERT model, there are two steps in its framework: pre-training and fine-tuning [43]. We skip the pre-training process and use a pre-trained model already. Because the Transformer's self-attention mechanism allows BERT to mimic multiple downstream jobs, fine-tuning is simple. We just feed the precise inputs and outputs into BERT for each operation and fine-tune all the settings.[43].

There are four fundamental elements at the heart of this BERT-based model:

- Attention
- Transformers
- BERT
- Siamese Network

Sentence-BERT takes two sentences as input and uses a Siamese network-like architecture. The embeddings for these two sentences are then generated using BERT models and a pooling layer. The embedding is then passed into our model and then passed through a few dense layers to obtain the desired results. We use the pre-trained “bert-base-nli-mean-tokens” model.

BERT in combination with LSTM and CNNs

Transformers, like Recurrent Neural Networks (RNN), are made to deal with sequential input. It can process natural language in the same manner that humans can, allowing it to perform jobs such as translation and text classification. Transformers, unlike RNNs, do not require sequential data, in this example text, to be processed in a specific order.

This means that when a text is received as input, it is not essential to process the beginning of the text before the end, allowing for far more parallelization and, as a result, significantly shorter training times. Transformers were created utilizing the attention mechanism, which was created to help machines remember long texts in machine translation tasks. It is built on an encoder-decoder architecture, in which the encoders are made up of a series of encoding layers that process the input layer by layer iteratively. The decoders, on the other hand, are made up of a series of decoding layers that accomplish the same thing at the encoder's output. As a result, when a Transformer receives a text, it is encoded by a stack of encoders. The output from the last encoder is sent to each of the decoders in the stack of decoders, resulting in the final output. Each encoder consists of two main components, an attention mechanism called self-attention and a feed-forward neural network.

For our model, we combine the previous two approaches for BERT and the combined model of LSTM and CNN. The sentiment part of the dataset is encoded categorically, while the tweet part of the dataset is cleaned and processed to obtain a cleaned dataset. The dataset is then passed word-wise to the CNN model, while character-wise to the LSTM model. We then build a model using various combinations of n-gram CNN channels or a combination of non-sequential LSTM channel and an n-gram CNN channel, along with BERT. For the testing part, we perform the same steps of data processing, load the model, and perform an evaluation on it using the testing dataset.

BERT in combination with BiLSTM

The phrase "bidirectional LSTM" refers to a sequence processing model that consists of two LSTMs, one of which accepts input in one direction and the other in the opposite. BiLSTMs effectively boost the network's data volume, providing the algorithm with better context (e.g., knowing what words immediately follow and precede a word in a sentence).

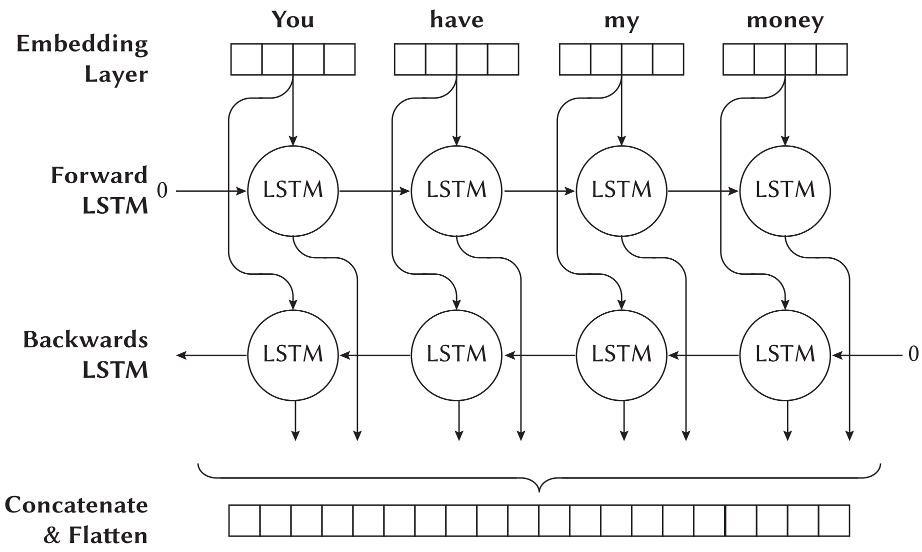


Figure 3. Architecture of a BiLSTM module.

Using bidirectional will run inputs in two directions, one from past to future and the other from future to past. This method differs from unidirectional in that information from the future is preserved in the LSTM that runs backwards, however by combining the two hidden states, you can preserve information from both the past and the future at any point in time. What they are best suited for is a really difficult question, but BiLSTMs perform quite well since they better comprehend the context, as I will attempt to demonstrate with an example. If we were to attempt to anticipate the next word in a sentence, a unidirectional LSTM would likely observe the following:

“You have my money…”

With bidirectional LSTM, we are able to view information further down the road, for example, and will try to forecast the following word only based on this context. Forward LSTM:

“You have my money…”

Backward LSTM:

“… so, find a way to get me there.”

The phrase "bidirectional LSTM" refers to a sequence processing model that consists of two LSTMs, one of which accepts input in one direction and the other in the opposite. BiLSTMs effectively boost the network's data volume, providing the algorithm with better context (e.g., knowing what words immediately follow and precede a word in a sentence). For our model, we encode the sentiment part of the dataset and process the dataset by the methods used for the previous algorithms. A character-wise tokenized dataset is passed for the BiLSTM. These two inputs are simultaneously passed through the model, the word embeddings are extracted for each of them. For the BiLSTM model, the vectors then become input for the BiLSTM units. The outputs of the two LSTM networks are concatenated into a final state that is coupled to a dense layer in a BiLSTM network. The output of the dense layer is then transferred to the Softmax layer, which generates sentiment classification output.

The exact similar model is used for a word-wise tokenized dataset passed on to the BiLSTM network.

3. Hybrid Approach

We enable a hybrid approach to analyze our dataset, combining the pre-trained BERT model along with lexicon-based approach of using the knowledge of a polarity wordnet in order to correctly classify the texts as one of positive, negative or neutral sentiment.

BERT

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model that considers a word's context from both the left and right sides at the same time [44]. BERT's left and right pre-training is accomplished using masked language model masks, which are modified language model masks (MLM). MLM's goal is to obscure a random word in a sentence with a low probability. When a word is masked, the model replaces it with the token [MASK]. With the help of transformers, the model then tries to predict the masked word by leveraging context from both the left and right sides of the masked word. BERT has an additional essential purpose that differs from earlier efforts, namely prediction of the next sentence, in addition to left and right context extraction utilizing MLM.

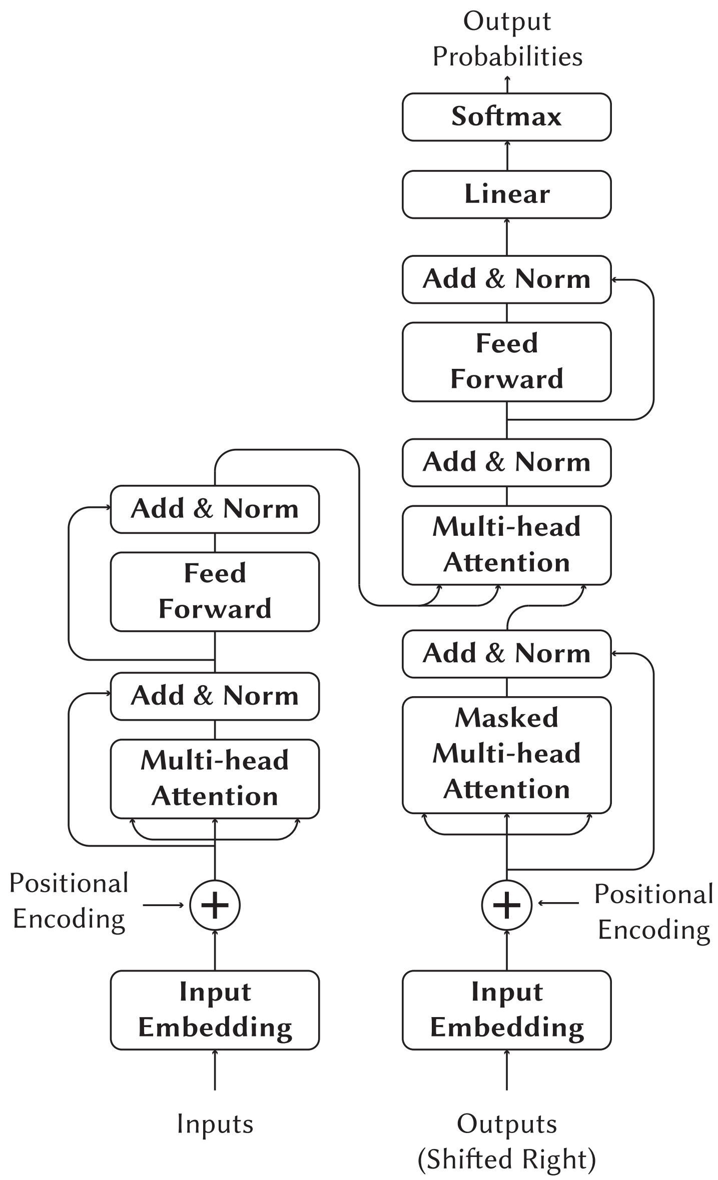


Figure 4. Architecture of a Transformer module.

The text input for the BERT model is first tokenized using a technique known as word piece tokenization. This generates a set of tokens, each of which represents a single word. There are additionally two specialized tokens added to the set of tokens: a classifier token [CLS] at the start of the set, and a separation token [SEP] at the end of a sentence. Transformers' architecture is built on attention mechanics [45], which determines which sequences are relevant in each computational phase. The encoder not only maps the input to a higher-dimensional space vector but also feeds the decoder with the relevant keywords. The model accepts two texts as input and returns a label indicating the sort of relationship between them. This assignment assesses a model's thorough comprehension of natural language as well as its ability to make inferences on whole phrases. Model of Masked Language BERT pre-trains deep bidirectional representations for the language model using a mask token [MASK]. BERT masks a random word in the sequence, as opposed to conditional language models that train left-to-right or right-to-left to predict words. To understand the relationship between two text sentences, Next Sentence Prediction is employed. BERT has been pre-trained to predict whether or not there exists a relation between two sentences. Each of these sentences, sentence A and sentence B, has its own embedding dimensions.

```
Sentence A: [CLS] The man went to the store. [SEP]
Sentence B: He bought a gallon of milk. [SEP]
Label: IsNextSentence
```

Lexicon-based Approach

Lexicon-based techniques assume that the semantic orientation of a text is inextricably linked to the polarity of the words and phrases that appear in it. This has to do with content words, such as adjectives [46], adverbs [47], nouns [48], and verbs, as well as phrases and sentences including them. Although hand-constructed lexicons are clearly more accurate than machine-constructed ones, especially in cross-domain sentiment analysis applications, manual annotation is a time and human resource-intensive operation [49, 50]. This has resulted in a boom of studies on the generation and dissemination of automatic polarity lexicons, which utilize morphological approaches [51, 52], the semantic relations of thesauri [53, 54, 55, 56] and co-occurrence algorithms in huge corpora [57, 58, 59]. Automatically generated dictionaries appear to be more unstable, yet they are typically larger than those constructed manually. In any case, size does not always imply excellence. These big dictionaries are notorious for having sparsely specified content. Furthermore, a substantial number of entries could indicate fewer specifics in the description, or it could indicate more noise.

Model

We have used a list of positive opinion words (or sentiment words) [61] to match with the n-grams of the positive labelled tweets, and another list containing negative opinion words (or sentiment words) [62]. The data processing steps remain the same, as they were for all of the previous models, it is then divided by a 1:1 ratio. One-half of the dataset is passed through BERT model to find out the sentence embeddings. The rest is tokenized, and a matching algorithm is used to find out the number of words that match with the polarity wordnet. A vector is created that contains the sum of positive tweet words found in the wordnet list and a sum of the negative tweet words in the corresponding wordnet list. Both the BERT embeddings and the lexicon vectors are passed through a dense layer, concatenated, and then passed through a SoftMax layer to finally obtain the output layer.

Chapter 4:

Parameter Tuning

Parameter Tuning

We have tested 13 methodologies and the hyperparameters for the methods were found out to be as follows.

Support Vector Machine (SVM)

- C: 1
- Gamma: 0.01
- Kernel: Linear

Multichannel CNN

- Filters: 32
- Kernel size: 1, 2 and 3
- Dropout: 0.5
- Pool size: 2

LSTM + CNN1 + CNN2

- Filters: 16, 32
- Kernel size: 1, 2
- Dropout: 0.5
- LSTM: 100
- Pool size: 2

BERT

- Model Name: bert-base-nli-mean-tokens
- Embedding: Sentence Embeddings
- Epochs= 50
- Optimizer: Adam
- Loss: Categorical Cross Entropy
- Batch size: 50

We have used the same hyperparameters for all models of BERT

BiLSTM

- Dropout: 0.5
- Embedding size: 128
- Bidirectional output layer size: 200
- Epochs: 50
- Batch size: 20

Polarity Wordnet

- Epochs: 50
- Learning Rate: 0.1
- Batch size: 20
- Dense layer activation function: ReLU
- Output layer activation function: Softmax

Chapter 5:

Description of Datasets

Description of Datasets

For the purpose of benchmarking the models, three different datasets were used. Both of the datasets were obtained from Kaggle.

The US Airline Twitter Dataset

The US airline twitter dataset is the base dataset on which the models were developed. This is a sentiment analysis dataset on each major US airline's difficulty. Contributors were requested to classify good, negative, and neutral tweets before categorizing unfavorable causes using Twitter data from February 2015. (Such as "late flight" or "rude service"). The objective of this dataset is to look at how passengers expressed their emotions on Twitter in February 2015. It would be fascinating if airlines could use this free data to improve their customer service. The dataset was acquired from *www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment*.

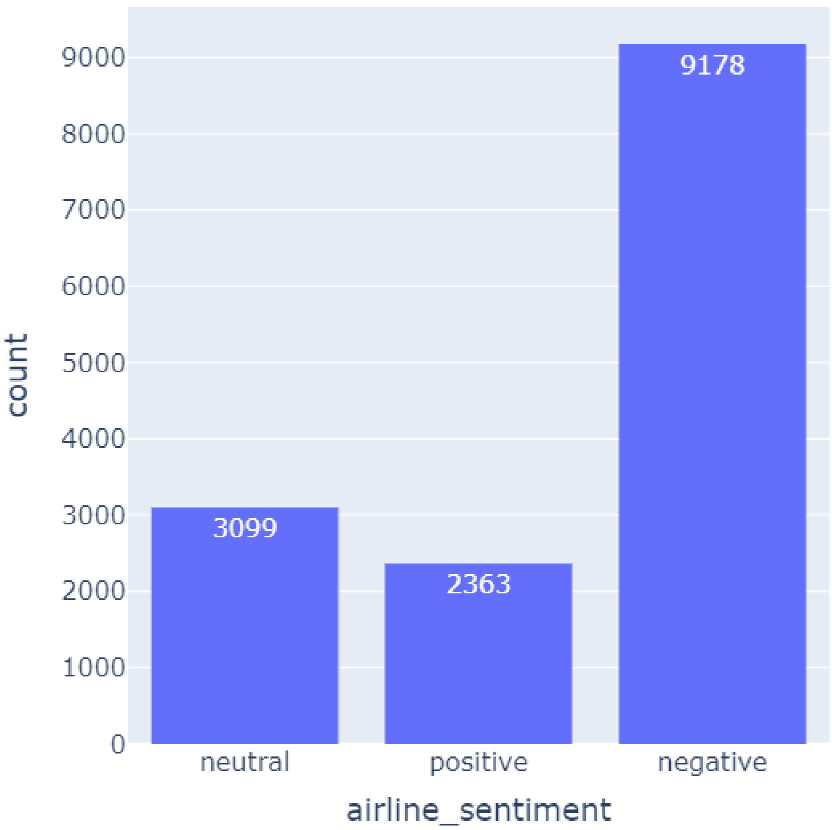


Figure 5. Distribution of the sentiment of the tweets.

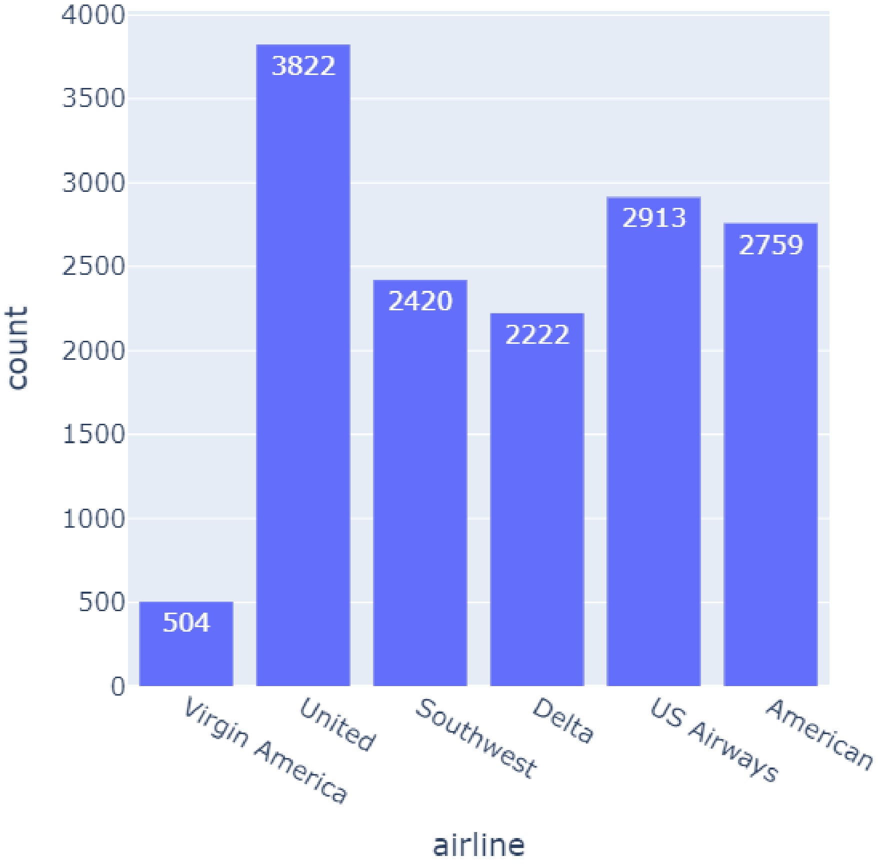


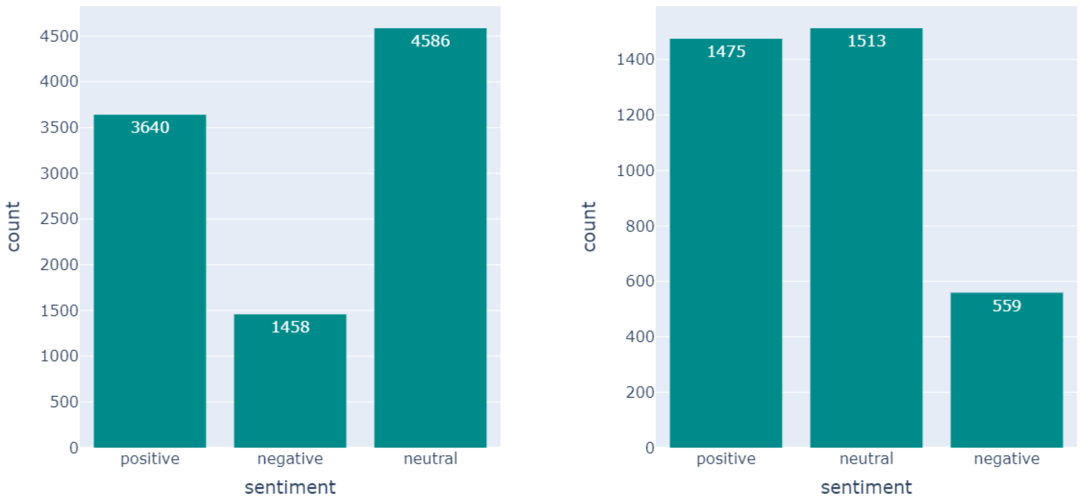
Figure 6. Distribution of number of tweets per airline.

SemEval 2013 Task A Dataset

The SemEval-2013 Task A dataset [86] contains data for three subtasks: A, a training subtask, and B, a testing subtask, and C, a development subtask. Crowdsourcing was used to label a large Twitter training dataset along with additional test sets of Twitter and SMS messages for all three subtasks.

SemEval (Semantic Evaluation) emerged from the Senseval word sense evaluation series and is a continuing series of evaluations of computational semantic analysis systems. The assessments aim to investigate the nature of meaning in language. Although people have an intuitive understanding of meaning, applying such understanding to computational analysis has proven difficult.

In 2013, SemEval-2013 took place in Georgia, USA, in conjunction with NAACL 2013, the North American Association of Computational Linguistics. It consisted of thirteen various evaluation tasks for computational semantic systems.



(a) (b)

Figure 7. Distribution of number of tweets of the SemEval dataset across the (a) training set and (b) testing set.

Chapter 6:

Experiments and Results

Experiments and Results

The following section discusses the performance of the different models on the dataset. We have tested a total of 13 models on the two different datasets in order to validate our models and tuned the hyperparameters accordingly to achieve best results.

For each model, precision, recall, macro-F1 score, and accuracy was calculated [87]. This was done using the following formula:

Precision (P) is defined as the number of true positives (tp) over the number of true positives plus the number of false positives (fp).

$$P = \frac{tp}{tp + fp}$$

Recall (R) is defined as the number of true positives (tp) over the number of true positives plus the number of false negatives (fn).

$$\mathrm{R} = \frac{tp}{tp + fn}$$

These quantities are also related to the (F1) score, which is defined as the harmonic mean of precision and recall, given as:

$$F1 = 2\frac{P \times R}{P + R}$$

Along with these metrics, the accuracy of the model is also given. This is given as:

$$Acc = \frac{tp + tn}{tp + tn + fp + fn}$$

US Airline Twitter Dataset

The performance of the various methods implemented on the US twitter dataset has been given below. We have obtained the best results by combining sentence embeddings through BERT along with using a lexicon-based approach.

Experiment	Accuracy (%)	Precision	Recall	F1 score (macro)
Support Vector Machine	78.29	0.7813	0.7731	0.7830
Multichannel CNN	77.02	0.7688	0.7789	0.7754
LSTM + CNN 1 (1-gram) + CNN 2 (2-gram)	78.38	0.7820	0.7898	0.7844
BERT	79.16	0.7913	0.7754	0.7989
BERT + LSTM	77.90	0.7769	0.7696	0.7757
BERT + CNN 1 (1-gram)	78.79	0.7890	0.7895	0.7891
BERT + CNN 2 (2-gram)	77.18	0.7801	0.7701	0.7745
BERT + LSTM + CNN 1 (1-gram)	79.00	0.7945	0.7896	0.7911
BERT + LSTM + CNN 2 (2-gram)	73.87	0.7389	0.7321	0.7345
BERT + CNN 1 (1-gram) + CNN 2 (2-gram)	73.46	0.7290	0.7365	0.7354
BERT + BiLSTM (character-wise)	78.24	0.7856	0.7810	0.7894
BERT + BiLSTM (word-wise)	77.83	0.7980	0.7534	0.7771
BERT + Polarity wordnet	80.12	0.7933	0.8060	0.8034

Table 2: Comparative performance of the different models on the US Airline Twitter Dataset

SemEval 2013 Task A Dataset

The performance results of the various models on the SemEval 2013 Task A Dataset are given in the table below. Like the other dataset, we have obtained the best results on our model combining BERT sentence embeddings along with lexicon-based approach.

Experiment	Accuracy (%)	Precision	Recall	F1 score (macro)
Support Vector Machine	62.34	0.6145	0.6271	0.6211
Multichannel CNN	58.66	0.5841	0.5864	0.5867
LSTM + CNN 1 (1-gram) + CNN 2 (2-gram)	56.20	0.5601	0.5718	0.5619
BERT	65.14	0.6614	0.6578	0.6542
BERT + LSTM	59.70	0.5912	0.5814	0.5998
BERT + CNN 1 (1-gram)	58.56	0.5856	0.5879	0.5814
BERT + CNN 2 (2-gram)	59.17	0.5719	0.5927	0.5994
BERT + LSTM + CNN 1 (1-gram)	59.12	0.5911	0.5764	0.5641
BERT + LSTM + CNN 2 (2-gram)	57.23	0.5638	0.5798	0.5834
BERT + CNN 1 (1-gram) + CNN 2 (2-gram)	58.36	0.5716	0.5856	0.5836
BERT + BiLSTM (character-wise)	62.13	0.6376	0.6217	0.6159
BERT + BiLSTM (word-wise)	63.22	0.6325	0.6374	0.6398
BERT + Polarity wordnet	65.54	0.6552	0.6574	0.6513

Table 3: Comparative performance of the different models on the US Airline Twitter Dataset

Chapter 7:

Conclusion

Conclusion

This paper studied a hybrid model combining BERT with polarity wordnet, along with a traditional machine learning model and various other deep learning models and for improving topic sentiment classification performance on Twitter data in Indian languages. To evaluate our work, we have used the freely available US airline twitter dataset. To prove the robustness and generalization capability, our developed models have also been evaluated on a publicly available SemEval 2013 Task A dataset.

The main focus has been drawn to a multichannel LSTM-CNN model that we implement for the thesis and use in various combinations to get better results. To improve the performance, we have used sentence embeddings using a pre-trained BERT model along with the deep learning models. BERT has also been used in combination with BiLSTM models, both character-wise and word-wise to tune the accuracy of our sentiment analysis. The final model that we employ for the thesis is the hybrid model that uses word embeddings from BERT and creates a vector of its own by calculating the polarity of each tweet word in the positive and negative list of words (obtained from GitHub) and passes through a deep neural network to obtain the results.

To validate the results obtained by the proposed hybrid deep learning model, statistical significance tests have been conducted. The significance tests reveal that the hybrid deep learning models achieve better results than individual models.

In this study, we have used a few deep learning models with word embeddings from BERT. Further studies can include working and improving on the polarity wordnet, with better wordnet from which to obtain results. A few deep learning models can be used in combination to tune the accuracy of the dataset. Furthermore, the hybrid models can be studied on hybrid datasets, or datasets can be created manually to be applied to the models. The presence of phony reviews or comments is a significant aspect that influences sentiment analysis [85]. A preprocessing module capable of detecting false reviews or comments can be integrated with the sentiment analysis system for upgrading the sentiment analysis-based policy-`making or product recommendation system.

References

[1] Middi Venkata Sai Rishita, Middi Appala Raju, and Tanvir Ahmed Harris, “Machine translation using natural language processing” in MATEC Web of Conferences 277:02004

[2] Dr. Kavitha. R, Nachammai. N, Ranjani. R, and Shifali. J, “Speech Based Voice Recognition System for Natural Language Processing” in IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4), 2014, 5301-5305

[3] Wei Yen Chong, Bhawani Selvaretnam, and Lay-Ki Soon, “Natural Language Processing for Sentiment Analysis- an Exploratory Analysis on tweets” in 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology

[4] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, “Text Classification Using Machine Learning Techniques” in WSEAS TRANSACTIONS on COMPUTERS, Issue 8, Volume 4, August 2005, pp. 966-974

[5] Shivani Singh, Nishtha Das, Rachel Michael, and Dr. Poonam Tanwar, “The Question Answering System Using NLP and AI” in International Journal of Scientific & Engineering Research Volume 7, Issue 12, December-2016 ISSN 2229-5518

[6] Soufyane Ayanouz, Boudhir Anouar Abdelhakim, and Mohammed Benhmed, “A Smart Chatbot Architecture based NLP and Machine learning for health care assistance.” In The Fifth International Conference on Smart City ApplicationsArtificial inteligence based chatbots

[7] Ishitva Awasthi, Kuntal Gupta; Prabjot Singh Bhogal; Sahejpreet Singh Anand; Piyush Kumar Soni, “Natural Language Processing (NLP) based Text Summarization - A Survey”, published in 2021 6th International Conference on Inventive Computation Technologies (ICICT)

[8] Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, and George Votsis, “Emotion Recognition in Human Computer Interaction” in IEEE Signal Processing Magazine 18(1):32 - 80

[9] Mar Saneiro, Olga C. Santos, Sergio Salmeron-Majadas, and Jesus G. Boticario, “Towards Emotion Detection in Educational Scenarios from Facial Expressions and Body Movements through Multimodal Approaches” published in The Scientific World Journal 2014(4)

[10] Kusal, S.; Patil, S.; Kotecha, K.; Aluvalu, R.; Varadarajan, V. AI Based Emotion Detection for Textual Big Data: Techniques and Contribution. Big Data Cogn. Comput. 2021, 5, 43. https://doi.org/10.3390/bdcc5030043

[11] Agata Kolakowska, Agniezkal Landowska Mariusz Szwoch, Wioleta Szwoch, and Michal R. Wrobel, “Emotion Recognition and its application in Software Engineering” in 6th International Conference on Human System Interaction, June 06-08, 2013

[12] O.B. Efremides, “From Emotion Recognition to Website Customizations” in (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 7, 2016

[13] Francisca Adoma Acheampong, Chen Wenyu, and Henry Nunoo-Mensah, “Text-based emotion detection: Advances, challenges, and opportunities” in 2020 The Authors. Engineering Reports published by John Wiley & Sons, Ltd.
[14] Prerna Mishra, “HMM Based Emotion Detection in Games” published in 3rd International Conference for Convergence in Technology (I2CT) 2018
[15] Rajani S Kamath, Shruti Jamsandekar, Mr. M.B. Patil, “Chatbot Response Mining using Sentiment Analysis” in AICTE Sponsored “National Conference on Emerging Trends, Challenges and Opportunities in Data Mining and Information Security” NTCOMIS2020, January 2020
[16] R.-P. Shen, H.-R. Zhang, H. Yu, F. Min, Sentiment based matrix factoriza- tion with reliability for recommendation, Expert Syst. Appl. 135 (2019) 249–258,
[17] BBC News, “Online hate speech rose 20% during pandemic: 'We've normalised it', 15 November 2021, url: https://www.bbc.com/news/newsbeat-59292509
[18] Kiritchenko S, Mohammad S, Salameh M (2016) SemEval-2016 task 7: determining sentiment intensity of English and Arabic phrases. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016). doi:10.18653/v1/s16-1004
[19]. Emma Haddi, Xiaohui Liu, Yong Shi, The Role of Text Pre-processing in Sentiment Analysis, Procedia Computer Science, Volume 17,2013,
[20]. T. Joachims, "Text categorization with support vector machines: Learning with many relevant features" in Machine Learning: ECML-98 ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 1398, pp. 137-142, 1998.
[21].**9.** A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, "Sentiment analysis of twitter data", *Proceedings of the Workshop on Languages in Social Media ser. LSM '11*, pp. 30-38, 2011.
[22] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques", *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10 ser. EMNLP '02*, pp. 79-86, 2002.
[23]. E. Martinez, V. Martin, M. Teresa, J. M. Perea Ortega and L. A. Urena Lopez, "Tecnicas de clasificacion de opiniones aplicadas a un corpus en espanol", *Procesamiento de Lenguaje Natural*, vol. 47, pp. 163-170, 2011.
[24]. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* 521, 436–444 (2015). https://doi.org/10.1038/nature14539
[25]. https://arxiv.org/abs/1408.5882
[26] Y. L. Lecun et al., "Gradient-Based Learning Applied to Document Recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.
[27]. Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems 25* 1090–1098 (2012).
[28]. Farabet, C., Couprie, C., Najman, L. & LeCun, Y. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1915–1929 (2013).

[29] Tompson, J., Jain, A., LeCun, Y. & Bregler, C. Joint training of a convolutional network and a graphical model for human pose estimation. In *Proc. Advances in Neural Information Processing Systems 27* 1799–1807 (2014).
[30] Szegedy, C. et al. Going deeper with convolutions. Preprint at http://arxiv.org/abs/1409.4842 (2014).
[31] Mikolov, T., Deoras, A., Povey, D., Burget, L. & Cernocky, J. Strategies for training large scale neural network language models. In *Proc. Automatic Speech Recognition and Understanding* 196–201 (2011).
[32] Hinton, G. et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine* 29, 82–97 (2012).
[33] Sainath, T., Mohamed, A.-R., Kingsbury, B. & Ramabhadran, B. Deep convolutional neural networks for LVCSR. In *Proc. Acoustics, Speech, and Signal Processing* 8614–8618 (2013).
[34] Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E. & Svetnik, V. Deep neural nets as a method for quantitative structure-activity relationships. *J. Chem. Inf. Model.* 55, 263–274 (2015).
[35] Ciodaro, T., Deva, D., de Seixas, J. & Damazio, D. Online particle detection with neural networks based on topological calorimetry information. *J. Phys. Conf. Series* 368, 012030 (2012).
[35] Kaggle. Higgs boson machine learning challenge. *Kaggle* https://www.kaggle.com/c/higgs-boson (2014).
[36] Helmstaedter, M. et al. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature* 500, 168–174 (2013).
[37] Leung, M. K., Xiong, H. Y., Lee, L. J. & Frey, B. J. Deep learning of the tissue-regulated splicing code. *Bioinformatics* 30, i121–i129 (2014).
[38] Xiong, H. Y. et al. The human splicing code reveals new insights into the genetic determinants of disease. *Science* 347, 6218 (2015).
[39] Collobert, R., et al. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* 12, 2493–2537 (2011).
[40] Bordes, A., Chopra, S. & Weston, J. Question answering with subgraph embeddings. In *Proc. Empirical Methods in Natural Language Processing* http://arxiv.org/abs/1406.3676v3 (2014).
[41] Jean, S., Cho, K., Memisevic, R. & Bengio, Y. On using very large target vocabulary for neural machine translation. In *Proc. ACL-IJCNLP* http://arxiv.org/abs/1412.2007 (2015).
[42.] Sutskever, I. Vinyals, O. & Le. Q. V. Sequence to sequence learning with neural networks. In *Proc. Advances in Neural Information Processing Systems 27* 3104–3112 (2014).
[43] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pretraining of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
[44] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019, "BERT: Pre-training of deep bidirectional transformers for language under- standing" in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:

Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Associ- ation for Computational Linguistics

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need",2017

[46] Hatzivassiloglou, V.; McKeown, K.R. Predicting the Semantic Orientation of Adjectives. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain, 7–12 July 1997; Cohen, P.R., Wahlster, W., Eds.; Morgan Kaufmann Publishers/ACL: Burlington, MA, USA, 1997; pp. 174–181.

[47] Benamara, F.; Cesarano, C.; Picariello, A.; Recupero, D.R.; Subrahmanian, V.S. Sentiment Analysis: Adjectives and Adverbs are Better than Adjectives Alone. In Proceedings of the First International Conference on Weblogs and social media, ICWSM 2007, Boulder, CO, USA, 26–28 March 2007.

[48] Vermeij, M. The Orientation of User Opinions through Adverbs, Verbs and Nouns. Available online: **https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.4909&rep1&pdf** (accessed on 1 October 2021).

[49] Taboada, M.; Brooke, J.; Tofiloski, M.; Voll, K.D.; Stede, M. Lexicon-Based Methods for Sentiment Analysis. *Comput. Linguist.* **2011**, *37*, 267–307.

[50] Bloom, K. Sentiment Analysis Based on Appraisal Theory and Functional Local Grammars. Ph.D. Thesis, Illinois Institute of Technology, Chicago, IL, USA, 2011.

[51] Moilanen, K.; Pulman, S.G. The Good, the Bad, and the Unknown: Morphosyllabic Sentiment Tagging of Unseen Words. In Proceedings of the ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, Columbus, OH, USA, 15–20 June 2008; The Association for Computer Linguistics: Stroudsburg, PA, USA, 2008; pp. 109–112.

[52] Neviarouskaya, A. Compositional Approach for Automatic Recognition of Fine-Grained Affect, Judgment, and Appreciation in Text (Soft Computing, < Special Issue> Doctorial Theses on Aritificial Intelligence). *J. Jpn. Soc. Artif. Intell.* **2012**, *27*, 88.

[53] Esuli, A.; Sebastiani, F. Determining the semantic orientation of terms through gloss classification. In Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, 31 October–5 November 2005; Herzog, O., Schek, H., Fuhr, N., Chowdhury, A., Teiken, W., Eds.; ACM: New York, NY, USA, 2005; pp. 617–624.

[54] Esuli, A.; Sebastiani, F. Determining Term Subjectivity and Term Orientation for Opinion Mining. In Proceedings of the EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, 3–7 April 2006; McCarthy, D., Wintner, S., Eds.; The Association for Computer Linguistics: Stroudsburg, PA, USA, 2006.

[55] Paulo-Santos, A.; Ramos, C.; Marques, N.C. Determining the Polarity of Words through a Common Online Dictionary. In *Progress in Artificial Intelligence, Proceedings of the 15th Portuguese Conference on Artificial Intelligence, EPIA 2011, Lisbon, Portugal, 10–13 October 2011*; Antunes, L., Pinto, H.S., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 7026, pp. 649–663.
[56] Paulo-Santos, A.; Ramos, C.; Marques, N.C. Determining the Polarity of Words through a Common Online Dictionary. In *Progress in Artificial Intelligence, Proceedings of the 15th Portuguese Conference on Artificial Intelligence, EPIA 2011, Lisbon, Portugal, 10–13 October 2011*; Antunes, L., Pinto, H.S., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 7026, pp. 649–663.
[57] Awadallah, A.H.; Radev, D.R. Identifying Text Polarity Using Random Walks. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010; Hajic, J., Carberry, S., Clark, S., Eds.; The Association for Computer Linguistics: Stroudsburg, PA, USA, 2010; pp. 395–403.
[58] Qiu, G.; Liu, B.; Bu, J.; Chen, C. Expanding Domain Sentiment Lexicon through Double Propagation. In Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, CA, USA, 11–17 July 2009; pp. 1199–1204.
[59] Kanayama, H.; Nasukawa, T. Fully Automatic Lexicon Expansion for Domain-oriented Sentiment Analysis. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, 22–23 July 2006; pp. 355–363.
[60] Baroni, M.; Vegnaduzzo, S. Identifying subjective adjectives through web-based mutual information. In Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019), Erlangen, Germany, 7 May 2019.
[61] Hu, M., & Liu, B. (2004, August). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 168-177).
[62] Liu, B., Hu, M., & Cheng, J. (2005, May). Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web* (pp. 342-351).
[63] T. K. Shivaprasad and J. Shetty, "Sentiment analysis of product reviews: A review," 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), 2017, pp. 298-301, doi: 10.1109/ICICCT.2017.7975207.
[64] Baid, Palak, Apoorva Gupta, and Neelam Chaplot. "Sentiment analysis of movie reviews using machine learning techniques." *International Journal of Computer Applications* 179.7 (2017): 45-49.
[65] K. Dave, S. Lawrence, and D. M. Pennock, “Mining the peanut gallery: Opinion extraction and semantic classification of product reviews,” in Proceedings of the 12th international conference on World Wide Web, 2003, pp. 519–528.

[66] R. Stagner, “The cross-out technique as a method in public opinion analysis,” The Journal of Social Psychology, vol. 11, no. 1, pp. 79–90, 1940.
[67] A. L. Knutson, “Japanese opinion surveys: the special need and the special difficulties,” Public Opinion Quarterly, vol. 9, no. 3, pp. 313–319, 1945.
[68] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah, and D. C. L. Ngo, “Text mining for market prediction: A systematic review,” Expert Systems with Applications, vol. 41, no. 16, pp. 7653–7670, 2014.
[69] P. Burnap et al., “Tweeting the terror: modelling the social media reaction to the Woolwich terrorist attack,” Social Network Analysis and Mining, vol. 4, no. 1, pp. 1–14, 2014.
[70] A. Reyes and P. Rosso, “On the difficulty of automatically detecting irony: beyond a simple case of negation,” Knowledge and Information Systems, vol. 40, no. 3, pp. 595–614, 2014.
[71] A. Hogenboom, B. Heerschop, F. Frasincar, U. Kaymak, and F. de Jong, “Multi-lingual support for lexicon-based sentiment analysis guided by semantics,” Decision support systems, vol. 62, pp. 43–53, 2014.
[72] E. Cambria, P. Gastaldo, F. Bisio, and R. Zunino, “An ELM-based model for affective analogical reasoning,” Neurocomputing, vol. 149, pp. 443–455, 2015.
[73] Q. -H. Vo, H. -T. Nguyen, B. Le, and M. -L. Nguyen, "Multi-channel LSTM-CNN model for Vietnamese sentiment analysis," 2017 9th International Conference on Knowledge and Systems Engineering (KSE), 2017, pp. 24-29, doi: 10.1109/KSE.2017.8119429.
[74] K. Lavanya and C. Deisy, "Twitter sentiment analysis using multi-class SVM," 2017 International Conference on Intelligent Computing and Control (I2C2), 2017, pp. 1-6, doi: 10.1109/I2C2.2017.8321798.
[75] Shuverthi Maity and Kamal Sarkar. 2022. Topic Sentiment Analysis for Twitter Data in Indian Languages Using Composite Kernel SVM and Deep Learning. ACM Trans. Asian Low-Resour. Lang. Inf. Process. Just Accepted (February 2022). https://doi.org/10.1145/3519297
[76] Chiorrini, Andrea, Claudia Diamantini, Alex Mircoli, and Domenico Potena. "Emotion and sentiment analysis of tweets using BERT." In *EDBT/ICDT Workshops*. 2021.
[77] Dang, C. N., Moreno-García, M. N., & De la Prieta, F. (2021). Hybrid deep learning models for sentiment analysis. *Complexity*, *2021*.
[78] Ahmad, M., Aftab, S., & Ali, I. (2017). Sentiment analysis of tweets using svm. *Int. J. Comput. Appl*, *177*(5), 25-29.
[79] A. M. Ramadhani and H. S. Goo, "Twitter sentiment analysis using deep learning methods," 2017 7th International Annual Engineering Seminar (InAES), 2017, pp. 1-4, doi: 10.1109/INAES.2017.8068556.
[80] M. G. Sousa, K. Sakiyama, L. d. S. Rodrigues, P. H. Moraes, E. R. Fernandes, and E. T. Matsubara, "BERT for Stock Market Sentiment Analysis," 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), 2019, pp. 1597-1601, doi: 10.1109/ICTAI.2019.00231.

[81] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment Analysis of Comment Texts Based on BiLSTM," in *IEEE Access*, vol. 7, pp. 51522-51532, 2019, doi: 10.1109/ACCESS.2019.2909919.
[82] R. Cai et al., "Sentiment Analysis About Investors and Consumers in Energy Market Based on BERT-BiLSTM," in IEEE Access, vol. 8, pp. 171408-171415, 2020, doi: 10.1109/ACCESS.2020.3024750.
[83] Arturo Montejo-Ráez, Eugenio Martínez-Cámara, M. Teresa Martín-Valdivia, L. Alfonso Ureña-López,
Ranked WordNet graph for Sentiment Polarity Classification in Twitter, Computer Speech & Language, Volume 28, Issue 1,2014,
[84] Rodrigo Agerri and Ana García-Serrano. 2010. Q-WordNet: Extracting Polarity from WordNet Senses. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).
[85] Gaurav, A., Gupta, B. B., Chui, K. T., Peraković, D., Chaurasia, P., & Hsu, C. H. (2021, December). A Novel Approach for Fake Comments and Reviews Detection on the Online Social Networks. In *International Conference on Smart Systems and Advanced Computing (Syscom-2021).*
[86] UzZaman, N., Llorens, H., Derczynski, L., Allen, J., Verhagen, M., & Pustejovsky, J. (2013, June). Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (pp. 1-9).
[87] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.