Exploring Deep Learning Models for Sentiment Analysis on Tesla News

MSc Research Project
Data Analytics

Rajat Patil
Student ID: x22162259

School of Computing
National College of Ireland

Supervisor: Shubham Subhnil

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Rajat Patil
Student ID:	x22162259
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Shubham Subhnil
Submission Due Date:	14/12/2023
Project Title:	Exploring Deep Learning Models for Sentiment Analysis on Tesla News
Word Count:	6003
Page Count:	20

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:	
Date:	29th January 2024

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):	

Exploring Deep Learning Models for Sentiment Analysis on Tesla News

Rajat Patil
x22162259

Abstract

Sentiment analysis has been a significant area of research in natural language processing, and selecting the appropriate model is crucial. Three major models—the Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN)—as well as experimentation with various grams will be used in this thesis as classifiers for the Lexicon Bert model (LeBERT). It was discovered that LSTM was exhibiting encouraging results for the Trigram and CNN for the Unigram and bigram after closely examining the model accuracy data. Overall, it was shown that the CNN model combined with the leBERT model as a classifier provided a flexible alternative with a wider range of gram configurations. For sentiment analysis tasks, the model's flexibility is essential. The results show that CNN works well for sentiment analysis. This thesis will offer significant new perspectives in the field of sentiment analysis.

Keywords: Sentiment analysis,NLP, LeBERT, LSTM, RNN, CNN, Unigram, Bigram, Trigram

1 Introduction

It is critical to comprehend people's opinions since they are the driving force behind purchasing and selling stocks for their investments, which influences many areas including the financial markets. If we understand the mood of investors, we can analyse stock price changes. Tesla, one of the most well-known corporations in the world, has grown successful in the market and has captured the attention of the media and the public, making Tesla the ideal firm for emotional research. The traditional approaches for sentimental analysis rely on a vocabulary that is related to three sorts of emotions: positive, negative, and neutral. Now, these techniques are unable to adequately assess the feelings of the text, which is a critical component of this research. The well-known BERT model developed by Google is a deep learning model that has dramatically transformed the NLP field; it stands for Bidirectional Representations from Transformers. When compared to the earlier lexicon-based technology, BERT has shown to be more important in comprehending the context of the text.

Sentiment analysis (SA) is a data analytics method for interpreting text data that might help us comprehend better business insights Kune et al. (2016). We may do sentiment analysis on a document, word, or sentence level, and it categorizes the material into multiple opinion classifications Yang et al. (2022). The text classifications are at the sentence level (SA), which is used for classifying brief text in identifying opinion groups. There are relatively few methods available that can do this emotional analysis successfully from news text data because the text is attributed to be an unstructured form of characters and sentences, making it exceedingly difficult to extract the specific qualities. According to Kalarani and Selva Brunda (2015), text representation occurs after the production of text data in SA. If we employ vector space models, or VSM, documents and phrases are translated into numerical vectors that reflect the text in this stage.

The translation of text to vector representation is the cornerstone of text categorization models Jiang et al. (2021). The effectiveness and accuracy of sentiment analysis are affected by whether or not the word vector is suggestive of the text Rao (2022); Onan and Korukoğlu (2017). According to the literature, two common methods for representing text vectors are automated vector representation approaches based on deep learning, such as word embeddingsSaleena et al. (2018); Ahuja et al. (2019); Rao et al. (2018) , and natural language processing (NLP) techniques based on part of speech (POS) tags, bag of words, and emotion lexicons Bhadane et al. (2015); Mozetič et al. (2016); Bin and Yuan (2012). In text classification problems, word embedding is one of the best deep learning approaches for constructing vector representations of words and documents.

One of the deep learning techniques for creating vector representations of words and documents in text classification problems is word embedding. Their capacity to grasp the syntactic and semantic relationships between words explains this Mutinda et al. (2021). Deep learning Word2Vec Rezaeinia et al. (2019), global vectors (Glove) Mikolov et al. (2013), FastText Pennington et al. (2014), and the bidirectional encoder representations from transformers (BERT) model Bojanowski et al. (2017) serve as the foundation for word embeddings models. These word embedding techniques need to be improved upon despite being far more successful than traditional NLP-based techniques Kenton and Toutanova (2019); Sharma et al. (2020). For example, a very big corpus is needed for efficient training and vector representation of words and word embeddings. Owing to these drawbacks, researchers—especially those working with small datasets—use pre-trained word embeddings for transfer learning, which may not align well with their data Liu et al. (2018) . Furthermore, the pre-trained word embeddings vectors ignore additional word features like semantic orientation and word context. Sentiment analysis models based on word embeddings can perform better when existing NLP methods are used, such as sentiment lexicon, POS tags, and word locations Mikolov et al. (2013).

This thesis explores the exciting field of financial sentiment analysis with a particular focus on news about Tesla, a trailblazing business in the renewable energy and electric car industries. The study presents a method that uses convolutional neural networks (CNN) and Lexicon-BERT, a natural language processing model, in recognition of the importance of sentiment analysis in stock market for decision-making. Web scraping is used in the first stage to compile an extensive dataset of news headlines about Tesla. My study attempts to capture complex feelings beyond conventional positive and negative categories by utilizing Lexicon-BERT. The model integrates lexicons and modifies weights based on news headlines and content to offer a more sophisticated understanding of the opinions expressed in the press.

The study investigates lexical tokenization using rule-based taggers and Treebank tagged sentences to improve sentiment analysis even further. To capture linguistic subtleties and enhance the feature space for sentiment analysis, unigrams, bigrams, and trigrams are constructed. After that, the thesis moves on to using Convolutional Neural Networks (CNN) on the tokenized lexical data. CNN architectures are used to process unigrams, bigrams, and trigrams separately to assess how well they predict sentiment. Accuracy metrics are used to evaluate each model's performance and provide a comparative study of their predicting skills.

Initial findings show encouraging results, with Lexicon-BERT offering a solid sentiment analysis base. Adding CNN models for bigrams, trigrams, and unigrams to the study improves it even more and shows how lexical tokenization may improve sentiment predictions. This work adds to the rapidly developing field of financial sentiment analysis by putting forth a solid methodology that combines lexical tokenization with sophisticated language processing techniques. With the help of this sophisticated instrument, investors and analysts should be able to assess market mood and make well-informed judgments in the ever-changing financial markets. RQ:Which deep learning models works best as classifier for sentimental analysis for different n-grams?

2 Related Work

2.1 Sentimental Analysis and its Impact

According to studies, there are more intricate connections between news mood, company networks, and market dynamics. This is because markets are known to be internationally interconnected and to be growing more so every day Wan et al. (2021) . The main objective of their paper was to examine how the mood of the news is spread across corporate networks and to evaluate its effects on market movements, specifically with stock prices and how volatile they can become. The researchers used natural language processing (NLP) tools to examine the sentiment of 87 firms that they covered extensively over a seven-year period. It was discovered that high media attitude toward a certain corporation will signal a substantial change about that company when it is employed in the financial markets. This problem was shown to be particularly common in specific sectors. Additionally, they discovered a statistically significant relationship between volatility and normal market outcomes. Although the association was more pronounced at the level of individual organizations, it is still visible at the broader levels of industries and grouping firms. The study draws attention to the wide-ranging effects that news emotion has on both smaller and bigger market segments. This greatly advances our knowledge of the global influence of media attitudes on financial markets, and it is possible to ascertain this through the use of natural language processing (NLP) technology.

By extracting significant information from the text, the researcher's study Giatsoglou et al. (2017) demonstrated the value of emotional analysis and opinion mining. The writers emphasized the challenges of automating sentiment retrieval, the complexity of the processes involved, and the linguistic variations in expressions. This work presents a technique to sentiment identification in textual snippets that is generic and flexible enough to represent people's opinions across several languages. They now propose training a polarity classification model by expressing texts as vectors and utilizing a machine learning framework. They also look at hybrid algorithm concepts and lexicon-based word embedding for document vectorization. To determine how effective these feature representations are, they conducted tests on four data sets that included online user evaluations in both Greek and English. One significant aspect of sentimental analysis is addressed in the judgment. One notable feature of the methodology is its minimal computational resource requirements, which opens up the prospect of applying some of the techniques in real-world scenarios where resource constraints may be a problem.

2.2 N-grams and Lexicon Based Techniques

One method to calculate the sematic score of sentences is to use SentiWordNet, which analyzes the pattern structure of sentences to extract contextual information. This study by Khan and Baharudin (2011) classifies opinions and presents a rule-based sentiment analysis technique that emphasizes on the sense of individual phrases and the understanding of information from the text. It is possible to classify opinions into three categories: positive, negative, and neutral by determining the semantic orientation of each phrase in the sentence and the semantic weight for each phrase. The importance of phrase structure and contextual information for the orientation and classification of sentiments. They suggest a rule-based sentiment analysis method that will give weight to word meaning and information extraction in order to categorize points of view. They discuss the significance of sentence form and context for categorizing and locating emotions. In order to achieve more effective semantic orientation, they have proposed that future research should focus on improving the extraction of acute senses from phrases while minimizing noisy material. As semantic score databases for all speech segments continue to grow.

Sentiment analysis is one of the most significant uses of word n-grams in natural language processing (NLP). Particularly when it comes to text classification. In order to create the vectors and word co-occurrence patterns required to train the machine learning classifiers, word-gram will always be crucial. According to study by Aisopos et al. (2016) , N-gram models are usually appreciated for their simplicity and effectiveness; nevertheless, they also have a disadvantage in that they are unable to accurately capture the delicate information that is required inside the word sequence. Negations and contextually dependent changes in word meaning are not handled by the traditional N-gram model.

Kumar et al.'s work Jain et al. (2022) uses bi and trigrams to extract features from text data and produces encouraging sentiment analysis findings, it should also be recognized as notable in the area. In their technique, fuzzy cognitive maps were employed as classifiers inside the entire big data analytics system. Our research will be furthered by examining the interplay between several natural language processing (NLP) techniques, such as sentiment lexicons, pre-trained word embedding, and N grams. My research aligns with the suggestions made by Kumar et al. for more investigations into deep learning architecture ,as order to integrate the best characteristics possible as a whole, I want to combine two things, such as sentiment lexicons and pre-trained word embedding with n grams. This will lead to a more sophisticated understanding of sentiment and aid to enhance the representation of textual data. By utilizing both traditional methods like N-grams and technological tools like sentiment lexicons and pre-trained embeddings, my work adds to the ongoing discourse in the field of sentiment analysis.

2.3 Deep learning and BERT

According to research by Kilimci et al. (2018) , Random Forest (RF) and Convolutional Neural Network (CNN) were found to be the best classifiers for text categorization; RF had higher accuracy than both. The study also showed how effective ensemble learning can be in enhancing classification accuracy on data sets with varying feature and profile distribution. Let's take note of the fact that heterogeneous has demonstrated sustained performance when compared to homogeneous equivalents. In the ensemble integration, stacking outperformed majority voting in terms of effectiveness and demonstrated the capacity to aggregate classifier decisions. Deep learning models, such as the CNN classifier, made significant progress in improving ensemble performance, showcasing the benefits of various classifier architectures. When utilized for document representation, trained word embeddings are the most effective feature reduction strategy that preserves classification accuracy. The text classification enhanced an innovative approach that fused word embeddings with additional characteristics (TF=IDF weighted vectors). The hybrid representation technique identified the improved performance of RF and CNN if you utilize them as individual classifiers and incorporated contextual, semantic, and syntactic information encoded in word embeddings for dependable text categorization. The findings indicated that it was crucial to think about utilizing several classifier and document representation strategies if you wanted the highest accuracy for your text classification assignment.

Wang et al. (2016) employed pretrained word2vec in their study, which demonstrated enhanced pre-trained word2Vec models for cross-domain classification, They utilized pre-trained word vectors in their attention-based long short-term memory (LSTM) models for aspect level emotional analysis. D'silva and Sharma[39] employed neural networks and FastText pre-trained word embeddings to classify Konkani literature. Use of mental traits with brief text. Although they did not compare their results to those of other word embedding models, they did execute it more effectively and provided more research on the application of BERT in other text categorization scenarios. In the study by Prottasha et al. [30], which contrasted werden2vec, Glove, FastText, and BERT, they emphasized the benefits of employing models like BERT in text representation and read word sequences in both directions and their impact on outcomes. The researchers hypothesize that by reducing the input series to a small number of words that share emotional meaning with their neighbors, BERT's performance can be enhanced. A sentiment lexicon and N-grams will be integrated into a hybrid N-gram feature extraction model-pinted lexicon. The lexicon-selected BERT embedding for sentiment classification will be presented in this research study. To sum up, word embedding is going to be employed by existing deep learning models for sentiment analysis, like BERT, to produce word vectors that are unique to the parts that have been detected. Sentiment analysis and natural language processing research has progressed to LeBERT's creative use of sentiment lexicon, N-grams, and BERT.

Recently, there has been a lot of interest in the field of NLP to comprehend the significance of word embedding for vector representation techniques D'silva and Sharma (2022). Word embedding feature selection research began to gain traction in 2013, according to Mikolov. Among the well-known word embedding methods for transforming words into vector forms were FastText, Glvoe, and word2Vec . Noting that the bidirectional representation from Transformers' BERT) model was receiving a lot of attention was due to its bidirectional and process-focused nature. As a result, models utilizing BERT embeddings were showing improved performance, particularly in emotive analysis tasks. Word embeddings offer advantages over the conventional representation of a bag of words since they allow for synonyms and smaller dimensional vectors. In a word embedding study conducted by garg, the investigator found that word2Vec embedding outperformed other methods. Pre-trained word embedding vectors are currently used by researchers as input for machine learning classifiers for sentimental analysis because of their improved performance and compatibility with deep learning models. However, pre-trained word embeddings ignore the sentiment orientation of words and their context, which affects the accuracy of sentiment categorization. The main reason for the limitation was that word vector computation relies on synonyms and word distances.

According to the researchers Mutinda et al. (2023), they used word N-grams and sentiment lexicon to implement the lexicon-BERT (LeBERT) model. The BERT model word embedding technique was used to vectorize the sections, capturing sentiment direction as well as semantic content. The CNN classifier, which was based on vectorized input for assessing LeBERT, was employed at the conclusion of the model for sentiment classification. On three distinct data sets, the research compares it against models such as Word2Vec, Glove, and BERT. They discovered that the input vector's dimensionality was much decreased by the usage of sentiment lexicon, which improved the model's sentiment analysis performance. Secondly, a more representative word vector was generated by integrating the BERT embedding technique with N-grams. Even though the problem was solved, the study notes that there were still some shortcomings in their model. For example, the fact that CNN was the sole deep neural network classifier suggested looking into other models, such as LSTM, for further research.In conclusion, the paper demonstrated the efficacy of sentiment analysis and discovered a feasible solution to the problems encountered.

3 Methodology

The architecture for this thesis, called the Integrated Pipeline for Sentiment Analysis, is broken down into sections, each of which is covered in detail below.

3.1 Data Collection

The gathering of data for analysis is the first and most crucial phase. I have extracted several headlines, excerpts, dates, and other associated information on the Tesla firm using web scraping from Google News. I'm making sure that the news is fully represented throughout the given timeframe of January 2023 to August 2023. Since Google is a trustworthy source of information on a wide range of subjects, gathering news items from there was helpful for the emotional analysis. The libraries needed to construct the model were downloaded.

3.2 Preprocessing

Sentiment Intensity Analyzer from the Natural Language Toolkit (NLTK) was utilized by me to deliver sentiment ratings to every headline to meet my data requirements. The

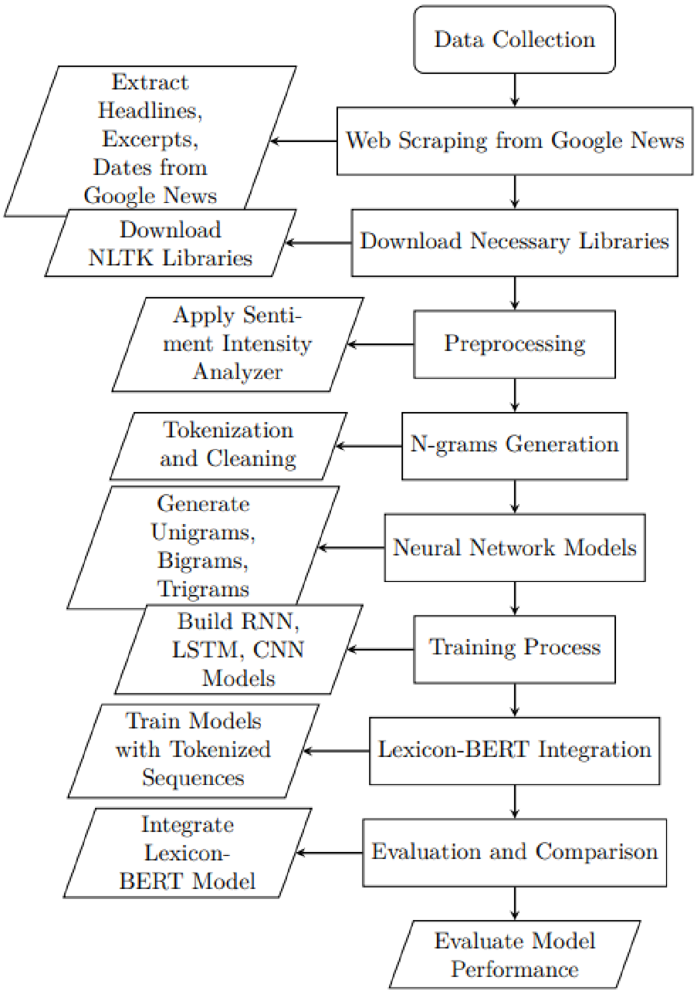

Figure 1: Methodology flow chart

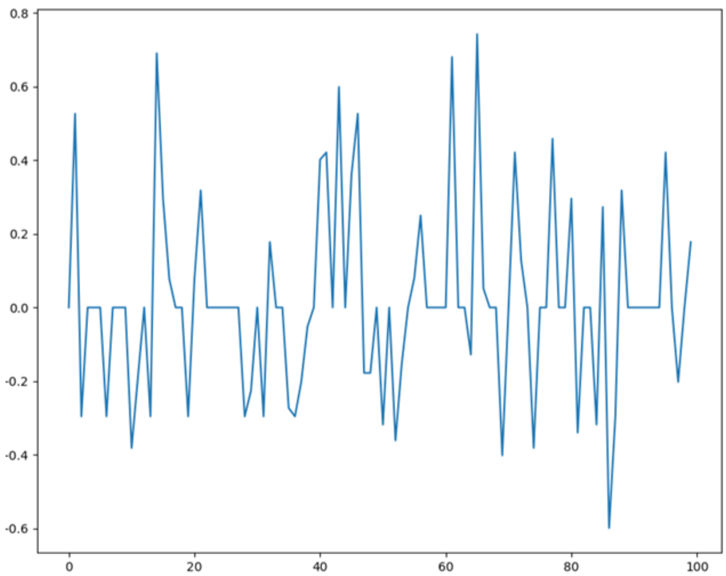

Figure 2: Sentiment Graph Distribution

score, which goes from -1 to 1, will be used to quantify the general mood. The analysis will be made simpler by the binary classification, which is based on a predetermined threshold. It will enable headlines to be clearly classified into positive and negative attitudes. Following the procedure, headlines were divided into three categories: unigrams, bigrams, and trigrams. Lexical tokenization, which is a crucial component of this thesis, was completed using NLTK's tokenizer and regular expressions. This phase will guarantee that the text data I have is appropriately cleaned before the model can utilize it, and it will tokenize the data to facilitate additional steps in cleaning and preparing it. The below graph shows

3.3 N-grams Generation

To improve the model's ability to extract contextual information from news headlines, the third phase of the code generates N-grams for the text. Unigrams are words that are represented as separate words and provide a basic knowledge. The other two grams—bigrams and trigrams—offer a more intricate perspective of the language by introducing higher order correlations between the words. We employ the more versatile produce ngrams function, which enhances the analysis's versatility.

3.4 Neural Network Models

Following the completion of all procedures, we begin applying the RNN, LSTM, and CNN models to increase the adaptability of neural network architecture for sentiment analysis. RNN is a good fit for this purpose since it excels at capturing sequential relationships within the language and includes recurrent connections, both of which are critical when considering context. The second model, LSTM, was selected to address the vanishing gradient issue and has established itself as a respected expert in the field. It is also capable of learning longer data sequences. utilizing CNN's hierarchical function to provide a different viewpoint on how to interpret the feelings. To guarantee the effectiveness and adaptability of the models mentioned above, we constructed them using the TensorFlow and Keras libraries in a Jupyter notebook. All of the aforementioned models include suitable activation functions, dropout layers, and loss functions that are designed specifically for problems involving binary classification. The goal was to create an architecture that would let models recognize and extrapolate patterns from tokenized and padded sequences.

3.5 Training Process

Training the models with tokenizes and padded sequences of three grams unigrams, bigrams, and trigrams is the fifth phase. Through this process, the models can pick up on trends, adjust, and correlate words with feelings. Dropout layers are essential for avoiding overfitting and guaranteeing that the models perform effectively when applied to unobserved data.

3.6 Lexicon-BERT Integration

The lexicon and BERT model together offer a special method for emotional analysis. BERT, a pretrained model, provides contextual comprehension of the words and phrases by considering their sentence-specific context. Unigrams, bigrams, and trigrams will all be tokenized into sequences using BERT's tokenizer, which will also perfectly match the input that the models need. The models' ability to comprehend sentiment more fully will be improved by the integration of sentiment lexicon.

3.7 Evaluation and Comparison

We must assess the models' performance to see which is working well. To do this, we utilize metrics like accuracy and loss, and we make sure the performance indicators give us information about the models' effectiveness. The contrast between Lexicon BERT and conventional lexical approaches provided a careful evaluation of the benefits and drawbacks of each strategy. Bar plots, which were used in the visualization process, helped to depict the model's performance clearly and more accurately.

4 Design Specification

When BERT assigns a vector, it takes into account the context of each word in the input text; this results in a high dimensionality vector. Second, the semantic information needed for sentiment categorization is absent from word vectors created with BERT. The sentiment lexicon, as opposed to BERT, may be used to recognize sentiment terms in a text and give them a particular emotion polarity. Sentiment lexicon, however, is unable to produce representative word vectors, which results in excessive data sparseness. Therefore, this study offers the LeBERT model, which integrates sentiment lexicon, N-grams, and BERT algorithms to improve sentiment classification.

The LeBERT model's design principle is to divide the input text into parts using N-grams first, and then to find the portion or sections that contain a sentiment word using a sentiment lexicon. It is important to keep in mind that text evaluations, like News headlines, typically comprise brief language, with semantic elements focused in one area of the text. Consequently, efficient, and effective text representation will result from feature extraction from such sections. BERT then turns the words of the selected section(s) into a vector. After that, features from the output word vector are produced by feeding it into a CNN /LSTM/RNN model that has a fully connected layer.

As discussed above we used classification layer at the end before getting the final output for our sentiments. Because of the various elements that come together to build a representation of the input text to the layers, we add an extra layer at the very end. The layer includes the necessary parameters, such as weights and biases, which are fine-tuned during the training of the data to align with the model's predictions. Ultimately, the layer makes the decision for the model, regardless of whether it is positive, negative, or neutral. The last layer of the model is connected to a loss function during training, which measures the difference between the sentiment that is actual and the sentiment that is predicted. The algorithm then modifies the entire network, including the classification layer, increasing the accuracy of the model.

5 Implementation

After discussing everything we move to the implementation of the models, I have used three models that is Convolution neural network (CNN), Long range short term (LSTM), recurrent neural networks (RNNs). With three different n grams 1,2 and 3 which is called as unigram, bigram, and trigram. I am going to start discussing about one model and its three grams at a time in the below description. Starting we use Hugging Face Transformers liabrary to create a tokenizer for BERT model by using bert-large-uncased variant which is known to be a pre-trained BERT model and have splitted the data in 80:20 where 80 for the training and 20 is for the testing, the first experiment is using unigram for all three models and do a comparison on which is performing better.

5.1 RNN Unigram

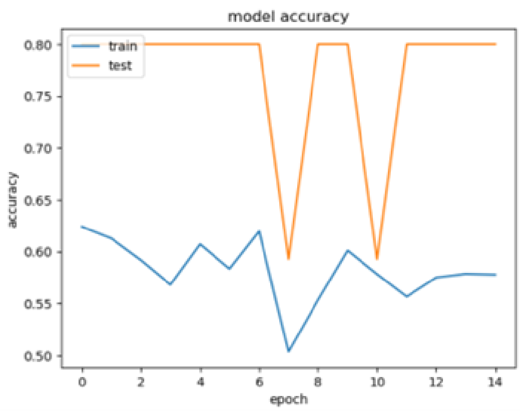

Figure 3: RNN Accuracy Unigram

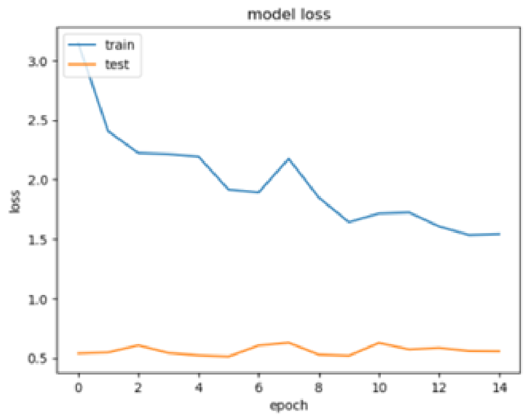

Figure 4: RNN Model Loss Unigram

The RNN model accuracy figure shown above shows the accuracy score of the model over various epochs. As the model learns to fit the training data, we can see that the training accuracy increases over time. The testing accuracy also increases over time, but it never surpasses the training accuracy because the testing data set contains data that the model has never seen before, making it harder for the model to predict.The model loss for the RNN model is depicted in the above picture as two lines: the red line represents the testing data set, and the blue line represents the training data set. The image illustrates how the model is attempting to learn over time for various epochs and testing loss. It is encouraging to observe that the difference between the training and testing is not too great.

5.2 LSTM Unigram

The next model is an LSTM classifier for lexicon BERT. From the model accuracy graph on the left, we can see that the line gradually increases to an accuracy score of 0.80 from its starting point of 0.65. This indicates that the model is correctly predicting the training data points from 0.65 to 0.80.Only after four epochs can the model reliably capture at least 75 percent of the testing data points.We can see that the train loss is dropping from 5.0 to about 3.0 on the model loss graph.The test loss line shows a progressive decrease around 4.5 after epoch 4, but it also shows a minor increase after epoch 6. This

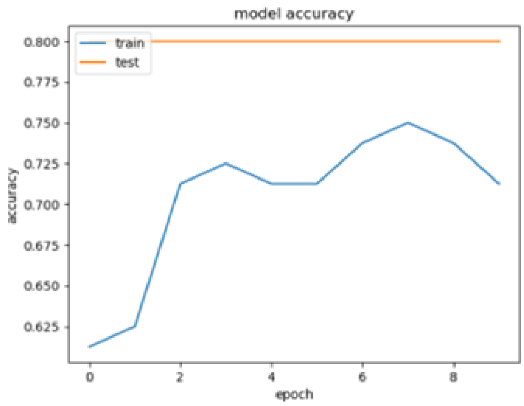

Figure 5: LSTM Accuracy Unigram

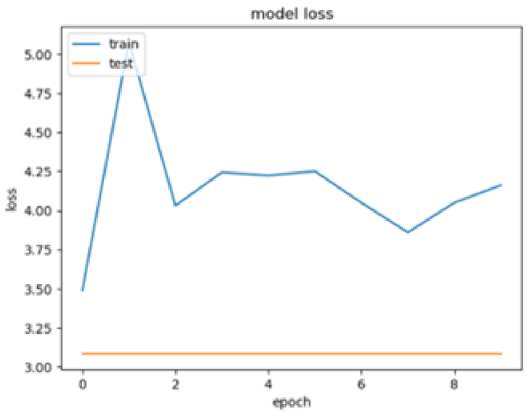

Figure 6: LSTM Model Loss Unigram

demonstrates that the model's loss on the test data is somewhat decreasing throughout the first four epochs before beginning to increase after the sixth.

5.3 CNN Unigram

The accuracy of random classifier is seen to be around 0.5, it is the point whre the train accuracy begins. Starting there for about 0.4 the test accuracy is seen to be somewhat to be lower than the train accuracy. The model loss for epoch can be seen decreasing for both train and test.

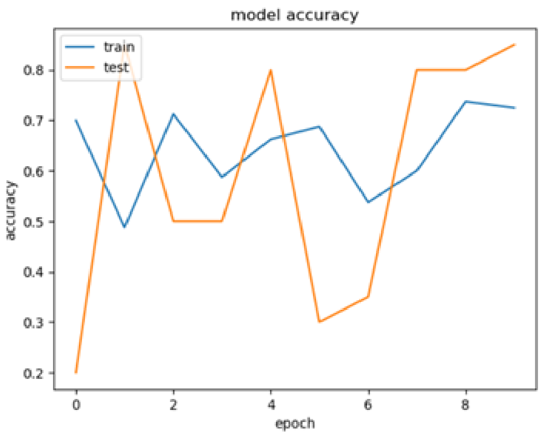

Figure 7: CNN Accuracy Unigram

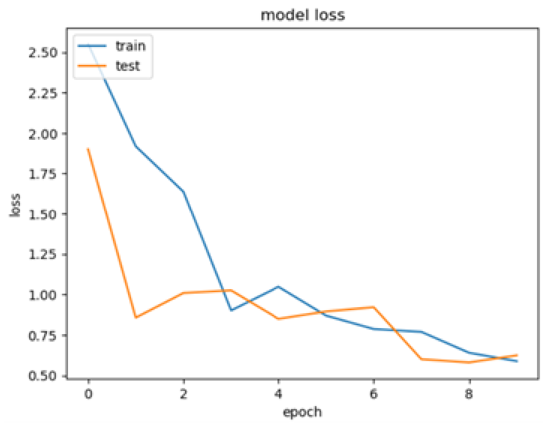

Figure 8: CNN Model Loss Unigram

The accuracy of random classifier is seen to be around 0.5, it is the point whre the train accuracy begins. Starting there for about 0.4 the test accuracy is seen to be somewhat to be lower than the train accuracy. The model loss for epoch can be seen decreasing for both train and test.

5.4 RNN Bigram

When compared to the testing data, we can see that the model is developing from 6.0 to 8.0, but it is not as effective for bigram as it is for unigram. This is because, according to the model analysis above, the training data set appears to grasp the data better after epoch 4.0 and then starts to deteriorate after epoch 8.0. On the model loss graph, both lines show a decrease in loss from epoch 2 to epoch 8, which is encouraging.

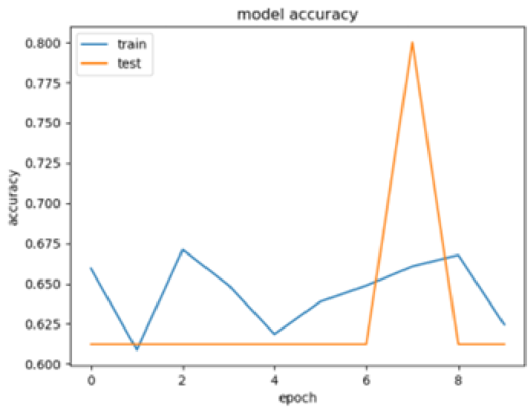

Figure 9: RNN Accuracy Bigram

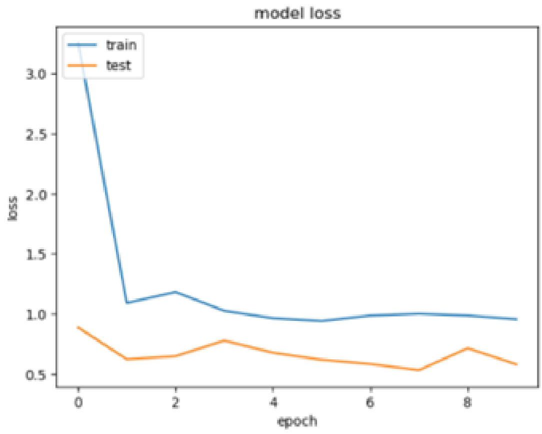

Figure 10: RNN Model Loss Bigram

5.5 LSTM Bigram

This model appears to have strong accuracy because the training accuracy constantly exceeds the test accuracy, and the difference between the two is not very great, indicating that the model is not overfitting the training set. The fact that the train and test accuracy curves are rising with time indicates that the model is becoming more intelligent. Even the test accuracy is above 50 percent, indicating that the model is generalizing well, and the training accuracy is above 50 percent, which is encouraging.

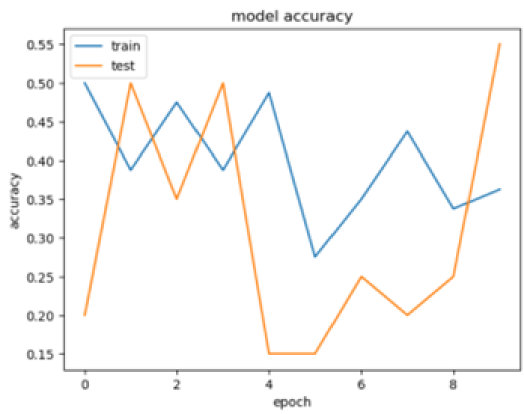

Figure 11: LSTM Accuracy Bigram

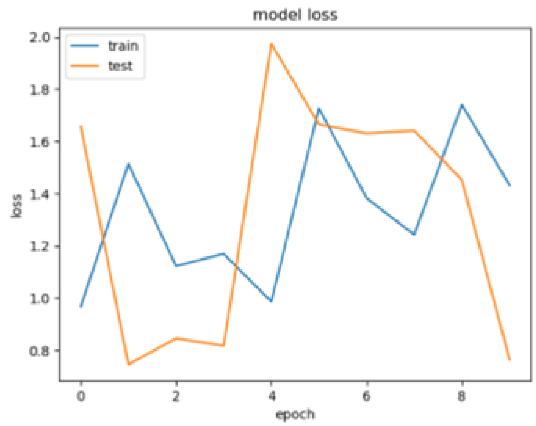

Figure 12: LSTM Model Loss Bigram

5.6 CNN Bigram

The CNN classifier for Bigram graphs demonstrate that the model is performing well in terms of accuracy as it learns during the early training stages. By the time the model reaches the final stage, its accuracy has surpassed 80 percent, and its loss has shown a good decline over the specified epochs, indicating that the model is able to capture a respectable amount of sentiment.

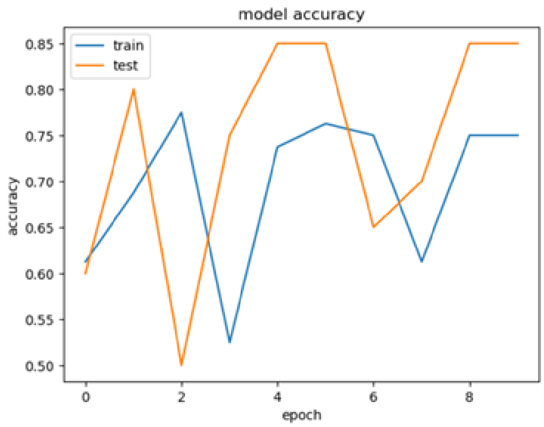

Figure 13: CNN Accuracy Bigram

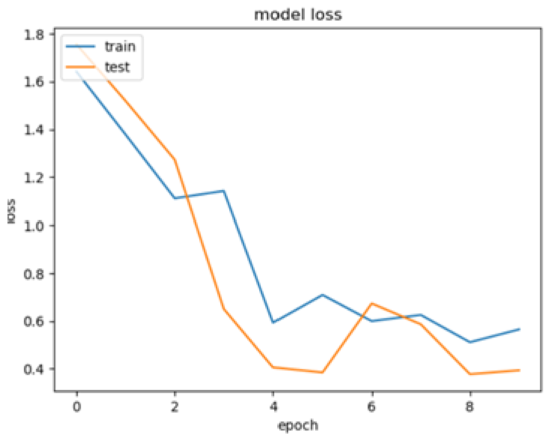

Figure 14: CNN Model Loss Bigram

5.7 RNN Trigram

We see that the RNN trigram model has high accuracy. This model's test accuracy is marginally lower than that of the train one, although this is to be expected given the occasionally difficult test dataset.Positively, the test accuracy is increasing over time as well. For every epoch, the test accuracy is 70 percent and there is little variation between the test and train accuracy. All things considered, the model is becoming better with time. It is a positive outcome.

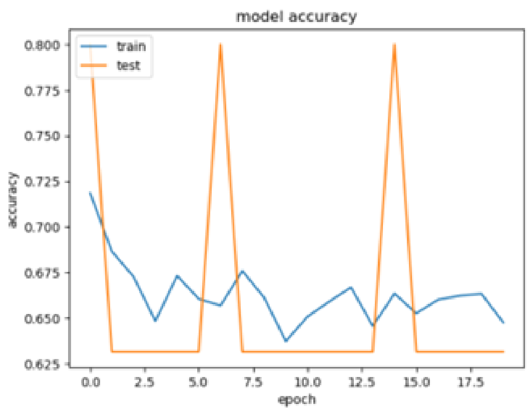

Figure 15: RNN Accuracy Trigram

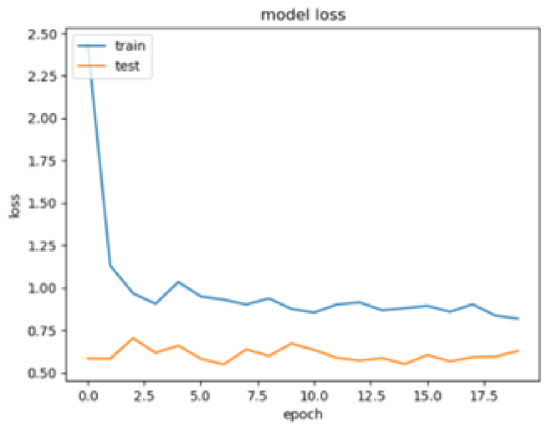

Figure 16: RNN Model Loss Trigram

5.8 LSTM Trigram

As can be seen from the above graphs, the test data set accuracy is also showing encouraging results from epoch 5.0 and above, with the difference between the two being minimal. The model accuracy graph on the left is showing strong results with accuracy score of 80 percent for most of the epochs. When we look at the model loss for this model, we can see that it starts to decrease at epoch 2.4 and even that the blue train line crosses the test line with the smallest gap, indicating that the model is doing an excellent job of generalizing to the new data. Additionally, the model loss is steadily declining.

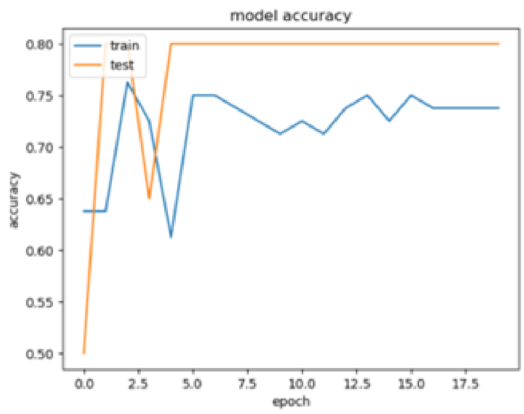

Figure 17: LSTM Accuracy Trigram

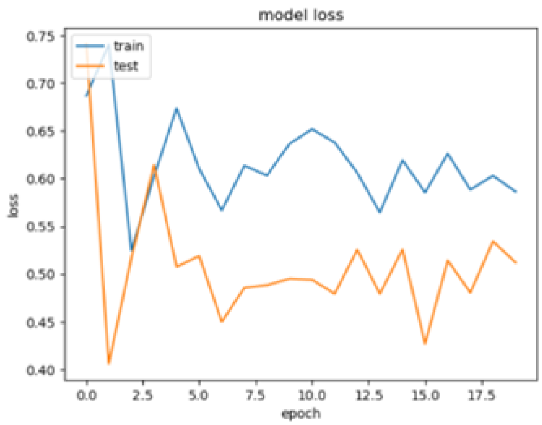

Figure 18: LSTM Model Loss Trigram

5.9 CNN Trigram

The CNN model is giving consistent results for all the grams with minor changes it is showing less accuracy score for trigram at the end of epoch 8. CNN model can be used for most of the grams even the model loss is decreasing significantly after epoch 3 and train and test lines come together at the end of the last epoch which is a good sign for the model.

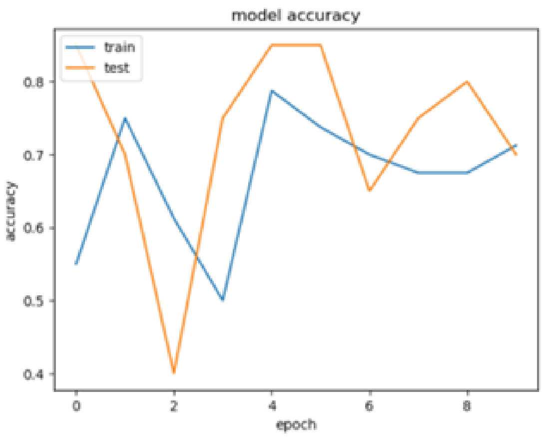

Figure 19: CNN Accuracy Trigram

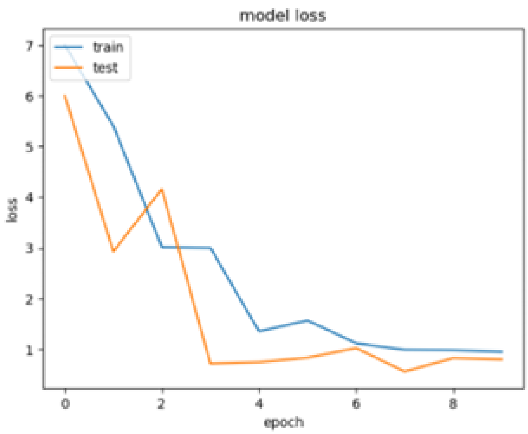

Figure 20: CNN Model Loss Trigram

6 Evaluation

The assessment is essential to the construction of any model. To view the performance of the models.In this part, we examine the efficiency of each classifier implanted with the lexicon BERT by utilizing accuracy as the evaluation parameter.One important metric for assessing the model's ability to perform appropriately across all categories is accuracy.Semantics is particularly significant in NLP tasks where accurate context identification is required. My goal is to clearly evaluate each classifier by limiting our evaluation metric to accuracy.We will see how each classifier is leveraging the rich contextual information in leBERT embeddings to make precise predictions as we proceed with the evaluation results. When combined with LeBERT to choose the best model for a given application, the analytical comparison will reveal the advantages and disadvantages of several classifiers.

6.1 Unigram

A unigram is a single word that serves as the fundamental analytical unit.Because it includes determining whether a sentiment indicated in a text is good, negative, or neutral, sentiment analysis is also known as opinion mining.Within the text, Unigram will stand in for individual words. For example, the unigram for "I Love Data Analytics" would be "I" "Love" "Data" "Analytics." Following the analysis of each word, the emotion of each unigram is averaged, resulting in the overall sentiment score. While certain words, like "love," often convey a positive attitude, others may be neutral or negative.

Table 1: Unigram Accuracy Table.

Models	**Grams**	**Accuracy in Percentage**
RNN	Unigram (N=1)	79.99
LSTM	Unigram (N=1)	80.00
CNN	Unigram (N=1)	85.00

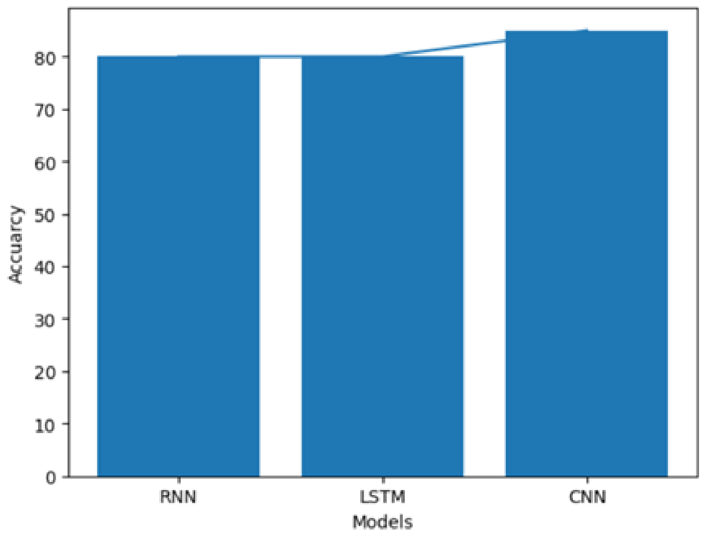

Figure 21: Unigram Accuracy Table

The three models' final performance scores for the Unigram experiment are displayed above. It is clear that all three models performed well, with the CNN model demonstrating the highest level of performance. This might be because of the comparable topologies of CNN and its ability to detect local patterns and attributes in the data. And there isn't much of a distinction between LSTM and RNN.

6.2 Bigram

Instead of analysing a single world as in the past or like a unigram, the model will analyse two words adjacent to each other and attempt to build the meaning. This is known as a bigram, or n=2, in emotional analysis. Occasionally, analysing two words together might reveal more meaning in the phrase than analysing just one. The bi gram comparison for each of the three models is shown below.

Table 2: Bigram Accuracy Table.

Models	**Grams**	**Accuracy in Percentage**
RNN	Bigram (N=2)	61.22
LSTM	Bigram (N=2)	55.00
CNN	Bigram (N=2)	75.00

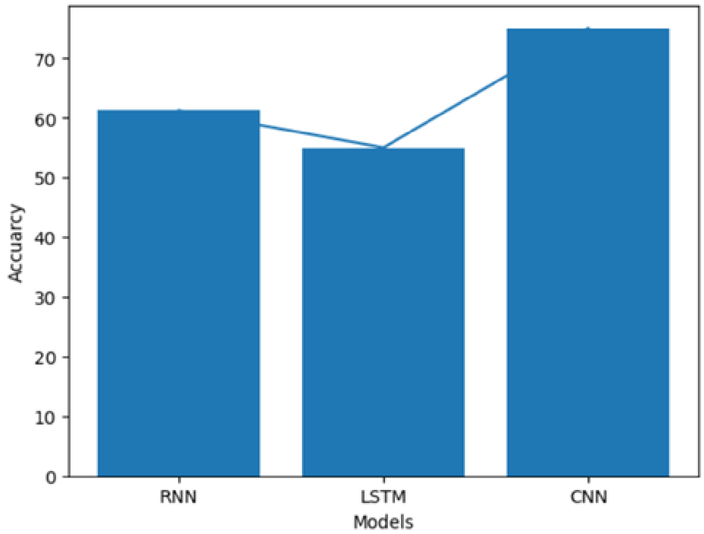

Figure 22: Bigram Accuracy Box Plot

Based on the final bigram results for the leBERT model using three models as classifiers, we can observe that the RNN and CNN models outperform the LSTM model, with the former scoring higher accuracy at 55 percent and the latter ranking second with over 60percent bigram accuracy. CNN, on the other hand, shows promising results at 75percent accuracy. This phenomenon may be explained by the fact that each model has an own architecture with comparable strengths and shortcomings. Although RNNs are designed to address the vanishing gradient problem, there are certain situations in which they may not work as well as they should. They can have trouble with long-range dependencies, even if they have shown to be rather successful in capturing sequential relationships. CNN has demonstrated a good ability to identify local patterns and an exceptional ability to extract features from bigrams. The model could show how successful the work is given how complicated it is. In the case of emotional analysis, for instance, CNN has proven to be more effective than RNN and LSTM as it primarily uses local text, such a word combination.

6.3 Trigram

A word sequence is referred to as a trigram. Trigrams are employed in text analysis and natural language processing (NLP) contexts as n-grams, where n is the number of words in the sequence.Compared to bigrams, a trigram can comprehend three words in a phrase and so offers more contextual comprehension. For illustration, let's look at a sentence. "Embrace the challenges, for they are the stepping stones on the path to your dreams." . The trigram will split phrases like "Embrace the challenges" and "for they are the stepping stones on the path to your dreams" in this statement.

Table 3: Trigram Accuracy Table.

Models	Grams	Accuracy in Percentage
RNN	Trigram (N=3)	63.16
LSTM	Trigram (N=3)	80.00
CNN	Trigram (N=3)	70.00

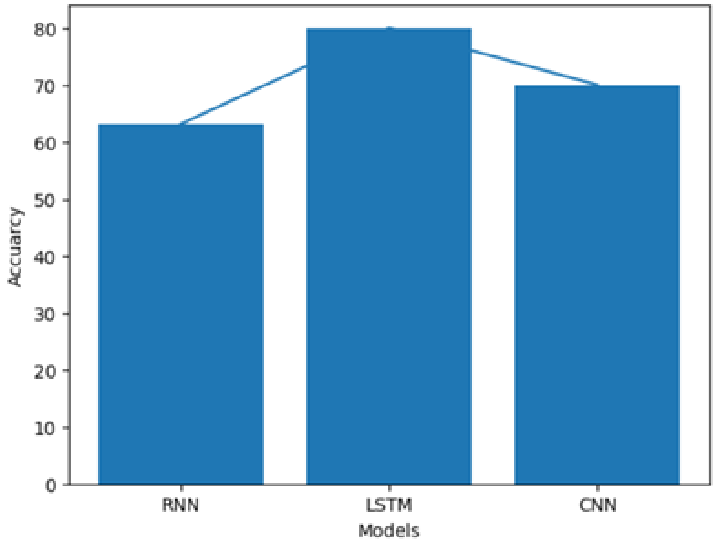

Figure 23: Trigram Accuracy Box plot

As we can see from the above graph and table, there are some discrepancies between the trigram and the other two models—the unigram and bigram. LSTM models perform best, scoring an accuracy score of 80 percent, followed by CNN with a score of 70 percent and RNN with a score of 63.16 percent This could be because long-range relationships in sequences are well-known to be captured by LSTMs, which is important for comprehension. Because of this, their accuracy is higher than that of other models. Longer sequences can be handled by LSTMs, which is useful for collecting more emotion patterns. The other two models may be having problems since they are not as capable of managing longer dependencies.

7 Conclusion and Future Work

In conclusion, the requirements and specifics of the sentiment analysis task determine the best option for an effective classifier model, whether it RNN, LSTM, or CNN. When the performance of the three models—unigram, bigram, and trigram—was compared, it was discovered that LSTM performed exceptionally well with trigram due to its capacity to capture long-range dependencies for nuanced sentiment, while CNN demonstrated encouraging results with unigram in terms of local pattern recognition. Most significantly, based on evaluation, we can state that CNN performs the best when taking into account a wider range of language patterns. This demonstrates the model's adaptability and efficiency across all grams, as demonstrated in the section on implementation. creating a robust model for tasks involving sentiment analysis when the total sentiment is influenced by both short- and long-range relationships. In the future, further research may be conducted on emotional analysis utilizing various deep learning models by averaging the strengths of several classifiers. Alternative BERT models can also be employed and their effectiveness examined with various data sets. The use of Lexicon BERT (LeBERT) for sentiment analysis on Tesla news items inside the financial markets is the main topic of this thesis. This work will be useful for future approaches and breakthroughs in natural language processing. The BERT mode can be applied across multiple domains with different datasets and larger datasets with greater computer resources. In summary, the model's capacity to comprehend challenging language and draw conclusions from it makes it the greatest option for further study in this area when it comes to sentiment analysis tasks.

8 Acknowledgment

First I would like to thank my supervisor Shubham Subhnil for supporting me and guiding me throughout this process and special thanks to God and my family for making me capable of this study.

References

Ahuja, R., Chug, A., Kohli, S., Gupta, S. and Ahuja, P. (2019). The impact of features extraction on the sentiment analysis, *Procedia Computer Science* **152**: 341–348.

Aisopos, F., Tzannetos, D., Violos, J. and Varvarigou, T. (2016). Using n-gram graphs for sentiment analysis: an extended study on twitter, *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, IEEE, pp. 44–51.

Bhadane, C., Dalal, H. and Doshi, H. (2015). Sentiment analysis: Measuring opinions, *Procedia Computer Science* **45**: 808–814.

Bin, L. and Yuan, G. (2012). Improvement of tf-idf algorithm based on hadoop framework, *2012 International Conference on Computer Application and System Modeling*, Atlantis Press, pp. 391–393.

Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T. (2017). Enriching word vectors with subword information, *Transactions of the association for computational linguistics* **5**: 135–146.

D'silva, J. and Sharma, U. (2022). Automatic text summarization of konkani texts using pre-trained word embeddings and deep learning, *International Journal of Electrical and Computer Engineering* **12**(2): 1990.

Giatsoglou, M., Vozalis, M. G., Diamantaras, K., Vakali, A., Sarigiannidis, G. and Chatzisavvas, K. C. (2017). Sentiment analysis leveraging emotions and word embeddings, *Expert Systems with Applications* **69**: 214–224.

Jain, D. K., Boyapati, P., Venkatesh, J. and Prakash, M. (2022). An intelligent cognitive-inspired computing with big data analytics framework for sentiment analysis and classification, *Information Processing & Management* **59**(1): 102758.

Jiang, Z., Gao, B., He, Y., Han, Y., Doyle, P. and Zhu, Q. (2021). Text classification using novel term weighting scheme-based improved tf-idf for internet media reports, *Mathematical Problems in Engineering* **2021**: 1–30.

Kalarani, P. and Selva Brunda, S. (2015). An overview on research challenges in opinion mining and sentiment analysis, *Int. J. Innov. Res. Comput. Commun. Eng* **3**(10): 1–6.

Kenton, J. D. M.-W. C. and Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding, *Proceedings of naacL-HLT*, Vol. 1, p. 2.

Khan, A. and Baharudin, B. (2011). Sentiment classification using sentence-level semantic orientation of opinion terms from blogs, *2011 National Postgraduate Conference*, IEEE, pp. 1–7.

Kilimci, Z. H., Akyokus, S. et al. (2018). Deep learning-and word embedding-based heterogeneous classifier ensembles for text classification, *Complexity* **2018**.

Kune, R., Konugurthi, P., Agarwal, A., Rao, C. R. and Buyya, R. (2016). The anatomy of big data computing, *Softw., Pract. Exper.* **46**(1): 79–105.

Liu, Y., Liu, B., Shan, L. and Wang, X. (2018). Modelling context with neural networks for recommending idioms in essay writing, *Neurocomputing* **275**: 2287–2293.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781* .

Mozetič, I., Grčar, M. and Smailović, J. (2016). Multilingual twitter sentiment classification: The role of human annotators, *PloS one* **11**(5): e0155036.

Mutinda, J., Mwangi, W. and Okeyo, G. (2021). Lexicon-pointed hybrid n-gram features extraction model (lenfem) for sentence level sentiment analysis, *Engineering Reports* **3**(8): e12374.

Mutinda, J., Mwangi, W. and Okeyo, G. (2023). Sentiment analysis of text reviews using lexicon-enhanced bert embedding (lebert) model with convolutional neural network, *Applied Sciences* **13**(3): 1445.

Onan, A. and Korukoğlu, S. (2017). A feature selection model based on genetic rank aggregation for text sentiment classification, *Journal of Information Science* **43**(1): 25–38.

Pennington, J., Socher, R. and Manning, C. D. (2014). Glove: Global vectors for word representation, *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Rao, G., Huang, W., Feng, Z. and Cong, Q. (2018). Lstm with sentence representations for document-level sentiment classification, *Neurocomputing* **308**: 49–57.

Rao, L. (2022). Sentiment analysis of english text with multilevel features, *Scientific Programming* **2022**: 1–10.

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

Saleena, N. et al. (2018). An ensemble classification system for twitter sentiment analysis, *Procedia computer science* **132**: 937–946.

Sharma, A. K., Chaurasia, S. and Srivastava, D. K. (2020). Sentimental short sentences classification by using cnn deep learning model with fine tuned word2vec, *Procedia Computer Science* **167**: 1139–1147.

Wan, X., Yang, J., Marinov, S., Calliess, J.-P., Zohren, S. and Dong, X. (2021). Sentiment correlation in financial news networks and associated market movements, *Scientific reports* **11**(1): 3062.

Wang, Y., Huang, M., Zhu, X. and Zhao, L. (2016). Attention-based lstm for aspect-level sentiment classification, *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 606–615.

Yang, J., Xiu, P., Sun, L., Ying, L. and Muthu, B. (2022). Social media data analytics for business decision making system to competitive analysis, *Information Processing & Management* **59**(1): 102751.