2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) | 978-1-6654-6200-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCCIS56430.2022.10037634

Analyzing Sentiment Towards a Product using DistilBERT and LSTM

Vishal Pramanik
Department of Computer Science & Engineering
Indian Institute of Technology Bombay
Mumbai, India
Email: vishalpramanik35@gmail.com

Maisha Maliha
Department of Computer Science & Engineering
Indian Institute of Technology Bombay
Mumbai, India
Email: maishamaliha776@gmail.com

***Abstract*— For improving customer experience towards a product, analyzing the customer's review is the best possible solution, where customers' positive and negative opinions can be identified. In our study, we have developed two different supervised natural language task models using (1) deep neural Long Short-Term Memory (LSTM) model and (2) advanced pretrained Distil Bidirectional Encoder Representations from Transformers (DistilBERT), where in the first model, we have analyzed how each of the words in a sentence contribute to the sentiment of that sentence, by training LSTM with our 50,000 Amazon reviews corpora, publicly available in the Amazon website. In the second model, we have examined how the positional embeddings of the words in a sentence and multi-headed self-attention help in faster with better sentence representation for sentiment analysis by finetuning DistilBERT, which contains 66 million parameters and 6 layers of encoders. The main contribution of our research includes- (1) a comparative analysis of the efficacy between these above mention models, where the Transformer based model outperforms the LSTM model in all metrics of evaluation, and (2) the establishment of a much faster and lighter sentiment analyzer model, DistilBERT.**

Keywords*— *DistilBERT, LSTM, sentiment analysis, amazon reviews

I. Introduction

With the growing numbers of products, people's have options to choose different products as per their necessities and choices. Due to that, it is really necessary for the product makers to know their customers choices towards a product so that the makers can be able to develop the features of their product. Daily a diverse number of data have been shared in a form of suggestions, review, tweets, reactions and comments. For reaching the maximum goal of customer satisfactions, opinion mining is undoubtedly playing a significant role for justifying positivity, neutrality and negativity of a particular product review [1]. Though it is quite challenging task to analyze each people's sentiment towards a product as there are billions people live in this world who share billions of opinions each day [2]. Due to that different researchers have proposed different methodologies for mining people's opinion accurately. Therefore, in our research, we have implemented two different models that makes the sentiment analysis process easier. Though our model shows better result by the help of DistilBERT approaches which is a transformer-based model that has been published by Google AI Language. It can be used for variety of Natural Language Processing tasks which are Question Answering, Sentiment Analysis, Natural Language inference and others. It uses the bidirectional training of transformers and attention technique for language modelling. Our second analyzer model which is implemented by the help of LSTM, predicts the rating by retaining the information of the previous words from a sentence. Our study is organized as follows. Section II refers the previous researcher's work. Section III focuses on the major terminologies which we have used for implementing our proposed work. The techniques and implementation task of our models have been discussed in Section IV where Section V shows the experimentation of our models. Section VI analyzes the result and Section VII concludes our study.

II. Related Works

For analyzing public views from twitter data, the authors [3] have proposed Bag of Words and Terms Frequency In-verse Document Frequency Vectorizer (TFIDF) based models in order to classify positive and negative tweets. Though the models have got higher accuracy in terms of analyzing sentiment but in terms of providing linguistic information between the words, it does not perform that much well. On the other hand, our model which uses DistilBERT can able to provide proper semantic information with a decent accuracy. In [4], the authors have used DistilBERT based model in order to classify positive, negative and neutral sentiments of banking financial news where the distilBERT based model has been passed into four different supervised machine learning algorithms. However, the accuracy of the Random Forest with DistilBERT was higher than the other machine learning based classifiers but the highest accuracy was below 80% in the above model. In our model, we have used distilBERT and have increased the accuracy above 80% with proper hyperparameters tuning, skewed dataset handling, multi-class classification and emoji handling. In [5], an unsupervised learning-based word embedding model have been introduced for predicting sentiments from twitter data. For Turkish natural language processing research, the authors [6] have proposed two major approaches which includes finetuning process of multilingual BERT model and the machine translation process of Turkish text into English text. Though the BERT based model makes the process less faster than the DistilBERT model. In terms of sentiment analysis, different types of issues need to be dealing with such as polarity shifting, barrier of reaching higher accuracy, data sparsity based problems and binary classification problems. Though different types of methods have been proposed to analyze sentiments including naive bayes, support vector machine, maximum entropy and other supervised or unsupervised based machine learning algorithms but none of these can be efficiently found effective for extracting sentiment features from a given text. Thus a comparative analysis between different sentiment analysis approaches which have been proposed in various research paper's have been discussed in [7], in order to show whether these methodologies can solve the polarity shifting or binary classification problems. Two different natural language processing techniques such as word sense disambiguation and semantics have been proposed [8] to analyze sentiments where the study shows that ensemble classifiers perform better than other machine learning algorithms. Ensemble classification techniques include different independent classifiers for solving classification based problems. In [9], the authors have used Recurrent Neural Network and Word2Vec based models in order to analyze sentiment of Indonesian language where the study shows higher accuracy. To fulfill the gaps of the existing works, we propose our models which handle not only the multi-class classifications of sentiments along with emoji handling but also analyze the sentiment of a review in a much better and faster way, and also can capture the semantic information present in a review.

III. Background and terminologies

We have used deep neural models and word embeddings to achieve our task of sentiment analysis of Amazon reviews. The following terminologies are important to describe our work.

A. *Word Embedding*

In our task we have used the most commonly used word embedding Word2Vec while training LSTM in one of our experiment. Word Embedding is a technique to represent words in an n-dimensional space. The value of n depends on the data vocabulary, the type of which word embedding are used and the user using the word embedding. These embeddings are capable of capturing the context of the words (both semantic and syntactic) in a sentence. The cosine values of the similar words are high. There are pre-trained word embeddings also-Glove, Word2Vec and fasttext. These embeddings are already trained on a huge amount of data.

B. *Transformer*

One of our model, which follows the implementation using DistilBERT is the encoder part of the vanilla Transformer where Transformer is a Encoder-Decoder model mainly used for the task of machine-translation. On the contrary to the traditional RNN and LSTM models which run sequentially, Transformers can run paralelly due to the positional embedding and the Attention mechanism. The Encoder part learns the sentence representation and creates a context vector which is then passed to the Decoder part. The Decoder part generates the output from the vector using auto-regressive technique.

C. *Deep Neural Networks*

1) *LSTM:* The Long Short-Term Memory (LSTM) networks was introduced to overcome the vanishing gradient problem in traditional RNNs. LSTM has feedback mechanism, therefore when we give our review to the network, it does not treat each word individually. It retains the information from the previous words before predicting the rating. The LSTM works with a series of gates and by using LSTM, it can control the amount of information it needs to retain and amount of information it needs to forget from the input sentence.

2) *DistilBERT:* DistilBERT performs faster than BERT base model by distilling it where it uses nearly half number of parameters than BERT base model. The token-type embeddings and the pooler have been removed from DistilBERT and the number of layers is reduced by a factor of 2(i.e. 6 encoder blocks). Since the target or aim of DistilBERT is to create a language model or representation of the review given as input, hence it is made of only the Encoder part of the vanilla Transformer. On the contrary to directional models which reads input from left-to-right or right-to-left, DistilBERT takes both the context (left context and right context) of the words in the review sentence.

IV. Techniques and Implementation

In this paper we have predicted the rating based on the users review towards a product. We have tested this dataset for two models- (a) LSTM and (b) DistilBERT. After that we have compared the different metrics that we have calculated during the experiment. The algorithm that we have followed for all the models is given in the figure 1.

A. Implementation using LSTM

In this section we have discussed how the LSTM predicts the rating of a review. The functionality of each gate and the flow of information through them has been discussed here. LSTM uses three gates to train itself- (1) Forget Gate, (2) Input Gate and (3) Output Gate. The implementation process using LSTM have been shown clearly in the figure 2.

1) Forget Gate: This gate decides how much information of review it needs to forget by selecting the words which have no contribution to the output rating. First the hidden input state and a review is given as input to a neural network. The output of the network is a vector whose values ranges in between 0 and 1, as the output layer of the network uses sigmoid function. This network is trained so that it's output remains closed to 0 when a word of the review is deemed irrelevant and closed to 1 when relevant. This vector is then pointwise multiplied with the previous cell state. The dot product refers that components of the cell state vector which have been deemed irrelevant by the forget gate network will be multiplied by a number close to 0. Thus, this will make less influence on the following steps.

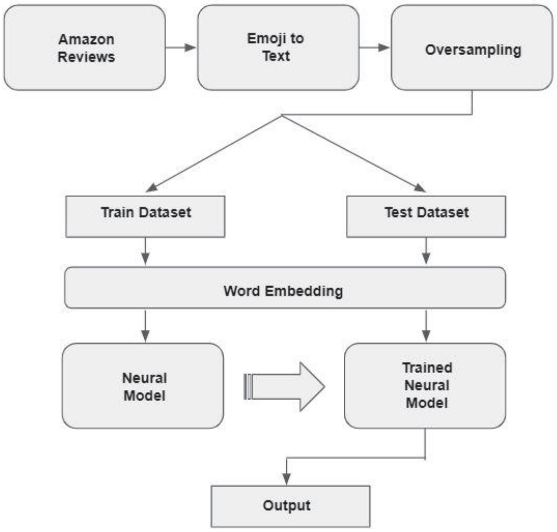

Fig. 1. The Proposed Algorithm of our work to train and test the models

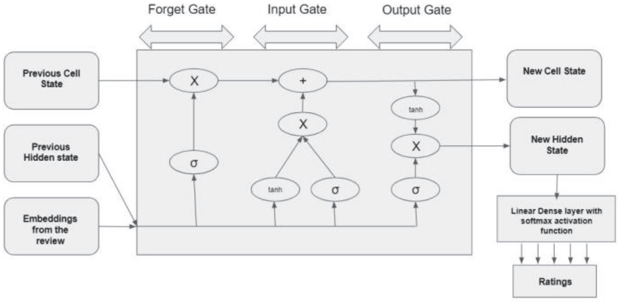

Fig. 2. The internal structure of our sentiment analyzer model using LSTM

Algorithm for Forget Gate

Step 1: $F_t = \sigma(E_t * U_f + H_{t-1} * W_f)$
Step 2: $C_{t-1} * F_t \sim 0$... if $F_t \sim 0$ (keep less information)
$C_{t-1} * F_t \sim C_{t-1}$...if $F_t \sim 1$ (keep more information)

In the above algorithm for the Forget Gate, E_t is the input review at time stamp t. U_f is the trainable weight associated with the input review at time stamp t. H_{t-1} is the hidden state at time stamp t −1. W_f is the trainable weight associated with the hidden state at time stamp t. C_{t-1} is the cell state at time stamp t −1.

2) Input Gate: This gate consists of two neural networks- new memory network and input gate. This layer decides how much new information needs to be added to the cell state. The input to both the networks is the hidden input state and a review from our dataset. The network layer contains *tanh* activation function. The output vector of this network essentially contains information from the new input review given the context from the previous hidden state. For the other network, i.e. the activation function of the input gate is sigmoid. This network decides how much of the new input data is worth keeping. Similar to forget gate network, the output vector is between [0,1]. The dot product of the vectors of the two networks is added to the cell state.

Algorithm for Input Gate

Step 1: $I_t = \sigma(E_t * U_i + H_{t-1} *W_i)$
Step 2: $N = \tanh(E_t * U_c + H_{t-1} *_{Wc})$ (this contains the new data)
Step 3: $C_t = (F_t * C_{t-1} + I_t * N_t)$ (the cell state of LSTM is updated)

In the algorithm for the Input Gate, E_t is the input review at time stamp t. U_i is the trainable weight associated with the input review at time stamp t. H_{t-1} is the hidden state at time stamp t − 1. W_i is the trainable weight associated with the hidden state at time stamp t. C_t is the cell state at time stamp t. C_{t-1} is the cell state at time stamp t −1.

3) Output Gate: This gate is used to update the hidden state of the LSTM. It contains a neural network which performs the same work and input of the forget gate neural network. It uses the sigmoid function as activation and gives a vector of values in between [0,1]. The cell state vector (the long-term memory) is passed through a *tanh* function to squeeze its value in between [-1,1]. Then the two vector is pointwise multiplied and then added to the previous hidden state. The hidden state is then passed through a linear layer to give the rating as output.

Algorithm for Output Gate

Step 1: $O_t = \sigma(E_t * U_o + H_{t-1} * W_o)$
Step 2: $H_t = O_t * \tanh(C_t)$
Step 3: Review® = Softmax(H_t)

This is the algorithm for the Output Gate where E_t is the input review at time stamp t. U_o is the trainable weight associated with the input review at time stamp t. H_{t-1} is the hidden state at time stamp t −1. Wo is the trainable weight associated with the hidden state at time stamp t. C_t is the cell state at time stamp t.

B. Implementation using DistilBERT

In this section we have discussed how the model predicts the rating based on the review given to it as input. The functioning of the Encoder block along with its Self-attention and Feed Forward layer has been discussed as shown in figure 3.

1) Encoder Block: The Encoder block of the Transformer is made up of two important components- the self-attention layer and the Feed Forward Neural Network layer. Firstly The review sentence that is given to the Encoder as input, passes through a self-attention layer after that it's output goes to the feed-forward neural network layer. Then the output of the neural network layer is given to the next Encoder block. In this way DistilBERT learns the context of each word (both left and right context).

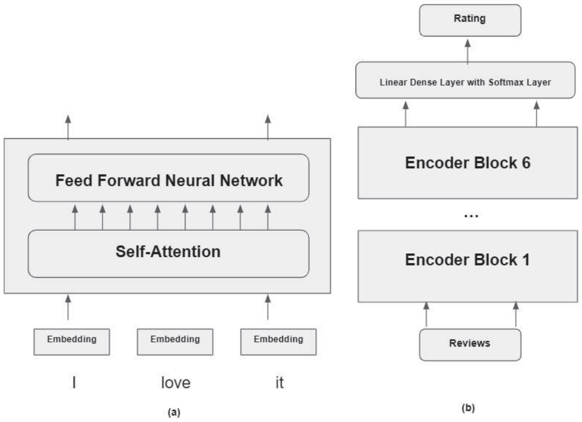

Fig. 3. In figure (a) the internal structure of a single Encoder block of DistilBERT has been depicted. The input to the Encoder block are embeddings, and the block is comprised of a Self-Attention layer and a Feed Forward layer. Figure (b) is the structure of DistilBERT containing 6 Encoder blocks

2) Self-Attention: For calculating the self-attention from the encoder, first the embeddings of the words of the reviews (for the first encoder block) or the output vector from the previous layer (E) are taken as input. The Key(K), Query(Q) and Value(V) vector is calculated by multiplying the input vector to three trainable weights(W^k, W^q, W^v). Once the three vectors are calculated then we have computed a score by the dot product of K and Q for each word with respect to a particular word. The score determines how much focuses are needed on the other parts of the input sentence as we have encoded a word at a certain position. Then we have divided the scores by 8 (the square root of the dimension of the key vectors used in the paper – 64), which is then passed through a softmax layer which normalizes the scores so that they are all positive and also added up to one. The softmax score is then multiplied with V and then summed up which is given as output to the next encoder. The algorithm to calculate the self-attention for a single word is given below.

Let us consider the review "Good Product"
Let X_1 be the embedding for "Good" and X_2 be the embedding for "Product"

Algorithm

Step 1: For the word "Good" the Q, K, V values are being calculated: $Q_1 = X_1 \cdot W^q$, $K_1 = X_1 \cdot W^k$, $V_1 = X_1 \cdot W^v$
Step 2: For the word "Product" the Q, K, V values are being calculated: $Q_2 = X_2 \cdot W^q$, $K_2 = X_2 \cdot W^k$, $V_2 = X_2 \cdot W^v$
Step 3: Score for both the words is calculated w.r.t "Good". $S_1 = Q_1 \cdot K_1$,$S_2 = Q_1 \cdot K_2$
Step 4: Divide both the scores by 8
Step 5: Perform softmax on the scores using the Equation1 and store the values in SM_1 and SM_2
Step 6: $P_1 = V_1 \cdot SM_1$ and $P_2 = V_2 \cdot SM_2$
Step 7: $S = P_1 + P_2$
S is the output of the self-attention layer for the word "Good"

This algorithm shows how to calculate the self-attention of particular word (in the above case we are considering the word "Good") with respect to the other words.

3) Prediction Layer: The last layer implements the softmax function using the equation 1 in order to calculate the probabilities of all the five ratings.

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_k \exp(x_k)} \qquad (1)$$

In the equation, k is the number of classes, which is 5 in our case. Then the class with the highest probability is chosen as the winner class.

V. Experiments

A. Data Collection and Pre-processing

The data has been web-scrapped and collected from the Amazon website. The original dataset contains 50000 reviews of various customers around the globe for different electronic products. The data contains two column- 1) Reviews and 2) the Ratings. The Reviews column contains a verbal description of the sentiment of the consumer and the Ratings column contains a rating out of 5 where 1 being the worst and 5 being the best. The data statistics has been shown in the figure 4.

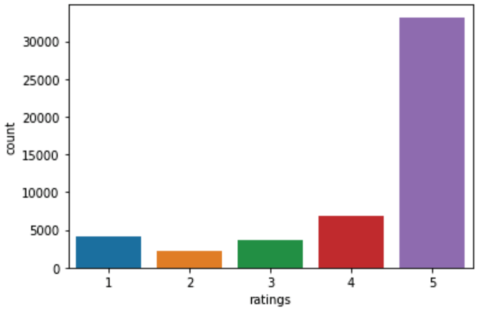

Fig. 4. The statistics of the data

1) *Emoji Handling:* In our models we have handled a specific number of emojis, in case they are given with a review. We have converted the emoji to text, which is then passed to the neural network model. The emojis given in figure 5 has been converted to word "good" while the emojis given in figure 6 has been converted to "bad". The "good" will make model to predict higher ratings while the "bad" word to lower scores.

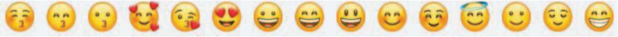

Fig. 5. These are emojis, if present in the review, it will be converted to "good"

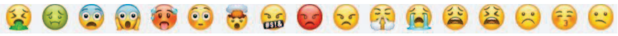

Fig. 6. These are emojis, if present in the review, it will be converted to "bad"

2) *Skewed Dataset handling:* If the number of reviews with rating 5 is more than the other ratings, the the skewed data can make the model biased towards rating 5. Therefore we have balanced the data with oversampling. We have oversampled each reviews to the number of reviews of rating 5, i.e. 33193. After oversampling, our dataset contains 165965 reviews with their corresponding rating.

B. Hyperparameters tuning

We have experimented both the LSTM model without any pretrained word embedding and the LSTM model with pretrained Word2Vec word embedding. For both of the model 128 hidden layers are used with a single dense layer where 5 neurons have been used. We have used the softmax activation function for the 5 classes of rating. For both of the model The number of epochs is 50, learning rate 0.0001 and batch size is 32. The model gained highest accuracy with the given number of epochs. The third experiment is with DistilBERT model where the number of epochs is 6000 and learning rate 0.0001 and the model gained highest accuracy with the given number of epochs

VI. Results and Analysis

In this section we have compared the results that we got after experimenting the following models with our dataset- (a) LSTM with pretrained Word2vec word embedding, (b) LSTM without any pretained word embedding and (c) DistilBERT. The results have been compared and shown in table I. It is clear from the table that DistilBERT outperforms the LSTM model with and without Word2Vec word embeddings.

Models	Metrics			
	Accuracy (%)	F1-Score	Recall	Precision
LSTM (Word2Vec)	81	0.808	0.772	0.758
LSTM	78	0.76	0.762	0.772
DistilBERT	**84**	**0.818**	**0.84**	**0.836**

TABLE I

We have analysed two outputs from our model with a visualisation technique. The visualisation is created with the LIME [10] technique. The input is perturbed at random to see how the prediction changes. By using the linear interpretable model, it can infer the relative importance of different words to the final prediction. The GREEN words contributed towards the prediction of the model while the words in RED (and PINK) detract from the model prediction. Shade of colour denotes the strength or size of the coefficients in the inferred linear model. This has been explained in details in the below tables with the help of a negative and a positive example. Below is the negative example with the review: "the quality of the product did not reach my expectations".

y=1 (probability **0.398**, score **-0.525**) top features

Contribution?	Feature
+0.322	Highlighted in text (sum)
-0.846	BIAS

the quality of the product did not reach my expectations

For the class 1, the words with green colour, i.e. 'did', 'not', 'my' have positive contribution to the prediction for class 1 while the red coloured words, i.e. 'the', 'quality', 'expectations' have negative contribution to the class. In the above table, y=1 have been considered, which refers the negative rating (1) and for the negative rating, our model focuses more on the negative word 'not'. Due to that the color of the 'not' word is more deeper than the other green coloured word such as 'did', 'reach', 'my'. The deeper the colour, the greater will be the contribution of that word. This has been detected by the last dense layer of the DistilBERT model with the softmax function as shown in figure 3.

y=2 (probability **0.351**, score **-0.718**) top features

Contribution?	Feature
+0.280	Highlighted in text (sum)
-0.998	BIAS

the quality of the product did not reach my expectations

This class has the highest probability which is 0.351 as most of the words are has a positive contribution to this class, with the word 'not' having the highest contribution, because of that, it has deeper green colour.

y=3 (probability **0.215**, score **-1.379**) top features

Contribution?	Feature
-0.522	Highlighted in text (sum)
-0.827	BIAS

the quality of the product did not reach my expectations

For this class, the words- 'quality' ,'of', 'the', 'product', 'not' and 'expectations' has made the positive contribution while the words 'did' and 'reach' has made the negative contribution. The overall softmax probability is 0.215

y=4 (probability **0.028**, score **-3.617**) top features

Contribution?	Feature
-0.600	BIAS
-3.017	Highlighted in text (sum)

the quality of the product did not reach my expectations

It can be seen from the above table that the words 'quality', 'of', 'the', 'product' and 'expectations' contribute positively to this class therefore it is colored by lime green while the words 'did', 'not', 'reach' and 'my' detract from the class and therefore coloured as light pink.

y=5 (probability **0.007**, score **-4.958**) top features

Contribution?	Feature
+0.108	BIAS
-5.066	Highlighted in text (sum)

the quality of the product did not reach my expectations

The words- 'quality', 'my' and 'expectations' has made the positive contribution for this class y=5, while the words 'of', 'the', 'product', 'did', 'not' and 'reach' has made the negative contribution. The word 'not' has the greatest negative contribution among the other words. The overall softmax probability is 0.007. Now we go through an example with positive sentiment. Here the review is: ‘the phone is awesome”. A detailed explanation about the contribution of the different words in the review have been given below.

y=1 (probability **0.009**, score **-4.735**) top features

Contribution?	Feature
-0.497	BIAS
-4.238	Highlighted in text (sum)

the phone is awesome

y=1 is a negative rating and the review definitely belongs to higher ratings which has been correctly identified by DistilBERT. The model has focused more on the word ’awesome’ which has negative contribution for this class than the other words. Hence the overall probability of the class becomes 0.009 and fails to become the winner class.

y=2 (probability **0.008**, score **-4.914**) top features

Contribution?	Feature
-0.595	BIAS
-4.319	Highlighted in text (sum)

the phone is awesome

For this class where the rating is 2 (as it is given y=2), the words ’the’ and ’phone’ has the positive contribution while the words ’is’ and ’awesome’ has the negative contribution. The softmax layer output is 0.008.

y=3 (probability **0.010**, score **-4.628**) top features

Contribution?	Feature
-0.485	BIAS
-4.143	Highlighted in text (sum)

the phone is awesome

For this class, ’the’, ’phone’ has positive contribution while ’is’ and ’awesome’ has the negative contribution. The softmax output is 0.010.

y=4 (probability **0.024,** score **-3.757**) top features

Contribution?	Feature
-0.559	BIAS
-3.198	Highlighted in text (sum)

the phone is awesome

For this class, all words has negative contribution except for “is” which has neutral effect. The softmax output is 0.024.

y=5 (probability **0.024**, score **-3.757**) top features

Contribution?	Feature
+2.334	Highlighted in text (sum)
+0.002	BIAS

the phone is awesome

For the class 5 where the rating is 5, the word with green colour, i.e. ’awesome’ have positive contribution to the prediction for class 5 while the red coloured words, i.e. ’the’, ’phone’, ’is’ have negative contribution from the class. In the above table, y=5 have been considered which refers the positive rating (5) and for the positive rating, our model focuses more on the positive word ’awesome’. Due to that the color of the ’awesome’ word is deeper green. As the deeper the colour, the greater will be the contribution of that word.

VII. Conclusion and Future Work

In our research, we have proposed a model using distilBERT which processes faster in order to analyze sentiment and compares the result with our other model which is implemented using LSTM. Though lots of researcher have proposed different methods using machine learning, BERT, TFIDF and so on, but in every models there are some disadvantages such as TFIDF model can not store the semantic information, ML based model can not give better result in terms of polarity shifting. However, by using distilBERT model, we have given semantic meaning in each of the words in a sentence and made the process faster than BERT model. In our future work, we will try to improve the performance metrics of our model.

References

[1] S. Vanaja and M. Belwal, ”Aspect-Level Sentiment Analysis on E-Commerce Data,” 2018 International Conference on Inventive Re-search in Computing Applications (ICIRCA), 2018, pp. 1275-1279, doi: 10.1109/ICIRCA.2018.8597286.

[2] Akram W, Kumar R. A study on positive and negative effects of social media on society. Int J Comput Sci Eng 2017;5:347-54

[3] M. R. Hasan, M. Maliha and M. Arifuzzaman, ”Sentiment Analysis with NLP on Twitter Data,” 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), 2019, pp. 1-4, doi: 10.1109/IC4ME247184.2019.9036670.

[4] M. T. Varun Dogra Assvknj and, “Analyzing DistilBERT forsentiment classification of banking financial news,” in In-telligent Computing and Innovation on Data Science,S.-L. Peng, S.-Y. Hsieh, S. Gopalakrishnan, and Balaganesh,Eds., p. 582, Springer Singapore, Singapore, 2021.

[5] Z. Jianqiang, G. Xiaolin and Z. Xuejun, ”Deep Convolution Neural Networks for Twitter Sentiment Analysis,” in IEEE Access, vol. 6, pp. 23253-23260, 2018, doi: 10.1109/ACCESS.2017.2776930.

[6] U. U. Acikalin, B. Bardak and M. Kutlu, ”Turkish Sentiment Analysis Using BERT,” 2020 28th Signal Processing and Com-munications Applications Conference (SIU), 2020, pp. 1-4, doi: 10.1109/SIU49456.2020.9302492.

[7] A. M. Abirami and V. Gayathri, ”A survey on sentiment analysis methods and approach,” 2016 Eighth International Conference on Advanced Com-puting (ICoAC), 2017, pp. 72-76, doi: 10.1109/ICoAC.2017.7951748.

[8] M. Kanakaraj and R. M. R. Guddeti, ”Performance analysis of Ensemble methods on Twitter sentiment analysis using NLP tech-niques,” Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015), 2015, pp. 169-170, doi: 10.1109/ICOSC.2015.7050801.

[9] L. Kurniasari and A. Setyanto, “Sentiment Analysis using Recurrent Neural Network,” in Journal of Physics: Conference Series, Mar. 2020, vol. 1471,no. 1, p. 012018, doi: 10.1088/1742-6596/1471/1/012018.

[10] Ribeiro, M. T., Singh, S., Guestrin, C. (2016, August). ” Why should i trust you?” Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135–1144).