Comparison of four neural network based hybrid methodologies for sentiment and emotion analysis

1st Ramaswamy Iyer
Student of Masters in Data Analytics
National College of Ireland
Dublin, Ireland
x18183239@student.ncirl.ie

2nd Sankara Subramanian Venkatraman
Student of Masters in Data Analytics
National College of Ireland
Dublin, Ireland
x18179541@student.ncirl.ie

3rd Iswaria Nagarajan
Student of Masters in Data Analytics
National College of Ireland
Dublin, Ireland
18183379@student.ncirl.ie

4th Shreya Merkaje Ravi
Student of Masters in Data Analytics
National College of Ireland
Dublin, Ireland
x18190910@student.ncirl.ie

***Abstract*—Twitter tweets plays an important role in every organisation. This project is based on analysing the English tweets and categorizing the tweets based on sentiment and emotions of the user. The literature survey conducted showed promising results of using hybrid methodologies for sentiment and emotion analysis. Four different hybrid methodologies have been used for analysing the tweets belonging to various categories. A combination of classification and regression approaches using different deep learning models such as Bidirectional LSTM, LSTM and Convolutional neural network (CNN) are implemented to perform sentiment and behaviour analysis of the tweets. A novel approach of combining Vader and NRC lexicon is used to generate the sentiment and emotion polarity and categories. The evaluation metrics such as accuracy, mean absolute error and mean square error are used to test the performance of the model. The business use cases for the models applied here can be to understand the opinion of customers towards their business to improve their service. Contradictory to the suggestions of Google's S/W ratio method, LSTM models performed better than using CNN models for categorical as well as regression problems.**

***Index Terms*—Twitter, Emotion, Sentiment, LSTM, CNN, Bidirectional LSTM, Classification, Regression, Polarity, Hybrid, Google S/W**

I. Introduction

In the 21st Century, Social Media is the popular platform to understand people sentiment on a social or political issue, user experience on a product and medium of communication. This increase the size of web data and usage of the internet. Micro-blogging sites have become a mode of communication and social life. Twitter alone has 140 million active users daily, and this has made sentiment analysis and opinion mining one of the hot topics for research. The main challenge for Web data mining is ameliorating size of unstructured data and short text. Web Mining has Web Content, Structure and Usage Mining. In this study, we will focus on Web Content Mining to discover valuable insights from vast web content [1].

Sentiment Analysis and Opinion Mining has carried out using various text mining methods such as word embedding, Tf-IDF vectorization, lexicon and skip-grams in [2], [3], [4] and [5]. Similarly, twitter political opinion mining [6] has using deep learning and NRC lexicon has predicted who will triumph 2017 Gujarat State Election based on the emotion of the tweets. In addition to these mining methods, various Machine learning (ML) and Deep Learning (DL) models were applied to draw insights from the tweets, comments and reviews. Research [7] has applied Random Forest, Naïve Bayes and Sequential Mining Optimization (SMO) for opinion mining. Based on the review of a movie its sentiment has classified using Long-Short Term Memory (LSTM) DL model [8]. Its performance was compared against RNN and CNN models.

This research is focused exclusively on text analytics using various DL models and Hybrid DL models. The research is divided into 3 phase. In the first phase, data extraction, pre-processing and data-preparation are implemented. Secondly, different DL models and several word embedding techniques have experimented. Finally, the models are estimated using a variety of evaluation metrics. The major challenge faced is data extraction and pre-processing. The data available for a single day is about 1 00 000 million, after adding filters on language and retweets, data reduced to 41 000. Also, this research is restricted to the English language.

In this first phase, extraction is handled by Apache Spark from a huge volume of data. The extracted data is pre-processed using Natural Language Programming (NLP) techniques of tokenization, Lemmatization, Stopwords Removal, Stemming and Emoji Removal. Secondly, for the analysis 3 Classification and 1 Regression models are applied. Emotional category classification is carried using LSTM with NRC Lexicon and hybrid LSTM-CNN with Vader lexicons. Sentiment analysis is carried using Bidirectional-LSTM with Vader sentiment which is a regression model and LSTM-CNN with Vader lexicons a classification model. Finally, models are evaluated using metrics of Accuracy and confusion matrix for

classification, Mean Square Error (MSE) and Mean Absolute Error (MAE) for regression.

In the upcoming sections, a brief background on various NLP techniques, word embeddings, lexicons, ML and DL models are discussed. Methodology, Evaluation, Conclusion and Future Work conducted in this research are also explained in detail.

II. Related Work

Sentiment analysis is the interpretation of emotions and sentiments using natural language processing and text analysis. It also helps to understand public opinions, experience, and product influence which in turn is beneficial for companies to plan marketing strategies and product features. Multiple attempts have been made in this area and few of them have been outlined below.

In [2] Convolution Neural Network (CNN) for used for classifying sentiment in tweets. The goal of this experiment was to classify the tweets into two categories: positive and negative. The proposed model used word embedding, n-grams, and lexicon for the identification of sentiments. The pre-processing of data involved removing URLs, emoticons, and stop words. The proposed model showed an accuracy of 87% which was higher than the baseline model. The model can be improved by maximizing the number of classification categories.

A study on the hybrid model was proposed that made use of both text mining and neural networks. This hybrid model was also tested against different techniques such as decision trees, Naïve Bayes, and Convolution Neural Networks (CNN). For training and testing purposes the dataset was divided into the ratio 0.75 and 0.25. Accuracy of 83% was achieved for a hybrid model. It was also tested against different techniques such as decision trees, Naïve Bayes, and Convolution Neural Networks (CNN) [9].

Research by [3] has attempted at evaluating opinions of tweets such as positive, negative, and neutral and build a sentiment classifier. The tweets have extracted in the form of hashtags and then pre-processed where each word was tokenized. Stop words have removed and stemming has applied to classify similar words under one category. Bag of words was used for comparing and classifying tweets. In this scenario, using deep learning methodologies has improved the efficiency of the system.

Interpreting emotions by capturing tweets is a complex task due to several reasons. Multiple languages in a single tweet, using short forms is a common practice and domain-specific words. Considering all these shortcomings a study on software users' tweets was conducted to classify and interpret emotions [10]. The normal sentiment/emotion analysis was based on words like love, hate, good and bad. Words like a crash, slow, fix, etc. play an important role in tweets related to software. For the classification, Naive Bayes and Support Vector Machines were used and both these were successful in classifying the emotions with decent accuracy.

In [11] a model was proposed to classify emotional states of individuals into 8 categories. Algorithms such as Support Vector Machines (SVM), K-Nearest Neighbour, Decision Tree, and Naive Bayes were implemented. The pre-processing of data involved removing URLs, symbols, and inappropriate words. Among all the applied algorithms, SVM had the maximum accuracy. The model is well suited for suggestion systems where the prediction of emotion is important. In this study, emphasis on pre-processing has made to improve the model's behaviour.

The research conducted in [4] uses data from twitter with pre-assigned sentiments to create a model that can predict the sentiments. By appling DNN, CNN and RNN to perform a comparison between them by using mixed data. The mixed data provides a variety of information and is not limited to a specific demographic of text data. The use of TFIDF vectorization hasn't yield stable accuracy across the models, but word embedding overcomes this issue. TFIDF vectorizes the text as per count whereas word embedding assigns a specific unique value to each word irrespective of its count. The better performance of word embeddings may be due to the reason that a Neural Network is a sequential model and takes into consideration the previous input as well. The best results were obtained using a combination of word embeddings and RNN which provided an overall accuracy of 85 to 90 percentage.

The research conducted in [12] has used a hybrid approach to perform sentiment analysis at the sentence level. The researchers combined the use of lexicon and fuzzy sets on data collected from twitter to generate a new lexicon that predicts the sentiment of the text. By combining the results of pre-trained lexicon containing polarity scores and SentiWordNet sentiments were determined. It also included the use of Semantic rules which decides which part of the sentence to consider to generate the overall sentiment. By comparing their results with Naïve Bayes and Maximum entropy classifier, the hybrid approach performed better at predicting the sentiment with an accuracy of approximately 85%. Also, an increase of 20% compared to the machine learning approaches.

The researches have used a hybrid approach of using SentiWordNet and SVM on twitter tweets to generate a sentiment classifier. An approach of handling the negation using local contextual semantics in the tweet, the research was able to consider the neighbouring words and capture the semantics of the word in question [13]. The imbalance of classes within the data was tackled by assigning weights to these classes as per their frequency has improved the model. Using, the negation handling of tweets the sentiment lexicon classification might have worked better. But it has provided only an accuracy of 58% which seems to be low considering that the confidence in the model is only slightly better than having a 50-50 probability of the sentiment categories.

The research conducted in [14] has used a combined approach of word embeddings (Word2Vec, FastText, character-level embedding) coupled with deep learning methods to create a robust sentiment classifier. The proposed architecture of

the research involves the use of character-level embedding with CNN feature extraction. By using this hybrid approach of CNN and Bi-LSTM along with Character and FastText embeddings, the model provided an accuracy of 82% on the training data and proved to be better than using CNN and Bi-LSTM separately.

Sentiment analysis of short text using deep learning approach [15] uses Conv-LSTM. It is a combination of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). The usage of LSTM instead of pooling layer as reduces the loss of information from the previous layer. The major advantage of deep learning model over Machine learning models is Feature Extraction and Classification conducted in a single step. Unlike RNN, LSTM is not a biased model and enhances complexity. LSTM has 4 layers of input, output, forget and the candidate memory cell. Forget decides which information to retain for the next stage. The data processing has carried out using word2vector embedding method which implements skip-grams and Bag-of-Words (BOW) representation of words.

Similarly, the classification of the author's gender and text sentiment using deep learning models carried by [16] has used various (BOW) word2vector corpora such as Word2vec ruscorpora, Word2vec web and Word2vec twitter. Word2vec ruscorpora contains 107 561 399 tokens of Russian BOW. Word2vec web has 9 million documents with 660 628 738 tokens use Russian web crawled texts. Finally, it has utilized Word2vec twitter data which trains using a continuous skip-gram algorithm. Feature extraction of morphological and syntactical characters which extracts Part Of Speech (POS) tags and syntactical relations has removed. Both the model has used Softmax as activation function, Adam optimizer and each layer consist of 200 neurons. K- fold cross-validation of k - 1000 has used for splitting the train-test dataset.

Sentiment Analysis on GPU using DL models [17] has used airline tweets with CPU and GPU processing units. GPU improves parallel and distributed computing whereas CPU is serial. Generally, GPU is used for mathematical operations such as matrix multiplication. Data Parallelism (DP) and Model Parallelism (MP) can be used for scaling NN. But for big dataset DP is not suitable, MP is possible for handling big scale NN model. Associated Press News (APNews) word2vec has utilized for word embedding. It has concluded that GPU process the 30s/epoch less than CPU, stemming is not effective in this analysis and the pre-processing has a major contribution to improve accuracy.

Research on Sentiment Analysis of Google Play consumer review on the Chinese language [5] uses LSTM-RNN. The research has carried out in 3 phases and various DL and ML models are applied. In the second step, iSGoPaSD a data dictionary on the Chinese language which acts as BOW has created that integrates HowNet and NTUSD. Finally, the performance of average sampling and non-average sampling performance has analyzed. This research helps the users to understand the product quality and how the polarity of user sentiment affects the product. The similarity of the reviews has calculated using cosine distance of Term Frequency Inverse Document Frequency (Tf-IDF) and also repetitive words were removed. Hidden Markov Model (HMM) segments the Chinese characters fed into LSTM layers. The accuracy has increased by adding new words to the dictionary.

Research [18] talks about how an organisation or market place will be successful by using twitter to predict customer perspectives. An approach is used to measure the customer's perspective by developing a program using sentiment analysis. By the implementation of the prototype, a large amount of tweets has been extracted to develop the design for sentiment analysis. A pie chart and HTML page has been used to represent the customer's perspective results through positive, negative and neutral tweets. This work has been implemented using Natural Language Processing (NLP) with the help of support vector machine (SVM) as a classifier to classify the tweets.

Classification of tweets using machine learning techniques on trained and test data has implemented [19]. Identification and classification of tweets expressed in a source text have done using various machine learning techniques like Naïve Bayes (NB), Maximum Entropy (ME) and Support Vector Machine (SVM). Due to the presence of slang words and misspellings, analysis of twitter sentiments is quite hard when compared to general sentiments. Machine learning approaches have been used to analyse the tweets of electronic products like mobile phones, laptops, etc. This approach is successful in calculating the accuracy of the new feature vector and also machine learning techniques are easy in identifying the tweets when compared to symbolic techniques.

In [20] sentiment analysis of tweets was done using deep learning methods. By avoiding the additional features, Convolutional Neural Network (CNN) has been used to train an accurate model. Initial word embeddings have been trained using an unsupervised neural language model. Also, the deep learning model was used to train supervised corpus. By the comparison of results, it was found that deep learning model performed better than the machine learning model.

Interesting research [21] in which stock market movements were predicted have used public sentiments and current event opinions were easily represented in social media platforms. For research purposes, twitter attracts more attention to analysing public sentiments. This work found out how the tweets have been influenced by the rise and fall of the stock market company. For analysing the public tweets two textual representations such as N-gram and Word2vec were used. Random Forest was used to training the tweets. This work represents a strong correlation between the stock markets and public opinions.

As per the literature survey conducted, the sentiment and emotion analysis of text data can be divided into three approaches based on the methods applied, Lexicon based approach, Machine Learning based approach and a Hybrid approach [22]. Lexicon based approach involves the use of only Lexicons or dictionaries to assign the sentiment or emotion value behind the text. It also considers the average of these

values to assign the category of sentiment or emotion [22]. The inherent limitations of the lexicon and machine learning based approaches is that the lexicon based approach needs the data to be manually processed and consumes more time than the machine learning based approach whereas the machine learning based model requires pre classified data using human interaction to train the model. The use of hybrid approach helps to overcome these limitations. A hybrid approach makes use of a lexicon to automatically assign the sentiment or emotion behind the text and training a machine learning model to predict the same [22]. This drove the decision behind using a hybrid approach for our analysis.

III. Methodology and Implementation

The main idea behind the research was to use a single data set from twitter in order to perform a good comparison of how the different methodologies perform. In order to achieve this, the basic data collection and pre-processing involved the use of a single approach. The below subsections explain the data collection process and then provides the methodology and technical implementation of each of the four different analysis.

A. Data collection and pre-processing

The data used in the analysis was raw tweets from twitter which was extracted by the archive team[1] from twitter and stored in a compressed file. The data being used is only for one day i.e. 1st August 2019 and contains the information about the tweet id, the tweet, the user id of the user who posted the tweet and the hashtags used in the tweets. In order to gather and extract the required data, the compressed file was loaded into python using spark for multiprocessing into a spark data frame. The data was filtered as per language as English, tweets that were not retweets and contained hashtags which gave us a total count of approximately 40,000 tweets. This data was then exported into a .csv file and uploaded into a common google drive storage for all team members to access.

The implementation was carried out using google colab as a platform in which the team collaborated to perform the sentiment and emotion analysis. In order to carry out the sentiment and emotion analysis of these tweets, two group members were responsible for the emotion analysis methods and two members took charge on the sentiment analysis. The research was divided into sentiment category analysis, sentiment polarity score analysis and two different methodologies for emotion category analysis.

B. Sentiment polarity score analysis using Vader Sentiment Lexicon and Bi-Directional LSTM

The sentiment polarity score prediction involves the use of a regression based method which provides us with a value of how positive or how negative a particular text is. In order to perform this, the tweets were assigned with the polarity scores for positive, negative, neutral and compound using the Vader sentiment lexicon. The positive, negative and neutral polarities provide with the probability score of the text's positivity, negativity and neutrality between 0 and 1. The compound score provides a combination score between -1 and 1 where less than 0 is considered negative, greater than 0 is considered positive and 0 is neutral.

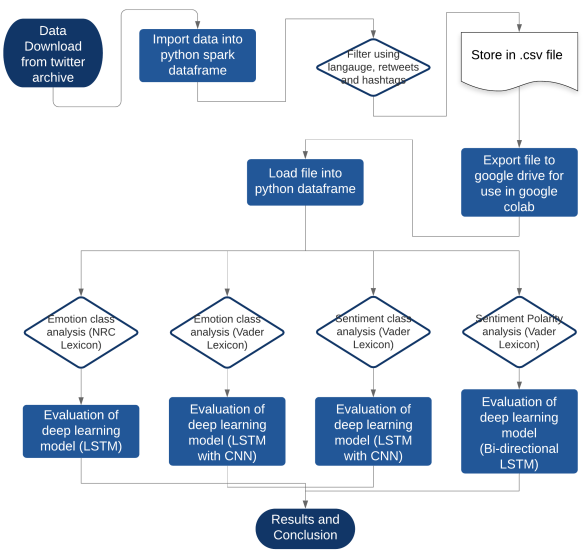

Fig. 1. Analysis Methodology

As the compound score provides us with the combined polarity value, a decision of using this for prediction purposes was made. Going by the Google S/W ratio model, it provided the direction of using tokenization and vectorization of the text along with padding the sentences to a fixed length. In order to gain the most information from these vectorized words, a combination of using GloVe as a weight matrix for embedding was used. Global Vectors for word representation (GloVe) contains a vocabulary of 400,000 tokens and was trained on a corpus of 6 billion tokens to provide with pre-trained word embeddings that help to improve the deep learning model.

The model chosen to apply for polarity score analysis was a bi-directional LSTM (Long-Short Term Memory). Bidirectional LSTM is a recurrent neural network architecture that can perform better when compared to traditional LSTMs by training two LSTMs on the input sequence. The two hidden layers of the opposite directions are connected to the same output in order to get information from past and future states simultaneously. The main advantage of this architecture is that it preserves information from the inputs that are already passed through it with the help of the hidden state. The backward running LSTM preserves the information from the future state. The other layers considered in the model were sequential, embedding, Global MaxPooling, dense and dropout. The dense layer is used to change the vector dimensions and a dropout layer prevents the model from overfitting. The maximum element of the feature vectors is selected by the maxpooling layer. The activation function used in the dense layer is ReLU which sets the negative values in the matrix to zero. Also, sigmoid activation fuction is used which transforms the input

[1] https://archive.org/details/twitterstream?and[]=year\%3A"2019"

into a value between zero and one. Mean square error is the loss function used in this regression problem which is the sum of squared distance between the target and the predicted values. Stochastic gradient descent is the optimizer used in the model which enables the model to jump to a better local minima by minimizing the error.

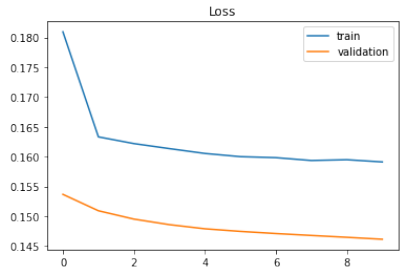

Fig. 2. Bi-directional LSTM Loss

Considering the training and validation loss of the model, it was observed that after the first epoch, the training and validation loss decreased linearly in parallel which showed that the model was not over learning and was able to perform better with more epochs.

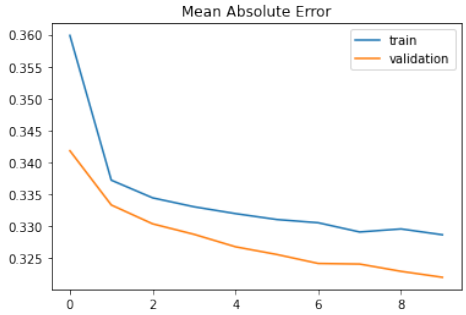

Fig. 3. Bi-directional LSTM Mean Absolute Error

As this was primarily a regression based problem, the mean absolute error was taken into account which showed a similar trend as that of loss. The model was able to perform better with more epochs on the validation set than the training set which showed that it slightly under learned but provided promising results.

C. Sentiment category analysis using Vader Sentiment Lexicon and LSTM-CNN

In order to perform the sentiment category prediction using deep learning model, the text data needs to be cleaned in order to gain the most information out of it. In order to achieve this, the garbage values or the words or combination of words that do not carry any sentiment in a sentence need to be removed. These include the urls, punctuation marks and stop words or words that act like fillers in a sentence. These were checked and removed from each of the tweets in our data before proceeding with the analysis.

The tweets by itself do not contain any sentiment in them, so to assign the sentiment scores, Vader Sentiment, a lexicon or dictionary of words that assign a sentiment polarity value was used. In order to gain the best polarity score of the word, there needs to be lemmatization and stemming to be carried out on each of the words. Lemmetization reduces the word to its base form whereas stemming reduces the word to a form that can be assigned suffixes or prefixes to make sense. It was observed that the tweets also contained emojis which show the emotion of a tweet but as the analysis considers only the words, these emojis were also removed as part of the data processing.

Once the transformation was complete, the vader sentiment analyzer was used in order to assign a sentiment score to the overall tweet. This score ranges from -1 to 1, where -1 is the most negative and 1 is the most positive. To generate classes, the sentiment score or polarity was used to determine five classes, namely ‘very positive’, ‘positive’, ‘neutral’, ‘negative’ and ‘very negative’. The initial distribution of the tweet count as per sentiment category showed that there was a huge class imbalance with neutral having the most number of tweets.

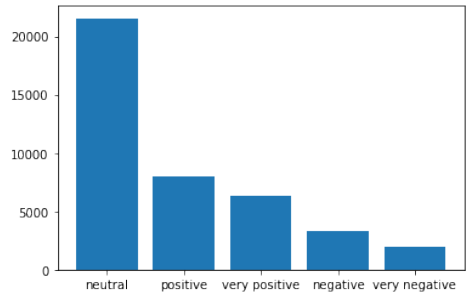

Fig. 4. Before ’Neutral’ category removal

To address this problem of class imbalance, the ‘neutral’ tweets were removed altogether from the data which reduced the data set to have approximately 19,000 rows. The methodology of this analysis was using the Google S/W ratio of finding the best suitable way for text classification where S is the total count of data present and W is the median number of words per sentence. An S/W ratio of 2174 showed the direction of the further steps to gain the optimal results.

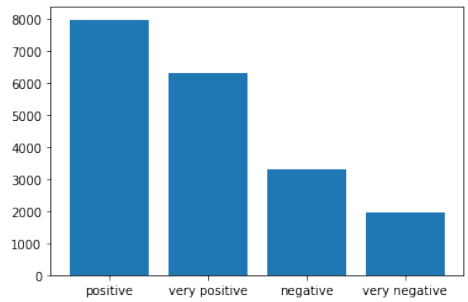

Fig. 5. After ’Neutral’ category removal

A word level tokenization and vectorization was carried out in order to convert the text data into a numerical representation. As the S/W ratio is lower than 15K, an implementation of a word embedding matrix from GloVe for twitter which decides the weight or importance of each of the word was considered. This was input as the weight matrix for the embedding layer in the model.

The neural network model applied for predicting the sentiment class was a combination of 1 LSTM (Long Short Term Memory) layer and 1 1D Convolutional layer. The first layer considered was an embedding layer that considered and assigned the various weights for the words as per the GloVe embedding matrix. This was then passed to a 1D Convolutional layer with 30 filters which was followed by the Max Pooling layer and a Spatial Dropout layer. The Spatial Dropout layer is useful to reduce the overfitting of the model where it randomly converts the input values to zero for a specific percentage of the data. The next layer was the use of LSTM, in which the model basically learns from the data and embeddings as to what information to store and what information to remove from the input data. Finally, the output of the LSTM was passed to a Dense layer for which it predicts the category of the data.

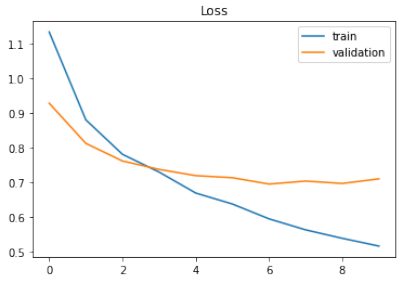

Fig. 6. LSTM-CNN Loss Plot

The model was run for 10 epochs where the loss decreased from 1.13 to 0.51 and accuracy increased from 49 to 80 percent between the first and last epoch. The training data was also randomly split into validation data where the model simultaneously performed predictions on the validation data to improve which gave a final validation accuracy of 73 percent.

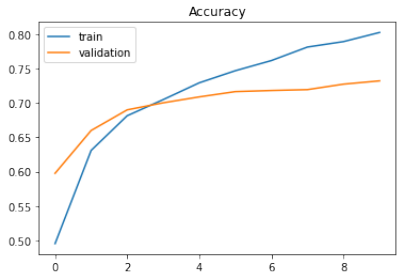

Fig. 7. LSTM-CNN Accuracy

D. Emotion category analysis using Vader Sentiment Lexicon and LSTM-CNN

As part of the data cleaning and processing for the emotion category analysis, punctuation, URLs, special characters, stop words, usernames and emoticons are removed from the dataset.The processed tweets were labeled as positive, negative, and neutral based on polarity scores obtained from using the VaderSentiment package. Using K means, five clusters were created and based on the values obtained from the above step the tweets were assigned into different clusters. The 5 clusters represent the 5 basic emotions: love, neutral, enjoyment, anger, and sadness. The obtained result was transformed from text corpus into a vector representation using tokenizer and the data was split into train and test data.

Based on the conducted literature survey, the model proposed here is a combination of Long Short-Term Memory (LSTM) and Convolution Neural Network (CNN) and is capable of identifying and classifying human emotions (love, anger, enjoyment, sadness, and neutral) in tweets. As part of the embeddings required for the data, the embedding layer applied here was without an embedding matrix and hence the layer itself learnt the weights and assigned them to the respective words in the text.

After the embedding layer, the CNN and LSTM layers have been added to the model. The CNN layer is constructed on top of the LSTM layer where the output vector of the LSTM model is fed into the CNN model as an input vector. The CNN model draws the characteristics of input sequence thereby maximizing the accuracy. A max-pooling layer is constructed above the convolution layer which is similar to the basic CNN model. Max pooling feature is used to extract the maximum feature map value. The softmax function is added as part of the last layer (Dense) which is used to extract output from the above step and feed as input to a fully connected layer to procure the probability of distribution over labels.

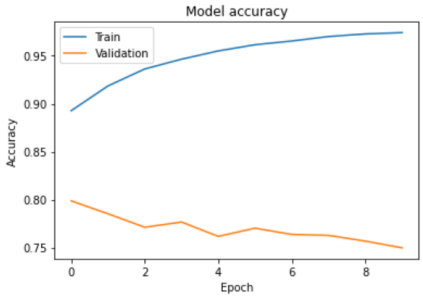

Fig. 8. LSTM-CNN Accuracy

The plot of accuracy showed that in the earliest epoch the model showed the scope for more training but during the last couple of epochs the accuracy remained at 0.9728 and 0.74. This shows that the model had over-learned the training dataset.

The plot of loss showed that the parallel plots are departing on a consistent basis. This indicated that training needed to be

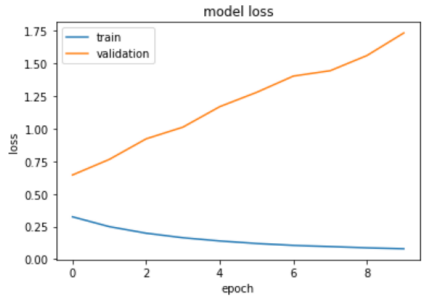

Fig. 9. LSTM-CNN Loss

stopped at an earlier epoch. Considering this, the decision was made to use an epoch of only 1 as it gave the optimum result of accuracy and loss between training and validation data

E. Emotion category analysis using NRC Lexicon and LSTM

Considering the pre-processing of the text data, a removal of URLs, punctuation, newline character and stopwords was carried out. Tokenization was carried out to convert the sentences into tokens of words and converted into lowercase. Also, stemming and lemmatization was carried in the next step to remove the prefix and suffix of the words. Lemmatization is better than stemming as it returns words with proper meaning, which is similar to the regex function. Finally, emoji icons which are expressed as symbols or special characters are removed [2].

Once the text data was cleaned nrclex package which uses NRC Lexicons was applied to the preprocessed tweet to understand its behaviour [3]. NRC lexicon provides 10 different emotions, but the maximum of raw emotion scores was assigned to the particular tweet and it became the emotion of the tweet.

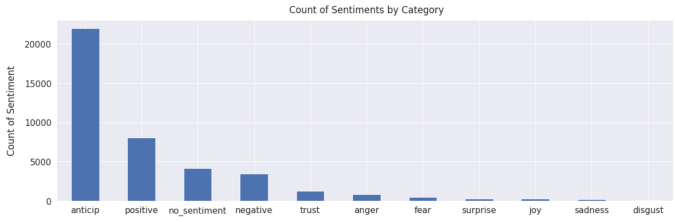

Fig. 10. Before 'anticipate' category removal

The visualization of the emotion count with respect to the tweets showed that the maximum number of tweets belonged to the emotion of anticipate. In order to avoid class imbalance, the tweets with 'anticipate' class was removed which reduced the data size from approximately 40,000 to 19,000.

In order to gain a train and test dataset, bootstrapping with replacement method was applied to split the data[4] which essentially helps to overcomes the problems of overfitting and

[2] https://towardsdatascience.com/nlp-for-beginners-cleaning-preprocessing-text-data

[3] https://pypi.org/project/NRCLex/

[4] https://medium.com/@annabiancajones/sentiment-analysis-on-reviews-train-test-split-bootstrapping-cross-validation

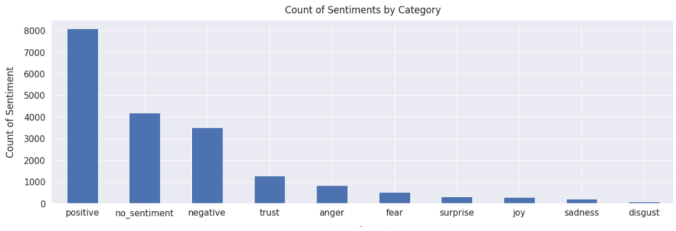

Fig. 11. After 'anticipate' category removal

underfitting by oversampling and undersampling to get the data for each class to a common value. By visualizing a word cloud to understand the popular words that represent positive, negative and disgust emotions of the tweets, it provided an idea of how the words are being used for these classes.

Fig. 12. Wordcloud of positive, negative and disgust emotions (L-R)

Based on the s/w ratio - 2409, two deep learning models CNN [5] and LSTM [6] are applied to the cleaned dataset separately for comparison purposes. The LSTM model provided the best results in terms of training as well as validation data and hence, the final model used was the LSTM model.

In order to apply the LSTM based neural network, the maximum number of words was restricted to 50,000 and the maximum length of a tweet was selected as 250. The embedding dimension, used for creating a word vector was applied without a pre-trained embedding matrix with a dimension of 100. A variational dropout layer (SpatialDropout1D) was added along with the LSTM layer for which a memory of 100 units was chosen. The final Dense softmax layer with 10 classes was created as per the number of emotions and the model was compiled using categorical crossentropy as the loss function and a batch size of 64 was was selected with 5 epochs to run.

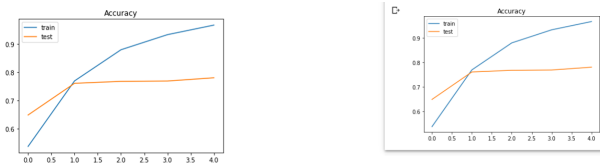

Fig. 13. Accuracy plots of dropout 0.2 and 0.5 (L-R)

The training accuracy gained in the model was 0.9677 and validation accuracy was 0.7810 for dropout rate 0.2. This

[5] https://towardsdatascience.com/cnn-sentiment-analysis-1d16b7c5a0e7

[6] https://towardsdatascience.com/multi-class-text-classification-with-lstm

showed that the model slightly overfitted the data and hence the dropout rate was increased to 0.5. This provided a lower training accuracy of 0.9129 and validation accuracy of 0.7712 which shows that the model was not overfitting as much as before.

The next section of Evaluation discusses the results of the models applied above on unseen or test data and provides an insight of how each model performed considering various evaluation parameters.

IV. Evaluation

As part of the evaluation for the above applied models, the model was used to predict the values for unseen data and test data. A comparison was made between both values and evaluated for not only the models accuracy but also the overfitting and underfitting of the model.

A. Bi-Directional LSTM sentiment polarity score

As this was primarily a regression based problem, the loss as well as the mean absolute error was considered for evalutaion. The loss value of 0.15 on test as well as training data showed that the model was not over or under fitting. In terms of the mean absolute error, a value of 0.33 for both training and test data was observed. These values show that for sentiment polarities between -1 and 1, the MAE was only 0.33 which showed significant accuracy in predicting the polarity values.

B. LSTM-CNN Sentiment category analysis

The data was infused with four primary categories of sentiments, for which the model was applied on the test data and a confusion matrix was created in order to understand how the model performed on each category of sentiment.

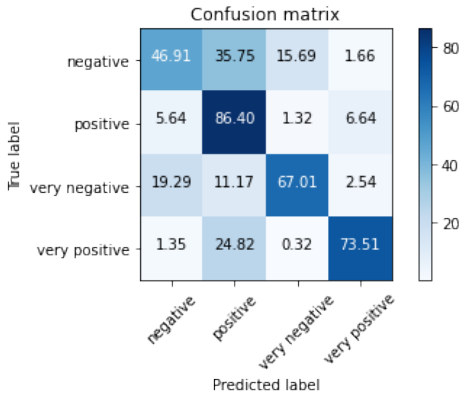

Fig. 14. LSTM-CNN Sentiment category confusion matrix

The overall accuracy of the model turned out to be 73 percent on test data however an accuracy of 80 percent on the training data showed that the model slightly overlearned. For the categories of positive and very positive, the model was able to perform with good accuracy, greater than the overall accuracy. However, for the very negative category, the accuracy dropped to 67 and the model performed the worst for the negative category with an accuracy of only 46.91. This shows that the overlearning of the model happened majorly for the categories of negative and very negative.

C. LSTM-CNN Emotion category analysis

The emotions extracted using vader sentiment for the tweets contained five categories, namely anger, enjoyment, love, neutral and sadness. The overall accuracy achieved by the model on the test data was 86 percent which shows that the model performs much better on unseen data rather than seen data which shows a little instability within it.

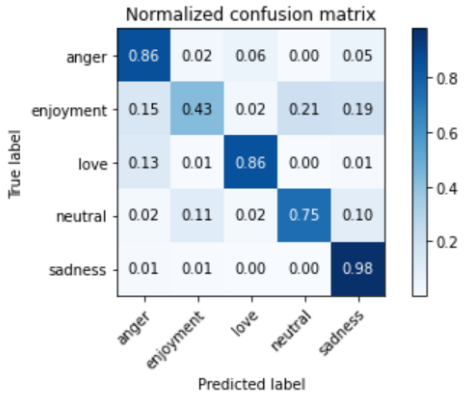

Fig. 15. LSTM-CNN Emotion category confusion matrix

The model was able to predict the love, anger, sadness and neutral categories with good accuracy but the enjoyment category showed poor results of only close to 40 percent accuracy. The imbalance of classes in the data seem to be one of the reasons why the model overlearned, especially for the above mentioned category.

D. LSTM Emotion category analysis

Using the NRC Lexicon, 10 emotions were extracted from the data, namely anger, disgust, fear, joy, negative, no emotion, positive, sadness, surprise and trust. The overall accuracy on the test data achieved was 76.2 percent, however, the confusion matrix provides results skewed towards only five categories, negative, no emotion, positive, anger and trust. Also, considering that the training data accuracy was 90 percent, the model clearly overlearned and this might help to possibly explain the skewness in the confusion matrix.

In order to compare whether the LSTM model chosen was a good performer, CNN was also applied on the same data considering that Google's S/W ratio based method showed CNN performing better than other models. However, in this case, LSTM shows better performance and accuracy for text analytics due to the nature of sequential processing of previous texts. Also, tuning hyperparameters of CNN and LSTM models, overfitting was reduced. Usage of GPU instead of CPU increased the processing time in CNN by 50 percent but the performance was still comparatively lower than LSTM. The CNN model provided only an accuracy of 46 percent on the test data, which showed that LSTM or RNN based models

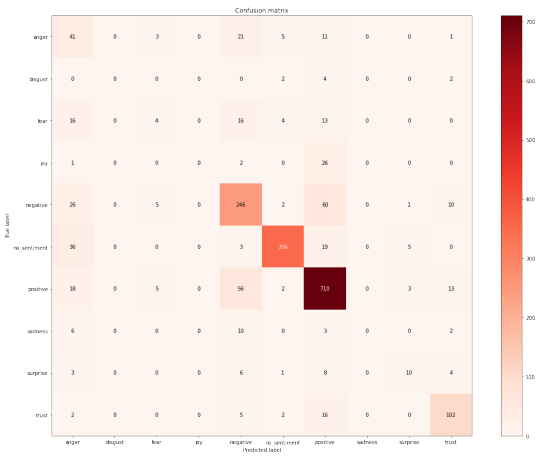

Fig. 16. LSTM Emotion category confusion matrix

clearly outperform the classification problem into account here.

E. Comparative results

The regression based problem of detecting sentiment polarity was one of its kind in the project and rest of the analysis involved, provided categorical information. However, the categorical information was derived from the numerical scores from the lexicons by limiting a threshold on the same. The regression based problem shows promises of an accurate detection of the polarity of how positive or negative the sentiment is. The underlying limitation of using regression is making a sense out of the regression results as the value has to be interpreted differently than just getting a class as an output.

The sentiment category analysis provides a good accuracy among detecting the classes as a whole, however the analysis only provides information on whether the text was positive or negative. Comparing this with the emotion analysis, the detection of emotions apart from sentiment can make the understanding of the text data far better allowing for a range of use cases.

The two emotion analysis approaches considered in this research each have their own positives and negatives. The LSTM-CNN based emotion analysis has only the basic five emotions into account but promises a very good accuracy in terms of predicting them. The LSTM emotion analysis model may have more detailed and fine grained emotions, but shows a lower accuracy than the LSTM-CNN approach. Having more number of categories may have an adverse impact on the predictive capabilities but by hyperparameter tuning, can be improved. With the use of basic five emotion analysis, the classification might have been restricted but the predictive capabilities show significant improvement owing to having less classes to deal with in the data.

As per the literature survey conducted, the SVM with SentiWordNet applied provided only an accuracy of 58 percent in research [13]. Comparing with the models applied above, the neural network based approaches provide a greater accuracy as they also take into account the sequential data while training. The CNN model applied in [2] provided a good accuracy of 87 percent but did not work well with our dataset which drove the decision of using RNN based models. The RNN and LSTM models applied in [4] and [5] respectively show great performance improvement but only take into account two basic categories of sentiment (positive and negative) whereas the approaches used in this research use more categories to define the data. This might be one of the underlying causes of slightly lower accuracies. The hybrid LSTM-CNN applied in [9] show lower predicitive capability compared to [4] and [5] but face the same limitation of low category count. The LSTM-CNN applied on emotion analysis in this research show slightly better results and also overcomes the class limitations presented in the above research papers.

V. Conclusion and Future Work

From the above analysis, we can conclude that the Bi-directional LSTM model is appropriate where regression analysis is required for sentiment analysis. CNN+LSTM model for sentiments plays an important role in classifying sentiments accurately. To obtain the basic emotions behind the sentiments, a combination of the LSTM and CNN model was outlined and has high accuracy. If the requirement is to have a high-level classification of emotions, the LSTM based model is well suited.

The twitter dataset which was considered for the study consists of general data and is not biased or topic-specific. Classification of sentiments and emotions were performed using both supervised and unsupervised learning models. The obtained accuracy was reasonably good and since the hybrid methods were implemented, it can be applied to any raw data. Depending on the business case scenario and the required granularity of categories, different classification models outlined in this paper can be used. For an accurate representation of sentiments, regression analysis can be used but there is also a limitation in generalization in the context of large output. The accuracy of models saw a considerable dip with the increase in classification categories. Based on different use cases, the decision can be made to either compromise on accuracy or granularity of categories in either sentiments or emotions.

The applied methodology can be used to understand the underlying emotion and intent of people which can further be used by organizations to enhance their product features. It can also be applied where reviews and ratings play an important role. The emotional value of reviews can be calculated and added to the numerical rating for a better and accurate result.

For the future study, a larger dataset can be used and a comparison between traditional machine learning models and applied models can be outlined to identify the better performing model. The models can be further enhanced to evaluate behavioural patterns from the text data presented.

References

[1] W. Xing and A. Ghorbani, “Weighted PageRank algorithm,” Proc. - Second Annu. Conf. Commun. Networks Serv. Res., pp. 305–314, 2004.

[2] Z. Jianqiang, G. Xiaolin and Z. Xuejun, "Deep Convolution Neural Networks for Twitter Sentiment Analysis," in IEEE Access, vol. 6, pp. 23253-23260, 2018.

[3] V. Prakruthi, D. Sindhu and D. S. Anupama Kumar, "Real Time Sentiment Analysis Of Twitter Posts," 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 2018, pp. 29-34.

[4] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," Electron., vol. 9, no. 3, 2020.

[5] M. Y. Day and Y. Da Lin, "Deep learning for sentiment analysis on google play consumer review," Proc. - 2017 IEEE Int. Conf. Inf. Reuse Integr. IRI 2017, vol. 2017-January, pp. 382–388, 2017.

[6] R. Bose, R. K. Dey, S. Roy, and D. Sarddar, "Analyzing Political Sentiment Using Twitter Data," vol. 107, no. January, S. C. Satapathy and A. Joshi, Eds. Singapore: Springer Singapore, 2019, pp. 427–436.

[7] B. Gokulakrishnan, P. Priyanthan, T. Ragavan, N. Prasath and A. Perera, "Opinion mining and sentiment analysis on a Twitter data stream," International Conference on Advances in ICT for Emerging Regions (ICTer2012), pp. 182-188, 2012.

[8] A. Tholusuri, M. Anumala, B. Malapolu, and G. Jaya Lakshmi, "Sentiment analysis using LSTM," Int. J. Eng. Adv. Technol., vol. 8, no. 6 Special Issue 3, pp. 1338–1340, 2019.

[9] M. H. Abd El-Jawad, R. Hodhod and Y. M. K. Omar, "Sentiment Analysis of Social Media Networks Using Machine Learning," 2018 14th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 2018, pp. 174-176.

[10] G. Williams and A. Mahmoud, "Analyzing, Classifying, and Interpreting Emotions in Software Users' Tweets," 2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion), Buenos Aires, 2017, pp. 2-7.

[11] R. Ahuja, R. Gupta, S. Sharma, A. Govil and K. Venkataraman, "Twitter based model for emotional state classification," 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, 2017, pp. 494-498.

[12] O. Appel, F. Chiclana, J. Carter, and H. Fujita, "A hybrid approach to the sentiment analysis problem at the sentence level," Knowledge-Based Syst., vol. 108, pp. 110–124, 2016.

[13] I. Gupta and N. Joshi, "Enhanced twitter sentiment analysis using hybrid approach and by accounting local contextual semantic," J. Intell. Syst., vol. 29, no. 1, pp. 1611–1625, 2020.

[14] M. U. Salur and I. Aydin, "A Novel Hybrid Deep Learning Model for Sentiment Classification," IEEE Access, vol. 8, pp. 58080–58093, 2020.

[15] A. Hassan and A. Mahmood, "Deep Learning approach for sentiment analysis of short texts," 2017 3rd Int. Conf. Control. Autom. Robot. ICCAR 2017, pp. 705–710, 2017.

[16] A. Sboev, T. Litvinova, I. Voronina, D. Gudovskikh, and R. Rybka, "Deep Learning Network Models to Categorize Texts According to Author's Gender and to Identify Text Sentiment," Proc. - 2016 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2016, pp. 1101–1106, 2017.

[17] S. S. Kolekar and H. K. Khanuja, "Sentiment Analysis using Deep Learning on GPU," 1st Int. Conf. Data Sci. Anal. PuneCon 2018 - Proc., pp. 1–5, 2018.

[18] A. Sarlan, C. Nadam and S. Basri, "Twitter sentiment analysis," Proceedings of the 6th International Conference on Information Technology and Multimedia, pp. 212-216., 2014.

[19] M. Neethu and R. Rajasree, "Sentiment analysis in twitter using machine learning techniques," Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp. 1-5, 2013.

[20] A. Severyn and A. Moschitti, "Twitter Sentiment Analysis with Deep Convolutional Neural Networks," Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 959-962, 2015.

[21] V. S. Pagolu, K. N. Reddy, G. Panda and B. Majhi, "Sentiment analysis of Twitter data for predicting stock market movements," International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), pp. 1345-1350, 2016.

[22] N. Azam, B. Tahir, and M. A. Mehmood, "Sentiment and Emotion Analysis of Text: A Survey on Approaches and Resources," pp. 87–94.