The Sentiment analysis of Movie reviews using the Transfer Learning approach

MSc Research Project
Data Analytics

Tanmay Shrikant Paranjape
Student ID: x20101015

School of Computing
National College of Ireland

Supervisor: Hicham Rifai

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Tanmay Shrikant Paranjape

Student ID: X20101015

Program: Data Analytics **Year:** 2021-22

Module: Research Project

Supervisor: Hicham Rifai

Submission Due Date: 15/08/2022

Project Title: The Sentiment Analysis of Movie reviews using the Transfer Learning approach

Word Count: 10517 **Page Count:** 24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Tanmay Shrikant Paranjape

Date: 15/08/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

The Sentiment Analysis of Movie reviews using the Transfer Learning approach

Tanmay Paranjape
x20101015

Abstract

Movie reviews are often considered an important factor in determining movie success. It has been often observed that the consumers refer the genuine reviews and decide whether the movie is worth the price of admission. The professional reviews posted by experts and reviews posted by consumers affect the decision to purchase a movie ticket. The transfer learning approach in deep learning is a method in which a model developed while solving one task is been used to solve another task. The use of Pre-trained models is a popular approach in Natural language processing and this study aims to use pre-trained models for Sentiment analysis of movie reviews. The dataset on IMDB movie reviews has been taken from Kaggle and BERT (Bi-directional Encoder representation from Transformers) & USE (Universal sentence encoder) has been used for the methodology to answer the research question. The NLTK (Natural Language toolkit) has been adapted for pre-processing because of its prominent features. The accuracy of 84.49% has been achieved by USE whereas the BERT and LSTM achieved 75.70% & 82.75% respectively.
Keywords: LSTM, BERT, USE, Encoder, Transformers

1 Introduction

Sentiment analysis is a type of contextual text mining that identifies and extracts subjective data from the source material. It helps companies analyze social sentiment of their products, services, and brands while monitoring online debates. To extract, recognize, and categorize various opinions given in text format, sentiment analysis uses computational linguistics and natural language processing. It has drawn researchers from a variety of fields, particularly computer science because it is a subfield of interactive computation or human-computer interaction (HCI). The sentiment analysis is a mixture of Natural language processing (NLP) and machine learning (ML). Because of its wide advantages, it has been now used by various businesses and companies by using various research tools[1]. Various advertisement agencies are now continuously watching reviews on social media sites to decide which advertisements that will catch the more users. The sentiment analysis has two prominent approaches, Lexicon based and machine learning based Qaisar, S.M. (2020).

The traditional machine learning algorithms have been used by many researchers lately, however, these techniques were based on analyzing repetitive patterns present in the textual data. It is very important to understand the context of the text in a very deep manner. Recently, transfer learning-based mechanisms have evolved in a very rapid manner and these mechanisms are achieving very high success as stated in various kinds of research[2]. Through transfer learning, the task which has already been performed on one task will be used to carry

[1] https://www.qualtrics.com/uk/experience-management/research/sentiment-analysis/
[2] https://levity.ai/blog/what-is-transfer-learning

out the other task. The knowledge-sharing pattern technique is now state-of-art and has proven to be effective in various sentiment analysis experiments Patel, D. et al. (2020). The traditional machine learning and LSTM algorithms have been used in many sentiment analysis types of research. The transfer learning-based approaches have not been explored prominently in the sentiment analysis domain and this study aims to feel that gap by implementing those approaches. The objective and contribution of this research have been explained in the following subsections.

1.1 Research Question

"To what extent Pre-trained transfer learning-based models can be used to accurately classify the movie reviews?"

The research question presented above aims to predict up to what extent the State-of-art pre-trained transfer learning models can able to classify the IMDB movie reviews. The performance of pre-trained models such as BERT and USE will be compared with the LSTM model. The comparison will help to evaluate the impact of transfer learning techniques.

1.2 Research Objective

The research objective has been chosen to answer the research question stated above. Gathering the overall goal of any research is the first step before moving on to the methodological element. This research aims to use a pre-trained modeling approach and along with that forecast the movie reviews without being biased towards any of the classes (Positive & Negative). The research objective is as follows.

- Comparing the performance of Sentiment analysis using State-of-art transfer learning models.
- To evaluate the effect of Hyperparameters on model performance.
- To add a positive contribution to the sentiment analysis domain.

The literature review in section 2, has been designed in a specific way to justify all the research objectives. The literature review commences with traditional techniques in Sentiment analysis and it concludes with state-of-art techniques.

2 Literature Review

Various machine learning algorithms have been used for many years for Sentiment aa nalysis in variety of domains. Different time-consuming approaches have been replaced by Machine learning approaches because of their cost-efficiency and effectiveness. In terms of Sentiment analysis, the traditional machine learning algorithms were based on repetitive patterns, and thus as technology advances these algorithms were proven to be inefficient. In recent years Neural networks and transfer learning come into place to replace the traditional methods. The following literature review describes the many approaches utilized by researchers, as well as their shortcomings and suggestions for how to address limitations in existing research.

2.1 Traditional Supervised Machine Learning algorithms for Sentiment analysis

The sentiment analysis is used not only by businesses and companies but also by governments to understand the context of data. With the help of machine learning techniques, it becomes very easy to classify the data according to its context. The following paragraphs will illustrate a few of these methods with the help of various research.

Han, K.X. et al. (2020) has done sentiment analysis on the Twitter dataset using a Support Vector Machine (SVM). Instead of relying entirely on Statistics of sentiment words author has considered vocabulary and latent semantic information. To achieve the proposed method author has used the Fisher Kernel (FK) function which is based on Probabilistic Latent Semantic analysis (PLSA). This method allows the use of latent semantic information involving probability characteristics for classification. Two experiments were carried out to obtain the most accurate results. In the first experiment, five rounds of cross-validation have been carried out using FK-SVM, and PLSA-SVM methods & in the second experiment various proportions of training samples have been used with FK-SVM & PLSA-SVM methods. The FK-SVM has been proven to be the most efficient method in both of these experiments, however, the research was restricted to only one modeling technique. Other modeling approaches could have been used.

The sentiment analysis of customer reviews on amazon products has been done by Dey, S et al. (2020). Instead of restricting to only one modeling technique as done by Han, K.X. et al. (2020), the author in this research paper has used two classifiers, Support vector machine (SVM) and Naïve Bayes. The active learner method has been used to label the data since there was a large number of reviews in the dataset. Only the Positive and Negative review categories have been taken and neutral reviews have been discarded. The data pre-processing has been done in 3 steps. In the first step, the Tokenization has been done in which the string has been separated into keywords, with the global constants. For feature extraction, TF-IDF, Frequent noun identifier, and relevant noun removal were used. The SVM has resulted in 84% accuracy as compared to naïve Bayes which has resulted in 82%. For pre-processing of data NLTK library could have been used which provides an easy platform for various pre-processing of data.

The comparison of SVM, Naïve Bayes & Logistic regression has been done on a Twitter dataset for sentiment analysis by Poornima, A. and Priya, K.S. (2020). In this research paper apart from sentiment analysis sentiment polarity detection has been done with the help of the term frequency method. The Twitter dataset used contains tweets from various people and hence it has emoticons, URLs, symbols, and references to the people, and extra punctuations. The pre-processing has been done with the help of Stemming, Lemmatization, and stop word removal and data has been converted into lower form. In this research, the author has used pre-processing techniques very effectively since it impacts the output of the model and it is represented very well as compared to Dey, S et al. (2020). The logistic regression model has achieved the highest accuracy of 86% as compared to the other two models. From this research paper pre-processing techniques can be taken as a guide to use on other datasets, however, this research has only restricted to the comparison of models. Instead, a hyperparameter could have been used to improve the performance of models.

The Naïve Bayes and SVM are the most commonly used models for sentiment analysis and these models have been proven to be effective in various research. However to check the performances of other models Mitra, A., (2020) has carried out experiments on Twitter data. In this research two approaches were used, in the first approach with Naïve Bayes & SKlearn NBM and SVC were used and in the second approach Decision Tree, Random Forest & KNN were used. The sentiments were classified into 5 categories to check whether the model will be able to predict the accurate result or not. This study has also utilized the pre-processing methods used by Poornima, A. and Priya, K.S. (2020). Approach 2 has resulted in better accuracy as compared to approach 1. However, the NLTK methodology has not been explored in this research and the model could have been tuned with the help of various Hyperparameters to achieve better accuracy.

Villavicencio, C. (2021), using Naïve Bayes classifier, conducted a sentiment analysis of comments made in response to the Philippine government’s efforts to distribute COVID-

19 vaccination. The tweets done by the public have been classified into English and Filipino to extract Positive, Negative, and Neutral sentiments with the help of Natural Language Processing. To prepare the dataset tweets from dedicated hashtags have been collected and all merge into one data frame using RapidMiner software. To label the dataset, data annotation has been used and pre-processing has been done with the NLTK technique. Apart from using NLTK for pre-processing, the tag replacement and case transformation has been done. This experiment has outlined the previous experiments done and achieved the highest accuracy using the Naïve Bayes classifier. However, the research was restricted to only one modeling technique and approach used by Poornima, A. and Priya, K.S. (2020) for pre-processing & data modeling can be adopted.

In the above section, various traditional machine learning approaches have been described and it has been observed that for data modeling Naïve Bayes, and SVM classifier has been used very prominently. From the above research, various pre-processing techniques can be adopted, however, the NLTK library can be used for better data pre-processing. The above methods have major drawback of dependency on analyzing repetitive patterns of data. In the next section, various deep learning approaches have been discussed with the help of various research done.

2.2 Deep learning approaches for Sentiment analysis

Opinion mining using traditional machine learning methods were having the advantage of scalability, however, it is based on learning the repetitive patterns and words from the text. The recent deep learning methods have replaced traditional methods because of the capability of processing a large amount of data and the capability to persist the data. In the following paragraphs, various approaches used by researchers have been discussed.

2.2.1 Recurrent Neural Network (RNN) approach for Sentiment analysis

The use of Term-Frequency inverse document frequency (TF-IDF) and word embedding methods on 3 distinct deep learning models has been done by Dang, N.C. et al. (2020). For this research 8 different datasets have been chosen and comparative analysis has been performed on these datasets. In this research to obtain the final data frame, various pre-processing methods have been used as done by Han, K.X. et al. (2020) and Mitra, A., (2020). This research has provided a comparative analysis of model evaluation as well as the processing time required for the model. The deep neural network (DNN), Convolutional neural network (CNN), and recurrent neural network (RNN) have been compared with the help of TF-IDF and word embedding. The research showcase that the RNN model has required the highest processing time and computational capacity, whereas DNN has performed exceptionally well when evaluated on basis of recall. This research will help to select the modeling technique depending on the nature of the dataset.

The Recurrent Neural Network (RNN) and Word2vec models have been used in many types of research of sentiment analysis. The sentiment analysis using these models on the Indonesian Traveloka website has been done by Kurniasari, L. and Setyanto, A. (2020). The research has proven to be a novel approach in the Indonesian language and for these positive & negative reviews have been taken for analysis. For Word2vec the data has been converted in the form of distinct vectors and provided as input to the model and for RNN concept of hidden state vector has been used. For this research, the author has used 3 different hyperparameters to tune the model. To test the performance of the model the dataset has been tested using Naïve Bayes and CNN models. Dang, N.C. et al. (2020) could have used hyperparameters used in this research for the evaluation of 3 different deep learning techniques as done in this research.

Another study using a Recurrent Neural Network has been done on the IMDB movie review dataset. The study has showcased the importance of assigning random weights. These weights are then propagated back to get the optimum result. The Word2vec and TF-IDF have been used for feature extraction. The model has been trained to the point where an error has been reduced. Patel, P. et al. (2021) has compared the result with Naïve Bayes and SVM by performing the test on these models. The approach used is similar to that proposed by Dang, N.C. et al. (2020), however, in this research model has trained with different epochs.

In the above paragraphs the research performed using a recurrent neural network has been described. In these researches hyperparameters have been used, however, there are high chances that the model has an overfitting issue since only a fixed number of epochs has been used to train the model. Also, RNN models require more processing time and high computational power. RNN models also have the drawback of vanishing gradient problems. In the next section concept of RNN-based LSTM has been described.

2.2.2 RNN-Based Long Short-term Memory (LSTM) approach to sentiment analysis

To achieve good results and efficiency pre-processing of the model is very important. Qaisar, S.M. (2020) has emphasized specifically on data pre-processing in research conducted on the IMDB movie review dataset using the LSTM model. For this research 5k positive, and 5k negative reviews were taken to avoid biasness in the data and 89% accuracy has been obtained. The dataset contained raw public comments hence various pre-processing techniques have been implemented. In this research other deep learning models could have been implemented for comparison purposes. A similar approach to the LSTM model has been used on the IMDB benchmark dataset by Murthy, G.S.N. et al. (2020) and for this research 50000 reviews were taken and various epochs and batchsizes has been taken as part of the experiment and comparison has been made. The comparison between both types of research indicates that Qaisar, S.M. (2020) has not performed hyperparameter tuning and thus achieved less accuracy.

Instead of analyzing only text data for sentiment analysis users' theme preferences from social media sites, and text sentiments have been used by Zhao, J. et al. (2020) to perform sentiment analysis using the LSTM model. In this approach, the attention layer has been used along with the LSTM to get the final output of the model. In this research various number of neurons has been taken to perform the analysis. Li, W. et al. (2020) used LSTM for sentiment analysis, however, this research has focused more on solving the challenges faced in sentiment analysis and hence used novel sentence padding. The research has focused on solving the complexity in data and to resolve this complexities the input data has been sampled. To resolve the complexities in the input data, the data has been sampled in consistent size and thus the proportion of sentiment in the text has been improved. On input data CNN model has been applied and then LSTM has been used. Zhao, J. et al. (2020) could have used the same approach, however, the attention mechanism was preferred as an additional layer along with input data.

The sentiment analysis has been performed often frequently using the English language in various research. However, to address the scope of sentiment analysis in other languages, Dashtipour, K. et al. (2021) have performed the research using Persian movie reviews and compared the CNN & LSTM models. For effective data pre-processing NLTK Tokenizer and stemmer have been used. To train the LSTM model Adam optimizer has been used with a dropout probability of 0.2 and to train the CNN input data has been converted into a 300-dimensional vector with fastText embedding. The approach of the NLTK library could have been used by Qaisar, S.M. (2020) since pre-processing plays important role in models outcome. Instead of comparing CNN with LSTM Mengistie, T.T. and Kumar, D. (2021) have used a hybrid approach of CNN and LSTM on Covid-19 public tweeter data. To tune the model 10 epochs and 128 batch size have been used as a hyperparameter and the model has yielded 99% accuracy. In this research, there are very high chances that the model has overfitted the data. Early stopping and checkpoint could have been used to get the valid epochs and to avoid overfitting.

In section 2.2.1 and 2.2.2 the Recurrent neural network and LSTM approaches have been discussed. However, both of this approaches has common drawbacks that it takes a longer time for a model to get train and also there are high chances of model overfitting. In following section state of art transfer learning approaches have been discussed. In these approaches the model is first trained on one task and then later by using this model other tasks can be solved.

2.3 Pre-trained transfer learning approaches for sentiment analysis

To surpass the drawbacks of RNN & LSTM-based models an attention-based mechanism has been developed by Google research scientists known as Transformer based approach Vaswani, A. et al. (2017). This mechanism is based on an attention layer and along with that stack of the encoders at input & decoders at the output. Each of the encoders consists of a multi-head self-attention and feed-forward network and in the decoder, there is a third sub-layer whose role is to perform multi-head attention over the output of the encoder stack. To convert the input & output tokens into vectors embedding has been used and softmax has been used to convert the decoder output to next token probabilities. The entire experiment has been carried out on an English to German translation dataset and 4.5 million data has been used for training. One of the transformer model is BERT which is trained by google on a Wikipedia dataset and it was mainly trained for next sentence prediction.

González-Carvajal, S and Garrido-Merchán (2020) have performed comparison of BERT and traditional machine learning algorithms on various datasets. Initially, TF-IDF has been tested as vocabulary feed and later on BERT has been used for comparison. The comparison has been made on basis of accuracy and precision. The result achieved indicated that BERT has outperformed other algorithms and thereby for NLP tasks BERT can be used prominently. On similar background, Fimoza, D. et al. (2021) used BERT on Indonesia movie review dataset. To make the model more robust epochs, batch size and learning rate were chosen as a hyperparameter and an experiment has been conducted. This study also concluded that BERT has outperformed other models.

The novel approach of BERT for Twitter sentiment analysis has been done by Pota, M. et al. (2020). In this research first Twitter jargon like emoji's, emoticons, and the hashtag has been transformed into plain text, and then the BERT model has been pre-trained on plain text and then it has used to classify the tweets. The approach of pre-training the BERT on plain corpora instead of tweets has helped to achieve better results in Subjectivity classification and Polarity classification. Chouikhi, H. et al. (2021) has conducted research on Arabic sentiment analysis using BERT. The research was very challenging since the Arabic language was very complex and also involves various dialects. To train the model Arabic BERT has been used instead of Base BERT. In this research plain text could have been used for model training as done by Pota, M. et al. (2020). To make the model more robust various Hyperparameters such as Batch size, dropout, and epochs have been used.

The pre-trained transformer based models has proven to be efficient as compared to other traditional methods of sentiment analysis. These transformer models can be further used to identify popular contexts based on the data available on the internet. Asgari-Chenaghlu, M. et al. (2020) has used a Universal sentence encoder (USE) to identify the trending topic going on Twitter based on pandemic data. In this research, sentence similarity has been used and data has been fed to k-means clustering to identify the semantic sense & then group the similar tweets. The proposed model has outlined other approaches such as LDA & TF-IDF. In recent years aspect-based sentiment analysis (ABSA) has been used prominently by businesses to understand the thinking of users regarding products as well sentiment while writing the review. Mohammad, A.S. et al. (2021) have performed ABSA on hotel data based on Arabic reviews using Gated recurrent units and a Universal sentence encoder. From this research two tasks have been achieved aspect extraction & aspect polarity classification. The GRE + USE has resulted in a robust model with a 93% f-1 score and has achieved a 62.1% rise in f1-score.

Summary of Literature review

Regardless of much previous work done on Sentiment analysis using various machine learning techniques, there are few shortcomings present in those models. In the above literature review comparison of various research papers has been made on basis of techniques used, advantages, and shortcomings. The LSTM models are advancements of RNN models and many researches have used LSTM and very effective results have been achieved. The pre-trained transformer based approaches has been evolved in recent years and because of their wide advantages, this models are used in sentiment analysis. Hence in this research paper, LSTM, BERT & USE will be used as a methodology and a comparison will be stated in the results and evaluation section.

3 Research Methodology and Specification

Any research's ultimate objective is to offer a noteworthy methodology because it reflects the researcher's perspective. To perform any Natural language processing task, data understanding is required. Data mining involves understanding the data and various methods that can be implied to prepare the data for modeling. The widely used methods are KDD and CRISP-DM. There is one more method which is known as SEMMA but it does not involve any specific structure. In the Natural language processing domain, usually, the data will be complex. The sentiment analysis involves dealing with raw data in which effective data pre-processing and cleaning techniques are required and hence KDD methodology has been chosen. In terms of KDD, a schematic representation of various steps undertaken for sentiment analysis has been described in figure 1.

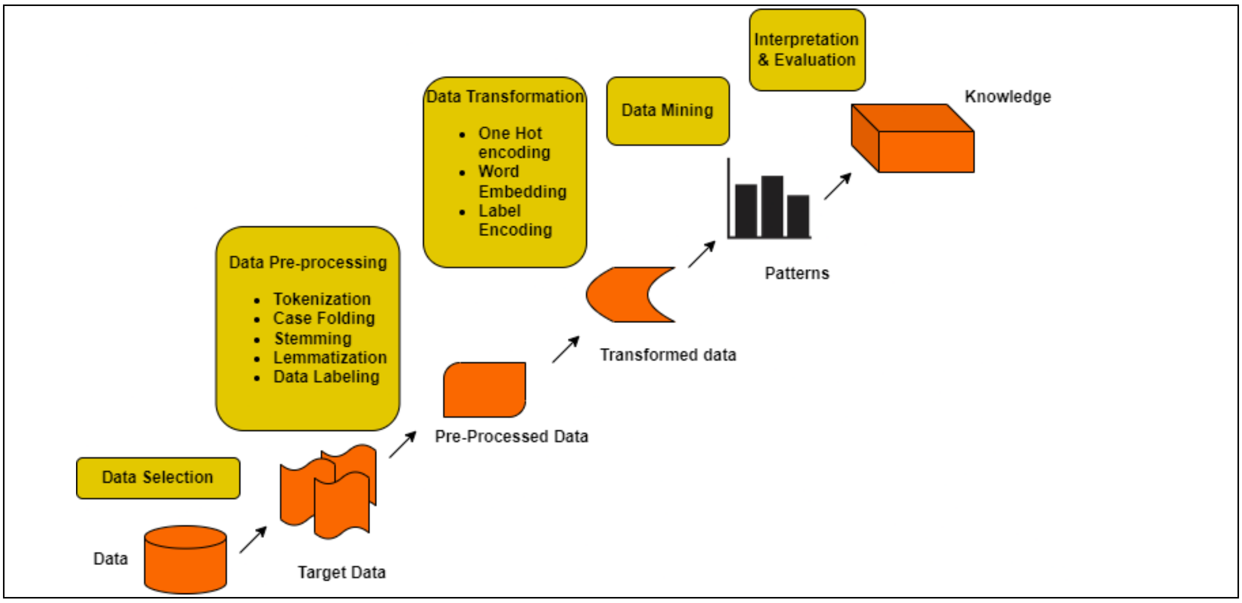

Figure 1: Process flow in terms of KDD

3.1 Domain knowledge

Having a thorough knowledge of the domain and context is the most crucial phase in any project. How the data is getting processed, the methodology involved, what are the challenges associated with it, and how the results have been analyzed are the primary factors that generally come into focus. The proposed research falls in the sentiment analysis domain and it is a very high emerging field. Various businesses and companies are using sentiment data and making big financial decisions. This domain involves high risk in taking business decisions since it is very important to understand the context of the sentiment and it is very easy to get misled by the sentiment. The pre-trained transfer learning-based approaches are now emerging as a powerful tool in the sentiment analysis domain and effective evaluation is required to implement it in real-life scenarios.

3.2 Data Understanding & Exploration

Table 1. Dataset Description

#	Feature	Data Type	Description
0	**Review**	**String**	**Reviews posted by users on IMDB Website**
1	**Sentiment**	**Categorical**	**Corresponding sentiment based on a review**

The next phase in KDD based model is understanding the data. The data understanding establishes the groundwork for future results. For the data gathering, it is crucial to identify the source of the data. The process of gathering data raises several ethical issues. The dataset used in this research has been taken from the publicly available repository Kaggle[3] and hence there are no ethical concerns associated with it. The dataset contains 2 features, the first one is reviews posted by users on the IMDB website and the second one is the corresponding sentiment of the review. The dataset must contains an equal number of target variables to avoid the biasness. In this research, both the positive and negative sentiments were balanced with 25000 positive & 25000 negative sentiments.

[3] https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

The distribution of positive and negative texts has been illustrated in Figure 2 below.

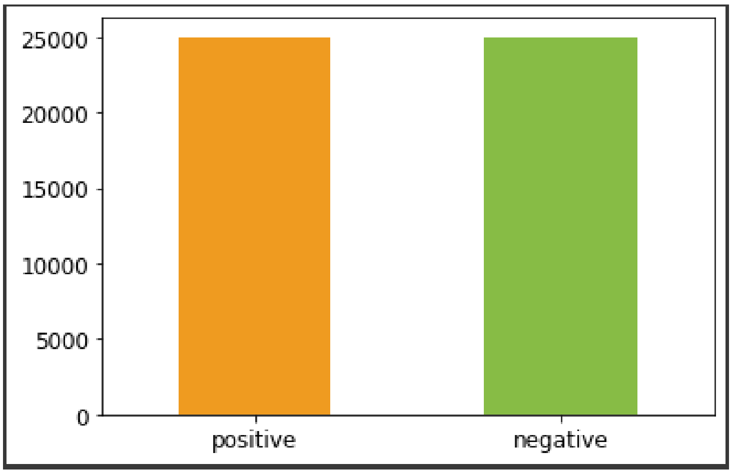

Figure 2. Distribution of target class

3.3 Data Cleaning and Pre-processing

After data exploration is done, then comes a very crucial phase, preparing the data so that it can then use for data mining. This phase is influential in data mining since it has a direct bearing on the model's and the study's findings' quality. The majority of the time, data is filled with irrelevant elements that degrade the model's quality & performance. The data may also include missing values, extra spaces, special characters, numerical values, and other elements. If the data is not handled appropriately, crucial information may be lost and the desired result may not be achieved. While performing the research, the quality of the data is more significant than the number of records and features. A small number of features can also help create a robust model and can be more significant when predicting the target variable. These irrelevant features sometimes result in increasing the cost of research and training time. Python Natural language toolkit (NLTK) contains the various libraries for effective pre-processing of data and hence it has been used in this research as a primary tool for pre-processing. There were various pre-processing steps required while performing this research and each of which has been explained below.

.

- **Data Labeling**: The data labeling in sentiment analysis involves the creation of an additional numeric class to provide the model. In this research, labeling has been done to classify the sentiments into two categories. The Positive sentiment has been labeled with value ‘1’ and the negative sentiment has been labeled with value ‘0’. For some models, it is mandatory to provide the labels in the form of numerical data.

- **Case folding**: The process of case folding has been used in these research to convert all the uppercase characters into lowercase characters. This process is majorly carried out in Natural language processing since it helps in making generalizations. The uppercase and lowercase characters have the same meaning but when we make entire data into lower dimensions, it helps in avoiding the creation of more vector spaces. If both dimensions are present in the text then it is represented as two different words and affects the model’s performance.

- **Stop words & frequently occurring words removal**: The stop word removal has been used in this research as the reviews were regarding movie reviews and there were a lot of pronouns and articles. Using this process commonly occurred articles & pronouns have been removed. This process has been carried out to focus more on high-level information & thereby ignore the low level information.

Stop words: 'the', 'what', 'so', 'an'

- **Stemming**: Stemming is one of the most important pre-processing operations which is required to carry out every NLP task. In this process a part of the word is getting removed and which helps to convert to its base form or root. There are few algorithms in NLP that decides how to chop the word. If much large part of a word is chopped off then it leads to a condition known as overstemming and if two or more words have been reduced to one root word unnecessarily then it is known as understemming. The example of stemming has been given below.

send, sent, sending → send

- **Normalization**: Normalization is another crucial step in Natural language processing. An attempt to carry out the normalization has been performed in this research to check the randomness of a text so that it can be converted into well standard. In normalization, the case of text is converted into one form (upper or lower) and punctuations will be removed thereby maintaining the text consistency. With the help of normalization overload on computation will decreases since it reduces the amount of information the computer has to deal with and thereby helps to increase efficiency.

- **Lemmatization**: Lemmatization is another process that is used in this research to convert to its normalized form or lemma. The algorithms refer dictionary to understand the base form of the word and convert it to its root word. In lemmatization, canonical form has been obtained and it depends on the lemma of the word. WordNet is a lexical database that helps to identify the semantic relations between words. WordNet has been used for the lemmatization process in this research the using WordNet library.

Feet → foot or better → good

- **Handling special characters**: The public reviews contains lots of Html tags, hashtags, emoji's, emoticons, URL'S and special characters. In this research, these characters have been removed from the text and then the remaining data has been used for modeling. The re.sub method has been used to substitute the above mentioned characters.

- **Tokenization**: Every Natural language processing pipeline starts with tokenization. This process has a very significant impact on the entire NLP pipeline. In this process, the unstructured text is split into small chunks of discrete elements. With the help of tokenization sentences, words, characters, and subwords can be separated into individual chunks. Tokenization also helps to identify the frequently occurring words in the entire data.
  In Figure 3. Most frequently occurred words in the reviews has been represented in the form of bar chart.

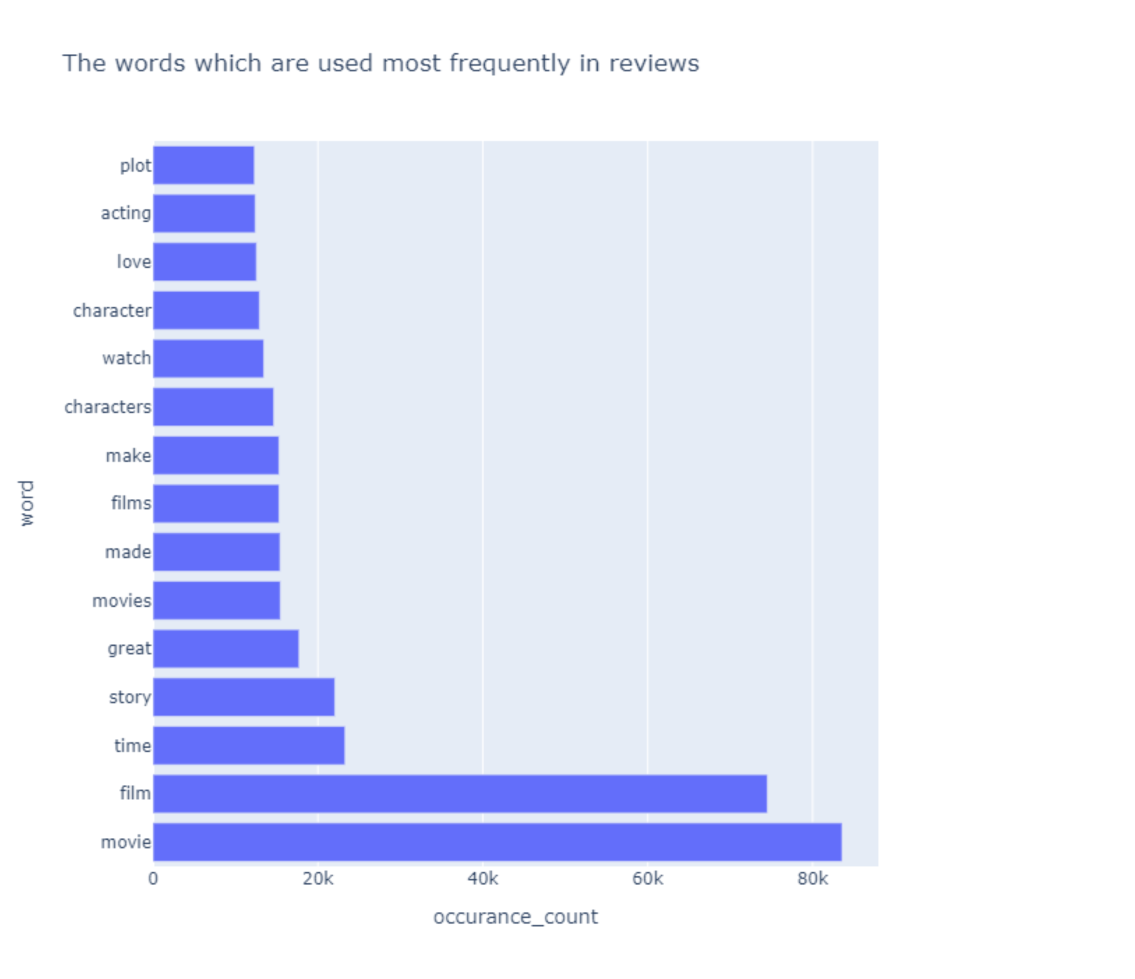

Figure 3. Most Frequently occurred words

3.4 Data Transformation

The data transformation phase involves transforming the clean data into a form that can directly be provided to the model for further processing. The data by this stage is cleaned, validated, and ready to use. In this research two methods have been used for data transformation. By performing the data transformation the data quality has been improved and it also helps to reduce the computational load. Two techniques have been used for data transformation Embedding and One hot encoding.

- **Word Embedding**: Word embedding has been used to identify how words are represented for text analysis. It typically takes a real-valued vector that encodes the meaning of a word. The words in reviews that are adjacent to each other actually has a similar meaning.

- **One hot representation**: The one hot representation has been used in this research to convert the texts into numerical representations to provide into the model. Since the huge amount of text data has been converted into a numerical form the processing of the model has been increased and it has also helped in reducing the computational load. The one hot representation has been done with respect to vocabulary size and the corresponding index has been obtained.

- **Label encoding**: The label encoding has been performed to take the texts in the form of labels and then convert them into numerical form starting with 0. Then with the help of keras, it has been converted into categorical one-hot vectors. The encoders have been used to perform the label encoding on the input text.

3.5 Data Mining

Data mining is a final process in which pre-processed and transformed data derived from raw data has been provided to various classification models and evaluation will be done. To identify an optimal model that performs well in contrast to others in terms of accuracy and performance, various models were built and assessed. Two separate approaches were used to build the models. In the first approach RNN- based LSTM model has been used and in the second approach pre-trained transfer learning-based models BERT & USE has been used. Further, the models were tuned using Hyperparameters such as Epochs and batch size. To avoid the overfitting of the models early stopping and checkpoint have been used. The literature review indicates that BERT and USE have not yet been prominently explored in this domain and hence these models have been implemented to compare the performance with the LSTM models.

4 Design Specification

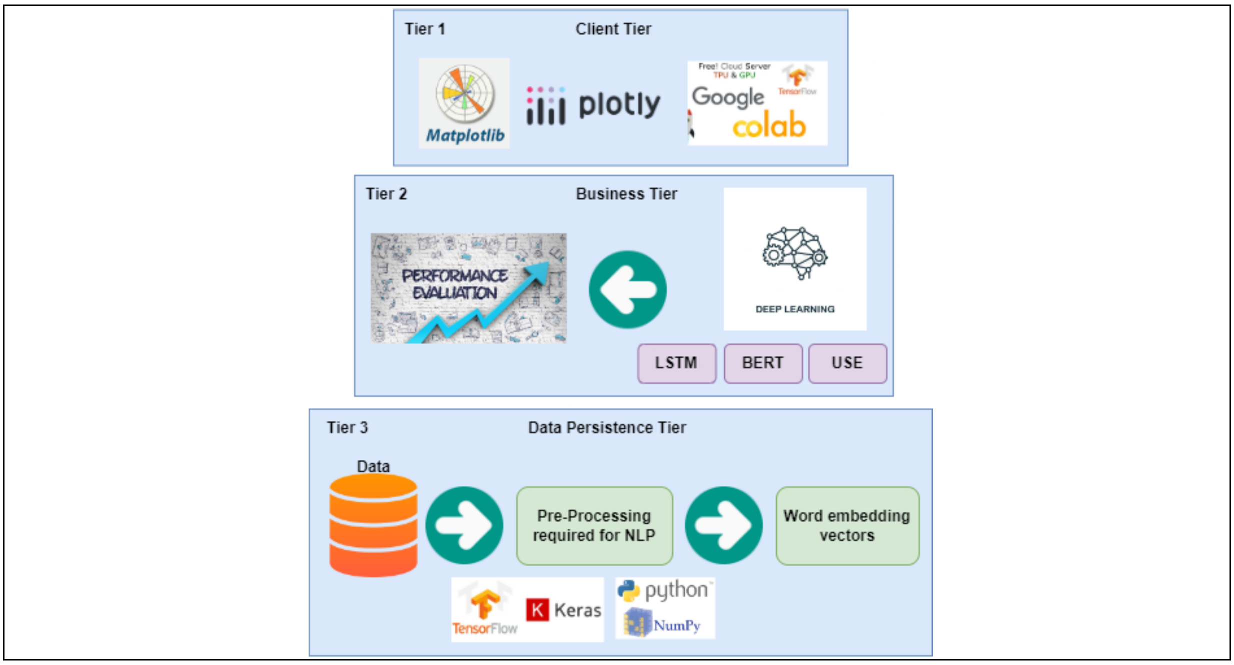

Figure 4. The design specification strategy

The design specification of the research comprises three tiers from data loading to final visualizations and it is illustrated in Figure 4.

- **Data Persistence Tier:** In this tier, the dataset required for the research has been loaded, and pre-processing & word embedding has been done. The various packages of python such as Numpy have been used.
- **Business Tier:** In this tier, the deep learning models used in the research, with the respect to business logic have been implemented. The Keras & TensorFlow libraries have been used.

- **Client Tier:** In this tier, the expected outcome from the previous two layers will be presented to the users in the form of numerical values as well as visualizations. The Google colab Pro has been used as an IDE.

Python was chosen as a programming tool because of its numerous advantages over other languages. It is a very easy-to-use tool, as well as it supports a wide number of libraries which helps to integrate effectively. TensorFlow has been used since it provides a collection of various workflows to develop and train the models and Keras has been used to implement the models used in this research. The Jupyter notebook has failed to handle a large number of records present in the dataset hence Google Colab Pro has been used as a Web Integrated Development Environment (IDE).

4.1 LSTM

LSTM models are special advancements of Recurrent Neural networks (RNN) specifically designed to solve the vanishing gradient problems faced by RNN. As the name suggests it allows the information to persist thereby helping to handle the long-term dependencies. The LSTM cell consists of three different parts or gates. The first one is Forget gate which practically decides whether to retain the previous state or information from the previous timestamp and to take the decision it makes use of the **sigmoid** function. Since the sigmoid has been used if the output is 1 then the information will be retained, if the output is 0 then the information will be discarded. The second part consists of the Input gate, which allows the new information to come by quantifying its importance. The new information will be passed with the help of the **tanh** function and for that, the value ranges from -1 to 1. The third part consists of the output gate where the token with the maximum score is considered as output. Because of gate architecture, the flow of the information will be better in LSTM models.

4.2 Transfer learning

Transfer learning is a machine learning technique in which a model created for one task is utilized as the foundation for a model on another task. The BERT and USE are transformer-based models used in this research paper and generalized architecture has been presented in Figure 5. The components of transformers[4] have been illustrated below.

Encoder: The encoder is where the input data is fed into the model. The encoder consists of a self-attention layer and a feed-forward neural network. The self-attention layer allows the encoder to look at the other sentences in the text in the process of encoding. The output of the self-attention layer is then fed into the feed-forward neural network.
Decoder: The decoder provides the output of the model. The decoder consists of self-attention and a feed-forward neural network and in addition, it contains an encoder-decoder attention layer which helps to focus on relevant parts of input sentences.
Embedding: In every NLP task first the words in input text are converted into a vector with the help of embedding. At the bottom-most encoder the embedding happens and after embedding the data flows through each layer of encoders.
Self-attention: With the help of the self-attention layer the comparison can be done between the input sequence with each other and the corresponding output can be modified. It performs the key-value searches for the input sequence and adds results to the output sequence.

---

[4] https://jalammar.github.io/illustrated-transformer/

Pre-processor: With the help of the pre-processor all the word embedding will be performed and directly the data can be fed into the model. The pre-processor is available in TensorFlow hub and can be downloaded and used.

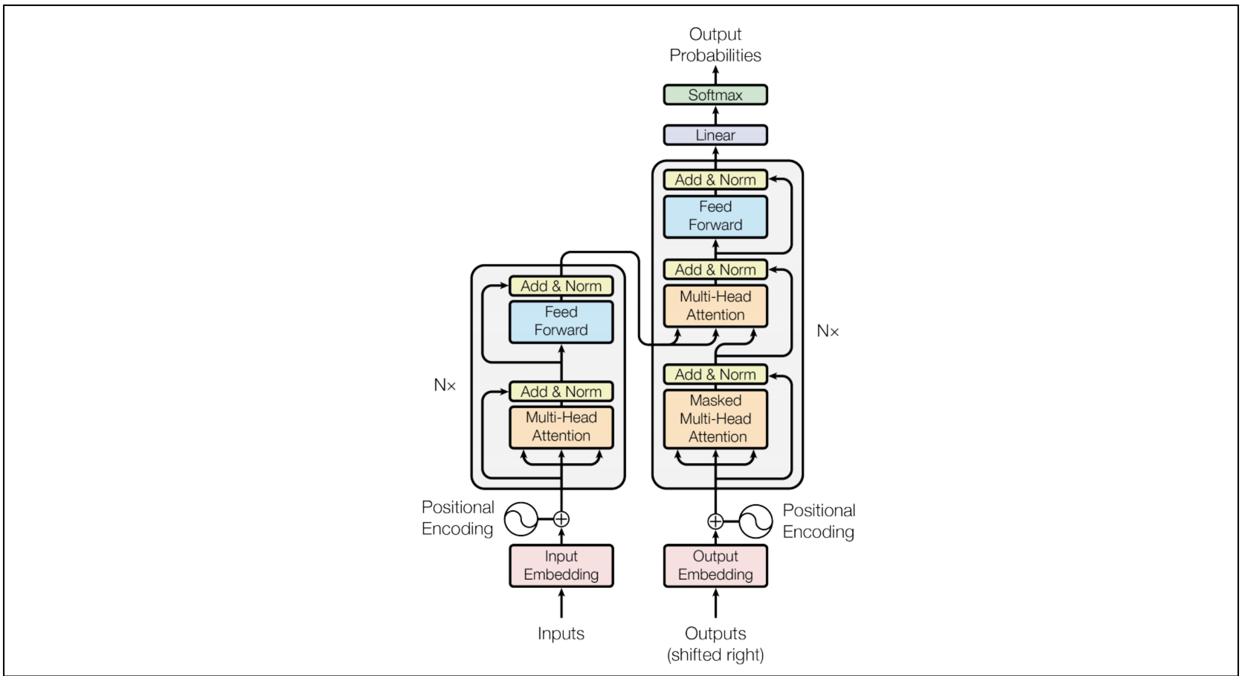

Figure 5. Transformer architecture Vaswani, A. et al (2017)

5 Implementation

The data implementation is a final step in which the final prepared data has been used for different modeling techniques. In this section various processes and steps carried out in order to execute the models have been discussed. Along with that parameters used in the modeling have been explained.

5.1 Development Environment

For deployment of machine learning projects the Python and R languages have been widely used. In this research project, the entire implementation has been done using Python (3.7.13) and Google Colab Pro has been used as a web Integrated development environment (IDE). The Google Colab Pro has provided an additional amount of GPU and RAM and thereby it has helped to reduce the execution time. There are numerous collections of workflows that are supported by python like TensorFlow and Keras which makes the implementation simpler have been used in this research. The Pandas and Numpy have been imported during the research as it was required for implementation. The Matplotlib, Plotly, and seaborn have been used to visually analyze the results.

5.2 Data Handling

The dataset required for this research has been taken from the publically available repository Kaggle and various irrelevant data has been removed using pre-processing techniques thereby retaining only relevant data. The Numpy has been used to transform the cleaned data into train, validation, and test arrays before actually applying various modeling techniques. When we pass the parameters to the model it is not necessary for the model will give the optimum outcome. To make the model more robust numerous Hyperparameters have been tried and evaluated. To check the outcome of the trained model, the generated model has been saved in google drive and later on called with the help of the load function of Keras library. Then the test data has been tested on the generated model.

The data has been partitioned in the Training set, Validation set and Testing set using Numpy arrays. The original data had 50000 records, out of which 37500 records have been used for training, 10000 records have been used for Validation, and 2500 records have been used as a testing set. Various ratio of train, validation and test split has been tried and the best result has been obtained using the combination mentioned.

5.3 Use of Early stopping and Model checkpoint

In every machine learning research, apart from accuracy, the loss function has very significant importance on the overall performance of the model. The loss defines how good the model is in terms of predicting the expected outcome. It is very common in machine learning that the model gets overfitted when trying to train it with the provided number of parameters. There must be some attributes that control the overfitting issue and to handle this early stopping and model checkpoint has been used in this research. This study's objective is to limit the loss to a minimum while still achieving good accuracy. The early stopping function stopped the training of the model when validation accuracy did not improve by 0.01. This also helps to provide the optimum number of epochs required while training.

```
# Early stopping and Model Checkpoint
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping

# early stopping
es = EarlyStopping(monitor = 'val_accuracy',min_delta = 0.01, patience = 3, verbose = 1)

# model check point
mc = ModelCheckpoint(filepath = '/content/drive/MyDrive/best_model_LSTM.h15', monitor = 'val_accuracy',min_delta = 0.01,
                     patience = 3, verbose = 1, save_best_only = True)

cb = [es,mc]
```

Figure 6. Early stopping and Model checkpoint

In Figure 6, various parameters used in the research to control the overfitting of the model can be seen.

- **monitor**: With the monitor parameter the validation accuracy will be observed throughout the training time.
- **min_delta**: With the help of minimum delta the improvement in validation accuracy has been defined. If the validation accuracy did not improve by 0.01 then the training of the model will stop.
- **patience**: The value of patience has been taken as 3. After 3 epochs if the min_delta value did not improve by 0.01 then the model will stop training and early stopping will be called.
- **verbose**: The verbose has been set to 1, hence when a callback has been called the model will display a message stating early stopping has been performed.

Through the model checkpoint when once the epoch has been performed the model has been saved to Google Drive on the desired path. After performing each epoch the best model has overwritten to google drive.

5.4 Model Implementations

Section 4 of the research classification models used in this research has been discussed. In this section, various parameters used in those models, Hyperparameters, data partitioning, and techniques used to avoid the overfitting of the models have been explained.

5.4.1 Long short-term memory (LSTM)

The reviews have been converted and saved in the form of a dictionary. The vocabulary size of 5000 has been taken. Each word in the sentences has been converted in the form of the index from the vocabulary size. Padding sequence has been used since each sentence has a different length and to make them uniform pre padding of 100 has been used. It has added the 0 in the form of an index to make the embedded docs uniform. 40 vector features have been taken and a model has been created. The sequential modeling has been performed and hence sequential layer has been created. The next layer is the embedding layer in which the vocabulary size, embedding vector & sentence length has been passed. After which the LSTM layer has been created with 100 neurons and it has faded into dense layer where the sigmoid activation function has been used since the output is binary classification. Initially model got over fitted and hence dropout layer has been added. Various parameters used for modeling has been listed below in Table 2.

Table 2. Parameters used for model training

Parameter	Value
Vocabulary size	5000
Padding sequence	100
Epochs	9
Batch size	64
Trainable Parameters	405729

5.4.2 Bidirectional encoder representation from transformers (BERT)

The BERT is a transformer-based pre-trained model and hence the pre-processor is available at the TensorFlow hub[5]. The pre-processor and BERT model has been downloaded from TensorFlow. The pre-processor has been used to convert the input text into tensors as required by the model. Then it has been provided to the encoder and to get the output pre-processed text has been given to the decoder. The sequential model has been used and the dropout layer with the value 0.1 has been added to avoid overfitting. The dense layer has been used at the output layer with a sigmoid activation function. The compilation has been performed with Adam optimizer and for loss, the binary cross entropy has been used. The trainable parameters are 769 since 768 as an input and 1 layer as an output. In Table 3 various parameters used in the modeling has been listed.

Table 3. Parameters used for model training

Parameter	Value
Vocabulary size	5000
Epochs	11
Batch size	64
Trainable Parameters	769

[5] https://www.tensorflow.org/hub

5.4.3 Universal sentence encoder (USE)

For implementing a Universal sentence encoder pre-trained USE model has been downloaded from the TensorFlow hub. The label encoding has been performed and numerical data has been provided to the encoder. Then encoder converted the text into labels and one hot vector. The input has been provided to the decoder to convert the vectors into indexes and then inverse transformation has been performed to again convert back the data to text. The pre-trained USE model has been provided to the functional lambda layer and in this process, the string has been converted into numerical vectors. The input layer with 1 neuron has been provided to the embedding layer where universal embedding has been performed and which resulted in 512-dimensional vectors. After that dense layer has been used with 256 neurons and a non-linear RELU activation function has been used. Finally, the output has been obtained at a prediction layer with a sigmoid activation function.

Table 4. Parameters used for model training

Parameter	Value
Non-linear activation function	Relu
Epochs	7
Batch size	64
Trainable Parameters	131842

In table 2, 3, and 4 particular to the model have been listed. In the following Table 5, common parameters used in all the models have been listed.

Table 5. Common Parameters used in the model

Parameter	Value
loss	Binary- cross entropy
Optimizer	Adam
Activation function at the output	Sigmoid
Metrics	accuracy

The loss function is very important in every machine learning research since the evaluation of the model is mainly done on basis of loss. In this research since the output is binary, binary cross-entropy has been used. The Adam optimizer has been used for the optimization to handle the gradient descent and it is very efficient since it consumes less memory[6]. The sigmoid activation function has been used at the output since there are two outcomes. The probabilities above the value of 0.50 will be positive reviews and below 0.50 will be negative reviews. The metrics have been taken as accuracy to check how the model is performing. The metrics don’t impact the model’s performance, any value can be used such as precision, recall, or f1-score.

6 Evaluation and Interpretation of Results

In this section detailed analysis of results achieved after implementations have been discussed. The primary focus of the research is to check the performance of the transfer learning models for sentiment analysis. The Universal sentence encoder (USE) is

[6] https://www.geeksforgeeks.org/intuition-of-adam-optimizer/

comparatively very new for this domain and has not been explored prominently. The same parameters of early stopping and checkpoint have been used across all three models. The evaluation has been carried out based on loss, accuracy, precision, recall, and f1-score. Along with that summary of a confusion matrix, has been discussed to evaluate the actual values and corresponding values predicted by the model.

6.1 LSTM

The Figure 7. Shows accuracy and loss curve obtained after implementing the LSTM model. It can be seen from the figure that after 2 epochs the training & validation loss has decreased rapidly and accuracy has increased very quickly. At epoch 7 the validation loss has been increased and hence early stopping has been called at epoch 8. The point that need to note here is that after epoch 8 early stopping has been called since there was no improvement in the validation accuracy and model checkpoint has occurred.

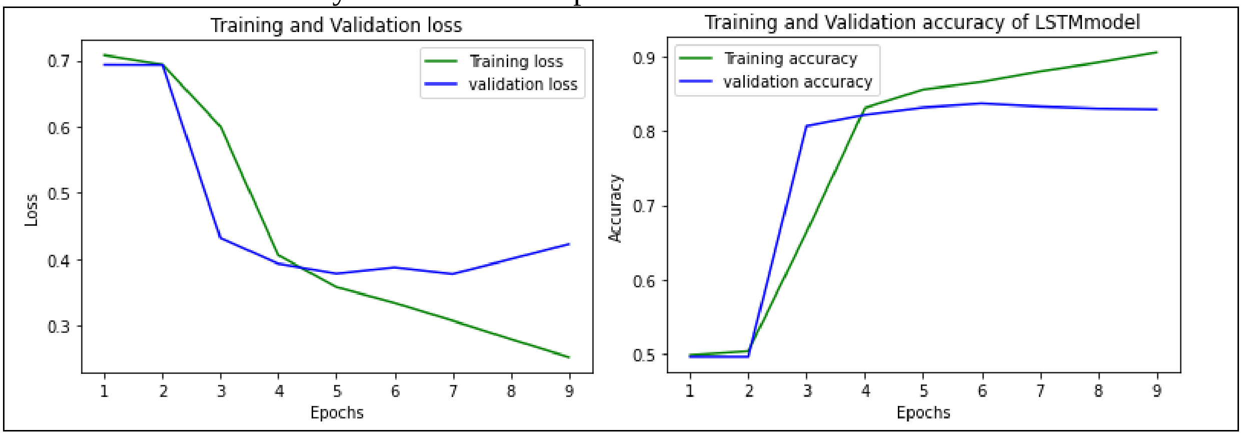

Figure 7. The accuracy and loss curve of the LSTM model

Confusion Matrix-

The Figure 8. Illustrates confusion matrix of LSTM model. The 0 indicates negative values whereas 1 indicates positive values. The validation data contained 10000 records out of which True positive reviews predicted by the model were 4441 and True Negative reviews predicted by the model were 3925. It can be seen from the confusion matrix that the number of False Positive reviews was 598 and the number of False Negative reviews was 1036.

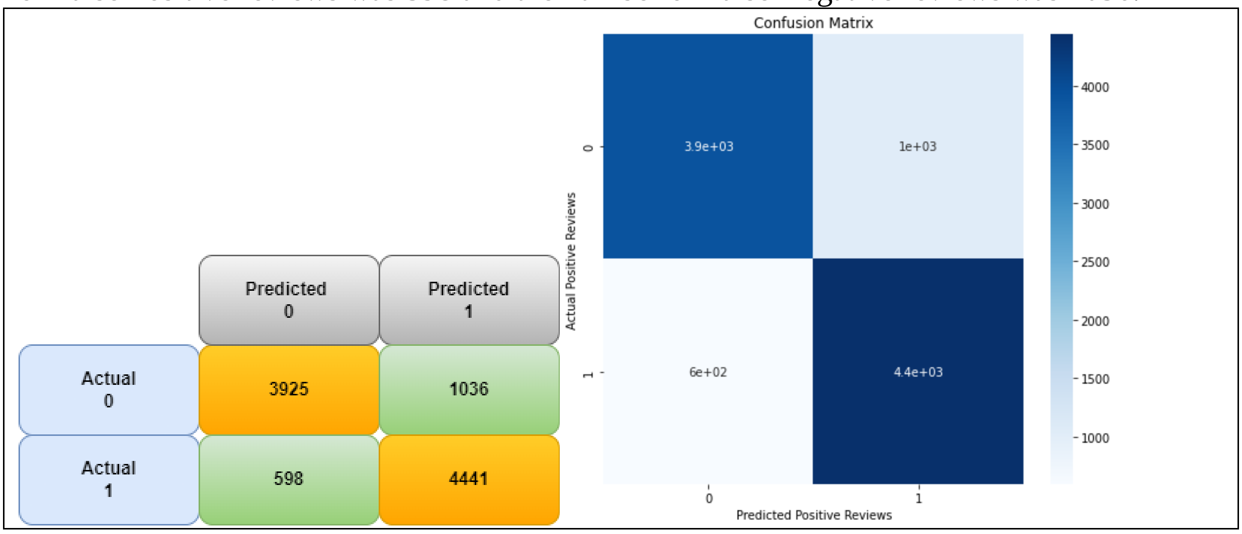

Figure 8. Confusion Matrix of LSTM model

6.2 BERT

From Figure 9 it can be seen that after 2 epochs the training & validation loss has decreased in a very steady manner and the same pattern has been followed by both the losses. In terms of accuracy, both the raining & validation set has followed the same pattern, and accuracy has increased after the 2^{nd} & 4^{th} epochs. For validation data, there are some up & down in terms of accuracy as seen from the curve. The optimum accuracy has been achieved after the 11 epochs and early stopping has been called.

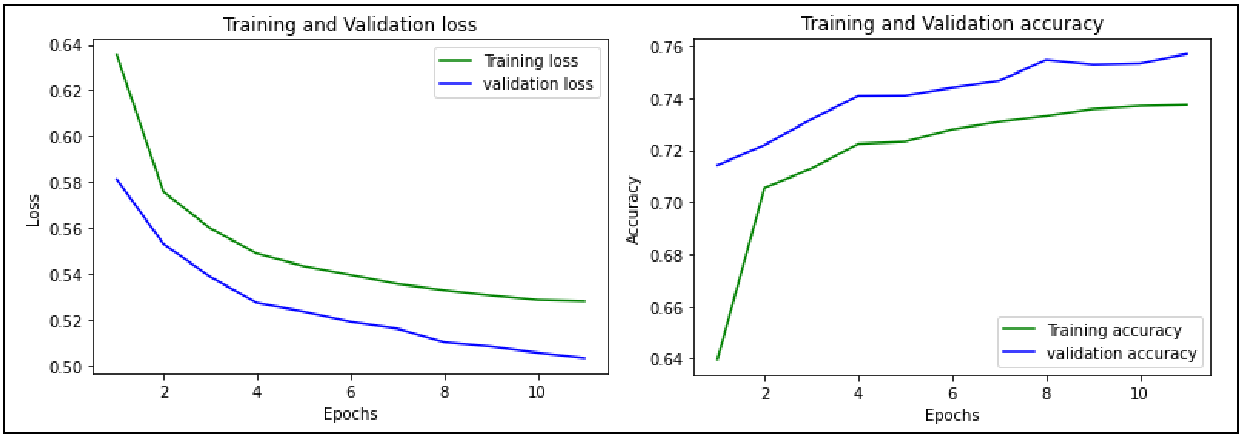

Figure 9. The accuracy and loss curve of the BERT model

Confusion Matrix: Figure 10 represents the confusion matrix of the BERT model, along with its graphical representation. It indicates that the model has performed well in predicting the True Positive reviews (3826) as compared to the True Negative reviews (3745). However, a number of false negative reviews were 1252, and number of false positive reviews was 1177. The model has performed balanced in terms of predicting the positive and negative reviews.

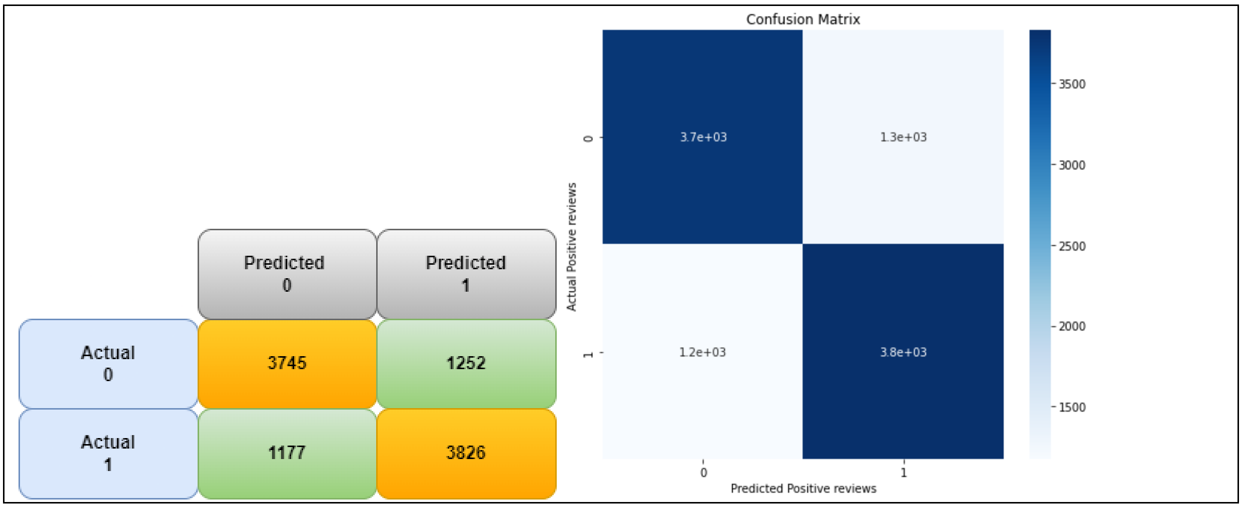

Figure 10. The Confusion matrix of the BERT model

6.3 USE

For the USE model the early stopping has been triggered at epoch 7, similar to the LSTM model. The loss of validation data is slightly higher as compared to training data and the accuracy of training data is slightly higher as compared to validation data. For validation data, the loss has been followed up & down pattern after the 4th epoch whereas, for training data, the loss has been decreased after the 2nd epoch. In terms of accuracy the training data has only an increasing pattern while, validation data has up and down, where the maximum accuracy of 84.49% has been achieved after 7 epochs. The early stopping has prevented the model from overfitting, therefore the up and down pattern on the validation set has been observed as seen in Figure 11.

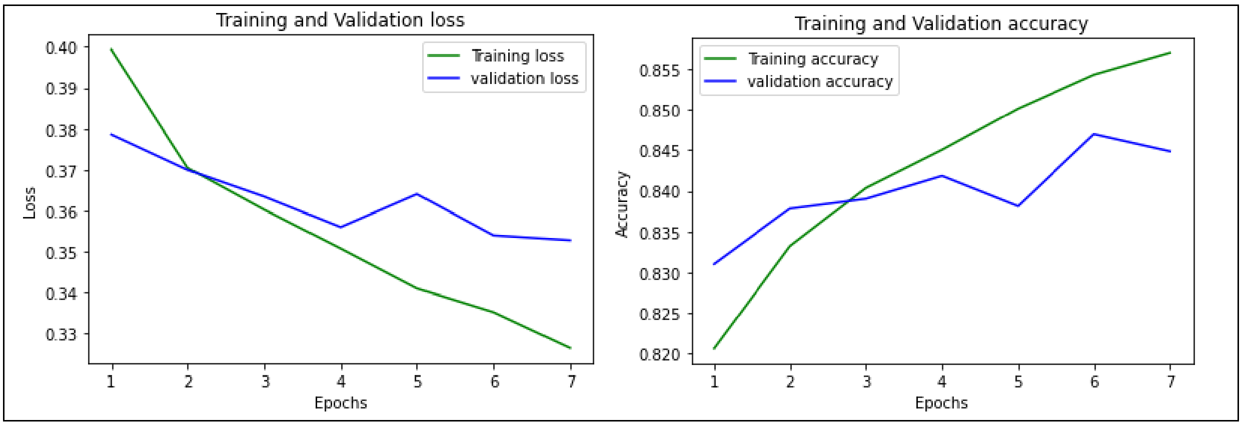

Figure 11. The accuracy and loss curve of the USE model

Confusion Matrix: Figure 12 describes the confusion matrix of the USE model. From the figure, it can be seen that model has performed exceptionally well since the number of false predictions is very less. The total number of true predictions was 8,469 and only 1531 reviews were false predictions.

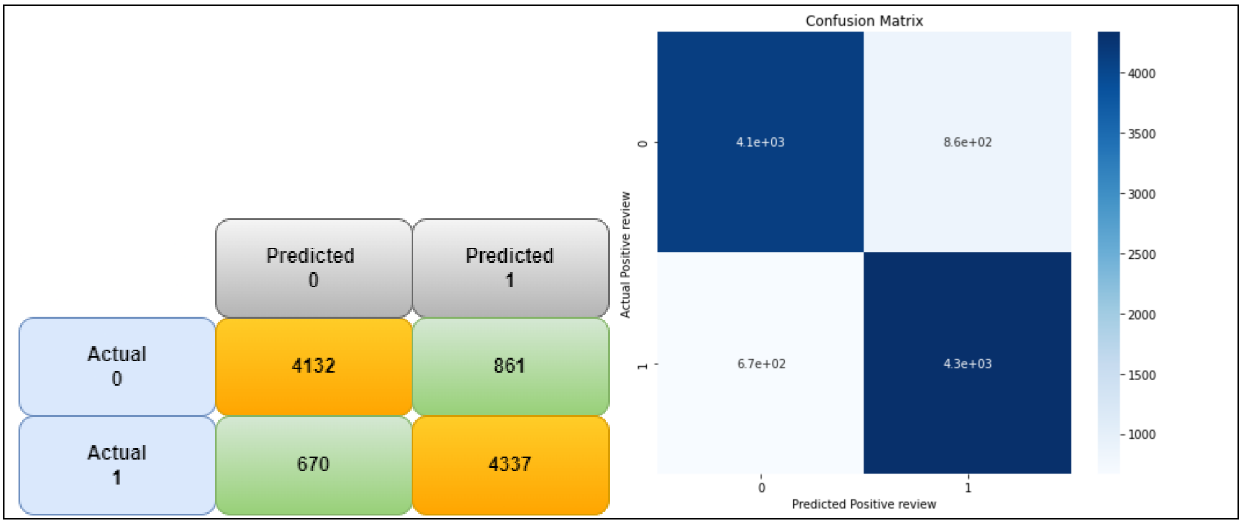

Figure 12. The Confusion matrix of the USE model

6.4 Model Comparison

Table 6 illustrates the total number of true and false predictions out of 10000 records considered for validation. It can be seen that the USE has performed exceptionally well as compared to LSTM & BERT in terms of predicting the true results. The BERT has more number of false predictions as compared to LSTM and USE.

Table 6. Confusion matrix comparison

Model	True Predictions	False Predictions
LSTM	8366	1634
BERT	7571	2429
USE	8469	1531

In any machine learning research, the loss must be evaluated along with the accuracy. The loss metric is very important for neural networks and hence it should be an objective function and need to monitor throughout the research. The loss is a penalty that the model incurs on its predictions and it tells how far the model is away from its ground truth table. Table 7 illustrates the maximum validation loss the model has incurred on its predictions along with the validation accuracy. It can be seen that the validation loss of BERT is 0.50 and which is more compared to LSTM and USE. The USE has achieved the highest accuracy of 84.49 as compared to the other two models.

Table 7. Loss & accuracy evaluation

	Epochs trained	loss	accuracy
LSTM	9	0.42	82.75
BERT	11	0.50	75.70
USE	7	0.35	84.49

Evaluation on the basis of Precision, Recall, and f1-score: The precision, recall, and f1-score are the important assessment matrix that has been chosen to evaluate the performance of the models in this research. Table 8 illustrates the comparison of Precision, recall, and f1-score of all three models.

Table 8. Precision, recall, f1-score comparison

Model	Prediction class	Precision	Recall	F1-score	Weighted Average
LSTM	Negative Reviews (0)	0.87	0.79	0.83	0.84
	Positive Reviews (1)	0.81	0.88	0.84	0.84
BERT	Negative Reviews (0)	0.76	0.75	0.76	0.76
	Positive Reviews (1)	0.75	0.76	0.76	0.76
USE	Negative Reviews (0)	0.86	0.83	0.84	0.85
	Positive Reviews (1)	0.83	0.87	0.85	0.85

The Precision defines the total number of actual positive reviews predicted correctly by the model as compared to a total number of predictions. The USE obtained the highest weighted average of Precision 0.85 as compared to LSTM & BERT.
The recall defines the total number of actual negative reviews predicted correctly by the model as compared to a total number of predictions. The USE obtained the highest weighted average of recall of 0.85 as compared to LSTM & BERT.
The f1-score is a weighted average of Precision and recall and in this research, all three models have predicted the balance distribution of classes. Since both positive and negative

reviews are important to analyze and the cost of both is equally important in terms of the domain.

6.5 Execution time

The execution time is another aspect on basis of which evaluation needs to be done. In this research, the evaluation has been done based on time required by epochs to get trained. Since a number of training and validation data has been taken equally the evaluation is possible. The average time required for epochs to get trained was 11 seconds for the LSTM model, whereas for the USE model the average time was 83 seconds per epoch. However, BERT was the most time-consuming and it took around 326 seconds for the epoch to get trained. Overall in terms of time aspect the LSTM and USE are more efficient.

6.6 Discussion

The primary purpose of this research was to incorporate the transfer learning approach in the sentiment analysis domain and thereby used it for movie review classification. From the literature review performed throughout the research, it has been observed that there is a research gap in this domain and the transformer-based models mainly Universal Sentence Encoder has not been prominently implemented in this domain. Along with USE, BERT and LSTM have been implemented to compare the performance of the models. To compare the performance precision, recall and f1-score have been chosen as a matrix with that loss has been evaluated. Pre-processing was very important in NLP tasks and hence in this research pre-processing was emphasized more. The data has been divided into train, validation & test with the help of Numpy, and the ratio of data splitting was decided to provide more data for training. To avoid the biases of the result equal number of data has been taken in the target classes. The early stopping and model checkpoint has been used to prevent the model from overfitting.

Results indicated that the transfer learning-based Universal sentence encoder has performed better in all aspects of evaluation. However, the performance of BERT was poor in comparison with USE and LSTM models. The various values of Hyperparameters have been tried and evaluated and the combination with the best results has been finalized to consider in the research.

The primary objective of the research was to check and evaluate the performance of transfer learning-based models and it has been implemented successfully. However, during implementation, various issues occurred that have been resolved during the implementation. One of the major hurdles was execution time and performance management. The Jupyter notebook failed to handle a large number of textual data hence Google colab Pro has been used as an IDE. During implementation, the few features of TensorFlow were not able to load and as a result, some unknown errors occurred. To resolve that, runtime and all environment variables have been cleared and all files have been executed again. The Stopwords removal process was time-consuming and took around 15 minutes to execute the code. Few untraced function warnings have occurred but they did not have any impact on execution and it has been confirmed by referring to documents provided by TensorFlow.

7 Conclusion and Future work

The results obtained, as discussed in the previous section highlight and support the use of the Universal sentence encoder-based transfer learning technique to classify the Positive & Negative reviews accurately, thus the objective of the research has been satisfied. However, the results obtained also indicated that transfer learning based on BERT has a wide scope of improvements as compared to RNN-based LSTM and USE. When the USE model has been tested on unseen testing data this model was capable enough to classify the movie reviews. The overall performance of BERT was concerning in all aspects of the evaluation. The USE model has performed well when evaluated on basis of Precision, recall, f1-score, loss & accuracy, except it took slightly more time to train the model. The LSTM model only performed well in terms of training time as compared to USE.

The research can be extended further by improvising the performance of transfer learning based on BERT and on a similar background the Roberta model can be tested. The performance of BERT can be further improved in terms of execution time by deploying it on cloud-based IDE which may provide more GPU and RAM. In this research epochs and batch size have been selected as Hyperparameters and the best combination has been finalized. However, along with epochs and batch size learning rate can be taken as a hyperparameter and tuning can be done. In this research, the web-based user interface has not been used to display the results that can be built to populate the results obtained. Another approach can be tested on this dataset in which for vectorization word2vec algorithm can be implemented.

8 Acknowledgement

The completion of this study could not have been possible without the expertise and constant help of Professor Hicham Rifai, my beloved dissertation advisor. Throughout the dissertation journey, he has created a very comfortable environment because of which I can able to clarify all the queries I had. I would like to thank all the faculty members for sharing the knowledge they have and which helped to complete the thesis. In the end, I would like to express my gratitude to my family members and friends for their unwavering support throughout this journey.

References

Asgari-Chenaghlu, M., Nikzad-Khasmakhi, N. and Minaee, S., 2020. Covid-transformer: Detecting trending topics on twitter using universal sentence encoder.

Chouikhi, H., Chniter, H. and Jarray, F., 2021, September. Arabic sentiment analysis using BERT model. In International Conference on Computational Collective Intelligence (pp. 621-632). Springer, Cham.

Dashtipour, K., Gogate, M., Adeel, A., Larijani, H. and Hussain, A., 2021. Sentiment analysis of persian movie reviews using deep learning. Entropy, 23(5), p.596.

Dey, S., Wasif, S., Tonmoy, D.S., Sultana, S., Sarkar, J. and Dey, M., 2020, February. A comparative study of support vector machine and Naive Bayes classifier for sentiment analysis on Amazon product reviews. In 2020 International Conference on Contemporary Computing and Applications (IC3A) (pp. 217-220). IEEE.

Fimoza, D., Amalia, A. and Harumy, T.H.F., 2021, November. Sentiment Analysis for Movie Review in Bahasa Indonesia Using BERT. In 2021 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA) (pp. 27-34). IEEE.

González-Carvajal, S. and Garrido-Merchán, E.C., 2020. Comparing BERT against traditional machine learning text classification. arXiv preprint arXiv:2005.13012.

Han, K.X., Chien, W., Chiu, C.C. and Cheng, Y.T., 2020. Application of support vector machine (SVM) in the sentiment analysis of twitter dataset. Applied Sciences, 10(3), p.1125.

Kurniasari, L. and Setyanto, A., 2020, February. Sentiment analysis using recurrent neural network. In Journal of Physics: Conference Series (Vol. 1471, No. 1, p. 012018). IOP Publishing.

Li, W., Zhu, L., Shi, Y., Guo, K. and Cambria, E., 2020. User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models. Applied Soft Computing, 94, p.106435.

Mengistie, T.T. and Kumar, D., 2021, April. Deep learning based sentiment analysis on COVID-19 public reviews. In 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) (pp. 444-449). IEEE.

Mitra, A., 2020. Sentiment analysis using machine learning approaches (Lexicon based on movie review dataset). Journal of Ubiquitous Computing and Communication Technologies (UCCT), 2(03), pp.145-152.

Mohammad, A.S., Hammad, M.M., Sa'ad, A., Saja, A.T. and Cambria, E., 2021. Gated Recurrent Unit with Multilingual Universal Sentence Encoder for Arabic Aspect-Based Sentiment Analysis. Knowledge-Based Systems, p.107540.

Murthy, G.S.N., Allu, S.R., Andhavarapu, B., Bagadi, M. and Belusonti, M., 2020. Text based sentiment analysis using LSTM. Int. J. Eng. Res. Tech. Res, 9(05).

Patel, D., Raval, P., Parikh, R. and Shastri, Y., 2020. Comparative Study of Machine Learning Models and BERT on SQuAD. arXiv preprint arXiv:2005.11313.

Patel, P., Patel, D. and Naik, C., 2021. Sentiment analysis on movie review using deep learning RNN method. In Intelligent data engineering and analytics (pp. 155-163). Springer, Singapore.

Poornima, A. and Priya, K.S., 2020, March. A comparative sentiment analysis of sentence embedding using machine learning techniques. In 2020 6th international conference on advanced computing and communication systems (ICACCS) (pp. 493-496). IEEE.

Pota, M., Ventura, M., Catelli, R. and Esposito, M., 2020. An effective BERT-based pipeline for Twitter sentiment analysis: A case study in Italian. Sensors, 21(1), p.133.

Qaisar, S.M., 2020, October. Sentiment analysis of IMDb movie reviews using long short-term memory. In 2020 2nd International Conference on Computer and Information Sciences (ICCIS) (pp. 1-4). IEEE.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

Villavicencio, C., Macrohon, J.J., Inbaraj, X.A., Jeng, J.H. and Hsieh, J.G., 2021. Twitter sentiment analysis towards covid-19 vaccines in the Philippines using naïve Bayes. Information, 12(5), p.204.

Zhao, J., Zeng, D., Xiao, Y., Che, L. and Wang, M., 2020. User personality prediction based on topic preference and sentiment analysis using LSTM model. Pattern Recognition Letters, 138, pp.397-402.