INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924

TEXT CLASSIFICATION INTO EMOTIONAL STATES USING DEEP LEARNING BASED BERT TECHNIQUE

Yedukondalu Annangi[*1], **Narasimha Reddy Boddapati**[*2], **Mr. S.Kanthi Kiran** [*3]

[*1] Gayatri Vidya Parishad College of Engineering (Autonomous), India,
yedukondaluannangi@gmail.com

[*2] Gayatri Vidya Parishad College of Engineering (Autonomous), India,
narasimhareddyboddapati@gmail.com

[*3] B.Tech., M.Tech., (Ph.D)
Associate Professor
s.kanthi.kiran@gvpce.ac.in

Abstract:

Poetry and formal texts have gotten less attention in recent years from experts in artificial intelligence than informal textual content such as SMS, email, chat, and online user reviews. Using Deep Learning, this study proposes a text-based emotional state categorization system. The text corpus is used to construct a (BERT) Bidirectional Encoder Representations from Transformers model. There are a number of distinct emotional states that can be classified from the text using the suggested method. These states include neutral, joy, fear, sadness.

Keywords: Deep learning, Emotion recognition, BERT, Formal text, Emotional states.

1. Introduction

There are a huge number of comment texts on the Web due to the fast growth of Internet technology and social networks in recent years. When it comes to analysing the emotional tendencies of remarks, artificial intelligence technologies [1] can aid. As a component of artificial intelligence, sentiment analysis is a highly useful tool for determining the sentiment trend of comments. Words have varied contributions to categorization in sentiment analysis.

It is the technique of automatically extracting a writer's insights and characterising them by means of relation: neutra, positive, and negative sentiment analysis. Moreover, Emotion analysis aims to identify the emotions conveyed in the text. As a result of the wider range of classes and their more nuanced distinctions, this activity is typically more complex than sentiment analysis. Despite the fact that lexicon-based [2] and learning-based techniques [3] have both been used to tackle similar problems in the past, the latter have performed better when it comes to categorization. As a result, subsequent studies have concentrated on massive deep learning models [4]. This type of model needs vast amounts of labelled data to be built before it can be used to train it correctly.

A smaller dataset has been utilised to fine tune pre-trained models, which are now extensively employed. Many word embedding layers which were pre-trained (e.g. GloVe [5]) and neural architectures which are task-specific have been suggested for neural networks, however the enhancements of these models were evaluated based on the accuracy of the model6]. In any case, Transformer-based designs [7] have recently proven that there is still space for development.

Experts in disciplines such as processing of natural languages, computational linguistics, and artificial intelligence have been interested in the categorization of views, feelings, and emotional states. It is possible for a machine to evaluate formal and casual writing [8]. While the formal textual content consists of novels, plays, essays, poetry and official/legal documentation, the textual content which are informal comprises messages, and posts from social media [9], among other things.

In text categorization (also known as text classification), free-texts are assigned to predetermined categories. He or she can give conceptual perspectives of document collections, and they may use this in the actual world. For example, reports in health care organisations are commonly categorised using taxonomies of illness categories, surgical procedures, insurance reimbursement codes, etc.

In this work, we looked at how well BERT[6], one of the language models based which was most popular presently, was pre-trained on Transformers .We used this to perform sentiment analysis and emotion identification tasks.

The rest of the article is arranged in the following pattern : related work to the proposed methodology is mentioned in second section ; Third section elaborates about the proposed methodology; In section IV, results and discussions of analyses are

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924

presented, and recommendation for the future work by means of its possibility were concluded in section V.

2. Related Works

A lot of academic researchers have been working on "Emotion recognition" using machine learning approaches in recent years. Using multiple emotion categories to categorise poetry texts, [1] created a technique for recognising emotional states from poetry texts.For the proposed system, They used machine learning classifiers like Naive Bayes. Although some poetry is miss-classified, most of the results were promising.

Different linguistic dimensions such as sarcasm, irony, and metaphor, were analysed by using tweets [11] and by applying multiple ML classifiers like Decision Tree and Naïve Bayes.A scale of 11-points was used to perform the comparison and results were up to the mark. Similar to work [1], in which they face an issue of dataset with limited size. These two studies' aim was to expand the dataset with adequate data by overcoming this limitation in the future.

SVM and Naive Bayes algorithms were used [12] to categorize poetry with respect to their category of emotion. The results of a tiny dataset were encouraging. Incorporating phonemic elements might enhance the system's versatility.

An audio input system was proposed [13]. As it is possible to determine a person's sentiments about music, based on their moods and topics. A support vector machine classifier is used to achieve a better outcome in this regard. Despite this, mood classification can be refined. The automated analysis of poetry in Arabic, Chinese, German, Malay, Persian, and other languages has been done in the past. According to [14] Arabic poetry is divided into several kinds such as Heja, Fakhr, Retha, and Ghazal. Machine learning techniques were used to classify Arabic poetry according to the categories listed above.

SVM (Support Vector Machine) and Ss-CNN (Super Skinny- Convolutional Neural Network) algorithms are used [15] to do sentiment analysis on German novels at the sentence level, and the results are impressive. The SVM classifier outperformed the Ss-CNN classifier in terms of accuracy. A supervised ML method was suggested in [16]. In experiments, a variety of classifiers were employed. A collection of emotions, on the other hand, can be improved in the future in order to achieve more hopeful outcomes. Table 1 describes the various works related to Bert and Emotion Classification.

Author & Year	Proposed	Finding/Outcomes
P. S. Sreeja and G. S. Mahalakshmi	Emotion recognition from poems by maximum posterior probability	Recognition of emotions from poem texts using the probability
A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, J. Barnden, and A. Reyes, 2015	Sentiment analysis of figurative language in Twitter	Analysis of sentiment from the Twitter data
G. Rakshit, A. Ghosh, P. Bhattacharyya, and G. Haffari, 2015	Automated analysis of Bangla poetry for classification and poet identification	Classification and analysis of Bangla poetry
K. Bischoff, C. S. Firan, R. Paiu, W. Nejdl, C. Laurier, and M. Sordo, 2009	Music mood and theme classification-a hybrid approach	Classification of theme and mood of the music
O. Alsharif, D. Alshamaa, and N. Ghneim, 2013	Emotion classification in Arabic poetry using machine learning	Using machine learning for the classification of emotions from Arabic poetry

Table 1: Related Works Summary

3. Methodology

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

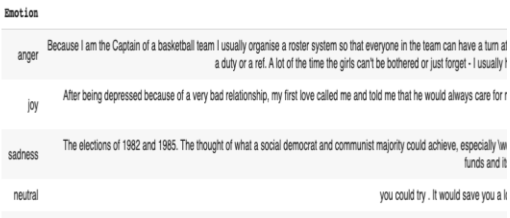
IJARCSET

ISSN 2454-9924

In this section, The procedure of developing our system is clearly described.

A. Data Collection

For our suggested system, we'll use text data to classify emotions. The initial step in text classification into emotion states is creating a dataset, which contains a significant number of pairs with emotion state labels and its respective text content as shown in Fig.1.



Emotion	
anger	Because I am the Captain of a basketball team I usually organise a roster system so that everyone in the team can have a turn at a duty or a ref. A lot of the time the girls can't be bothered or just forget - I usually
joy	After being depressed because of a very bad relationship, my first love called me and told me that he would always care for
sadness	The elections of 1982 and 1985. The thought of what a social democrat and communist majority could achieve, especially funds and it
neutral	you could try . It would save you a

Fig 1. Sample data in the dataset

B. Train-Test Split

After dataset creation, the dataset is divided into two parts for training and validating the model respectively. Approximately, 70 percent of the total dataset is allocated for the training data set. On the other hand, the test dataset accounted for the remaining 30 percent. This Split was clearly mentioned in Fig 2.

```
size of training set: 7934
size of validation set: 3393
joy        2326
sadness    2317
anger      2259
neutral    2254
fear       2171
Name: Emotion, dtype: int64
```

Fig 2. Metadata about dataset

C. Text Pre-processing

Emotion state labels were stored into classes and later encoded with unique ids, which will ease the classification of the emotion states. Pandas is the library, which was used to read the CSV file as a data frame. This encoding was described in table 2.

Emotion	Embedded Id
Joy	0
Sadness	1
Fear	2
Anger	3
Neutral	4

Table 2: Emotion Classification

D. Model Creation

In this proposed system, a deep learning based BERT model is used to perform the classification of emotion states with respect to its text content. At first, this model will be built by feeding the dataset. Once after the completion of the training, the model will be stored.

The BERT model was imported from ktrain library, which is a light weight wrapper of TensorFlow and keras.

We trained this model with a learning rate of 2e-5 and iterated till 3 epochs for better performance of the model. Iteration of the model until the epoch was mentioned in the below screenshot.

```
begin training using onecycle policy with max lr of 2e-05...
Epoch 1/3
1323/1323 [==============================] - 1574s 1s/step - loss: 0.9515 - accuracy: 0.6389 - val_loss: 0.5662 - val_acc
Epoch 2/3
1323/1323 [==============================] - 1540s 1s/step - loss: 0.4608 - accuracy: 0.8404 - val_loss: 0.5116 - val_acc
Epoch 3/3
1323/1323 [==============================] - 1536s 1s/step - loss: 0.2015 - accuracy: 0.9386 - val_loss: 0.5550 - val_acc
<keras.callbacks.History at 0x7f813ef64c90>
```

Fig 3. Model training

E. Emotion Prediction

The saved model will be utilized for evaluation, and the prediction of anticipated emotions is shown on the model's plotter. It predicts the emotion state of test data.

This whole process of creating the model and predicting the output is clearly in the flowchart in Fig 4.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924

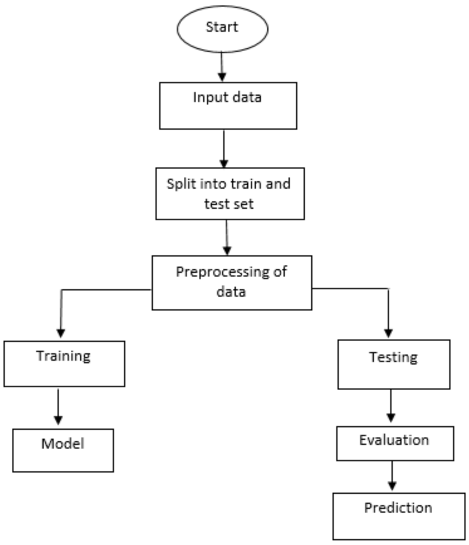


Fig 4. Proposed method block diagram

BERT :

Transformers is an attention mechanism used by BERT models that understand contextual connections among the words or the sub-words in a given textual content. There are two distinct processes in its basic version, an encoder, to read the textual content and a decoder that generates a forecast for the job. Since BERT's objective is to build a language model, just the mechanism of encoder is essential.

In contrast to directional models, which take the input sequentially in a text format (either by starting from left and moving towards right or the reverse), the Transformer encoder takes all the input at once with the full sequence of the words at only once. As a result, it's classified as bidirectional, while the correct term can be called non-directional. As a result, This feature of being bidirectional enables the model to understand the connection between the word and the context based on all of its surroundings or adjacent words (either left or right of that word). Unlike traditional RNN, LSTM and Bi- LSTM, BERT enables parallel processing and eradicates gradient vanishing and exploding problems.

There are two types of BERT models.

1. BERT BASE
2. BERT LARGE

BASE of BERT contains 12 encoder layers and 786 hidden layers (feed forward neural networks) as shown in the fig 5 whereas BERT LARGE contains 24 encoder layers and 1024 hidden (feed forward neural networks). In addition to this, BERT BASE contains 12 attention heads however BERT LARGE contains 16 attention heads. The model implemented in this proposed system is BERT BASE.

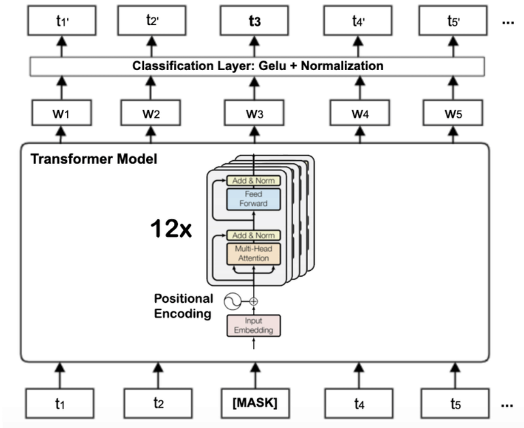


Fig 5. Architecture of BERT BASE

Encoder:

Encoder performs input embedding, position encoding and sends support vectors to feed forward layers and multi-head attention layers. The encoder for the Transformer is depicted in Fig 6 at a high level.

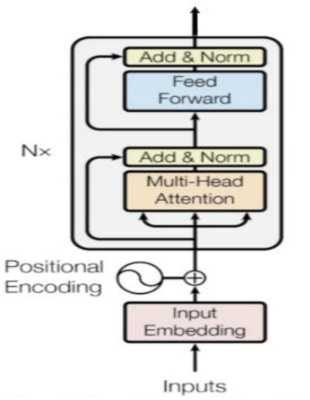


Fig 6. Encoder Architecture

Input Embedding:

Tokens are used as input, and they are first embedded into vectors that give meaning for the word. Fig 7 shows architecture of token, sentence and position embedding of input.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES
ISSN 2454-9924

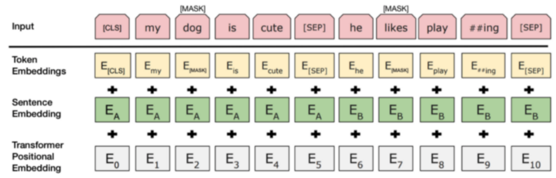


Fig 7. Input, segment and position embedding architecture

Position Encoding:

After input embedding, position encoding will be implemented on these vectors to extract the context of the word by means of its position in the sentence.

Normalization:

Normalization eradicates the internal covariate shift problem. Covariate shift occurring within a neural network is called internal covariate shift. When the weights are being continuously updated, the output distribution varies in a few network layers. This pushes the layers at a high level to adjust the changes. As a result, the learning rate becomes low. This can be avoided by normalization of the inputs.

Multi-Head Attention:

Subsequently, vectors obtained from the position encoding will be given as input to a multi-head attention layer to create attention vectors.

Feed Forward Neural Networks:

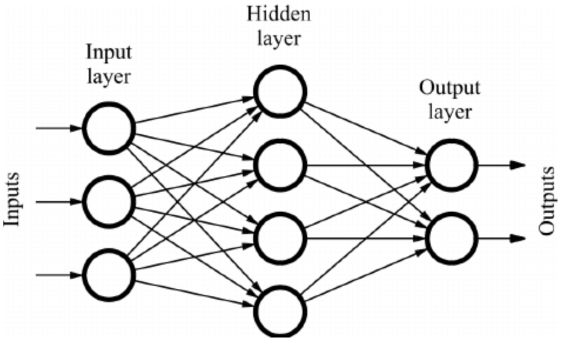


Fig 8. Feed forward neural network

Later these vectors will be processed by a feed forward neural network in order to extract meaningful information about the sequence of tokens given. An input token will be returned as a list of H-dimensional vectors shown in fig 8, each corresponding to the same index. These encoders are stacked up to form a transformer layer.

NSP (Next Sentence Prediction)

During the model creation, it takes input as pairs of sentences and predicts whether the later sentence will be the pair of the first one. During this process, Half of the inputs are pairs in which the 2nd sentence is the next sentence from the original text, on the other hand , the other half are sentences randomly selected as the second sentence from the corpus . Because of this, it's assumed that the second statement will be unrelated to the first.

Before it enters the model, the input is processed as follows to assist the model tell the difference between the two sentences:

- The starting of the first sentence has a [CLS] token and each subsequent sentence has a [SEP] token at the conclusion.
- Every token has an embedded sentence denoting if it belongs to the First Sentence or the Second Sentence .
- Each token has a positional embedding to identify where it belongs to the sequence.

To determine if the second statement is truly linked to the first, the following procedures are taken:

- The Transformer model processes all input data.
- The transformation of [CLS] token's output into a 21 shaped vector was done by a basic classification layer
- Using Softmax to calculate the likelihood of IsNextSequence.

When learning the BERT model, Masked LM and Next Sentence Prediction are coupled in order to minimize the combined loss function of the two methods.

Encoding Layers of the Implemented BERT BASE Model:

Text received from the input layers undergo token embedding, segment embedding and position embedding sequentially as shown in the fig 9, then normalization of input normalization takes place to avoid Internal covariate shift.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924

Layer (type)	Output Shape	Param #	Connected to
Input-Token (InputLayer)	[(None, 350)]	0	
Input-Segment (InputLayer)	[(None, 350)]	0	
Embedding-Token (TokenEmbedding	[(None, 350, 768), (	23440896	Input-Token[0][0]
Embedding-Segment (Embedding)	(None, 350, 768)	1536	Input-Segment[0][0]
Embedding-Token-Segment (Add)	(None, 350, 768)	0	Embedding-Token[0][0] Embedding-Segment[0][0]
Embedding-Position (PositionEmb	(None, 350, 768)	268800	Embedding-Token-Segment[0][0]
Embedding-Dropout (Dropout)	(None, 350, 768)	0	Embedding-Position[0][0]
Embedding-Norm (LayerNormalizat	(None, 350, 768)	1536	Embedding-Dropout[0][0]

Fig 9. Input layers and embedding's

There are 12 encoder layers in BERT BASE and each encoder layer contains 4 multi head attention layers, 4 feed forward layers and 768 hidden layers as shown in Fig 10 and Fig 11 show the continuation of encoder layer 1 to 12.

Encoder-1-MultiHeadSelfAttentio	(None, 350, 768)	2362368	Embedding-Norm[0][0]
Encoder-1-MultiHeadSelfAttentio	(None, 350, 768)	0	Encoder-1-MultiHeadSelfAttention[
Encoder-1-MultiHeadSelfAttentio	(None, 350, 768)	0	Embedding-Norm[0][0] Encoder-1-MultiHeadSelfAttention-
Encoder-1-MultiHeadSelfAttentio	(None, 350, 768)	1536	Encoder-1-MultiHeadSelfAttention-
Encoder-1-FeedForward (FeedForw	(None, 350, 768)	4722432	Encoder-1-MultiHeadSelfAttention-
Encoder-1-FeedForward-Dropout (	(None, 350, 768)	0	Encoder-1-FeedForward[0][0]
Encoder-1-FeedForward-Add (Add)	(None, 350, 768)	0	Encoder-1-MultiHeadSelfAttention- Encoder-1-FeedForward-Dropout[0][
Encoder-1-FeedForward-Norm (Lay	(None, 350, 768)	1536	Encoder-1-FeedForward-Add[0][0]

Fig 10. Encoder 1 layer

Encoder-12-MultiHeadSelfAttenti	(None, 350, 768)	2362368	Encoder-11-FeedForward-Norm[0][0]
Encoder-12-MultiHeadSelfAttenti	(None, 350, 768)	0	Encoder-12-MultiHeadSelfAttention
Encoder-12-MultiHeadSelfAttenti	(None, 350, 768)	0	Encoder-11-FeedForward-Norm[0][0] Encoder-12-MultiHeadSelfAttention
Encoder-12-MultiHeadSelfAttenti	(None, 350, 768)	1536	Encoder-12-MultiHeadSelfAttention
Encoder-12-FeedForward (FeedFor	(None, 350, 768)	4722432	Encoder-12-MultiHeadSelfAttention
Encoder-12-FeedForward-Dropout	(None, 350, 768)	0	Encoder-12-FeedForward[0][0]
Encoder-12-FeedForward-Add (Add	(None, 350, 768)	0	Encoder-12-MultiHeadSelfAttention Encoder-12-FeedForward-Dropout[0]
Encoder-12-FeedForward-Norm (La	(None, 350, 768)	1536	Encoder-12-FeedForward-Add[0][0]

Fig 11. Encoder 12 layer

In the output layer that is shown in Fig 12, next sentence predictions are generated. In this manner, the model will be trained.

Extract (Extract)	(None, 768)	0	Encoder-12-FeedForward-Norm[0][0]
NSP-Dense (Dense)	(None, 768)	590592	Extract[0][0]
dense (Dense)	(None, 5)	3845	NSP-Dense[0][0]

```
Total params: 109,361,669
Trainable params: 109,361,669
Non-trainable params: 0
```

Fig 12. Output layer

4. Results and Discussions

We will discuss the evaluations or predictions that were achieved by using the approach described above and how they were assessed.

Predicted Output:

```
message: " Very funny . What's wrong with you today ? You are my secretary
 and you are not supposed to talk to me in that tone of voice . Do you know that ?"

Predicted Emotion: anger
Predicted Time:(0.19)
------------------------------------------------------
message: "I was in the house at night and I heard a hyena crying outside.
 It cried for almost an hour and I feared it might break the window and enter inside the house."

Predicted Emotion: fear
Predicted Time:(0.15)
------------------------------------------------------
message: "He himself was thrilled , but his pleasure was diminished as
 he imagined Ken 's pallid , bandaged head shadowed on the screen ."

Predicted Emotion: joy
Predicted Time:(0.14)
------------------------------------------------------
message: "I hear many teenagers will spend hours and hours sitting at computers .
 And they don't care about their own health at all . How about you ?"

Predicted Emotion: neutral
Predicted Time:(0.14)
------------------------------------------------------
message: "But for the time being , Private Morrison 's family ,
 already overcome by grief , can only wait and hope his body will be released to them soon ."

Predicted Emotion: sadness
Predicted Time:(0.19)
------------------------------------------------------
```

Fig 13. Predicted Output

We predicted the emotion related to a text by using the model that we created and stored into h5 format. The performance of our model is impressive with predicted time for each prediction less than 0.20seconds. The predicted values for the given text can be shown in Fig 13.

Fig14 depicts validation matrix values that describe the precision, recall, f1-score and support vector values of each emotion.

	precision	recall	f1-score	support
joy	0.85	0.84	0.85	707
sadness	0.82	0.80	0.81	676
fear	0.86	0.86	0.86	679
anger	0.80	0.80	0.80	693
neutral	0.80	0.84	0.82	638
accuracy			0.83	3393
macro avg	0.83	0.83	0.83	3393
weighted avg	0.83	0.83	0.83	3393

Fig 14. Validation Matrix

After completing the training part, we plot accuracy and loss plots that will be used to evaluate our model. There is a steady increase in accuracy of the model with the epochs. Accuracy of the training data set is quite impressive which was peaking at 93% by the end of 3rd epoch. On the other hand, testing over the Validation dataset showed optimistic results with an accuracy of above 80 percent.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924

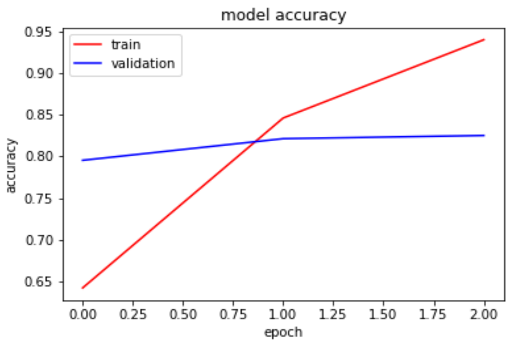


Fig 15. Model accuracy plot

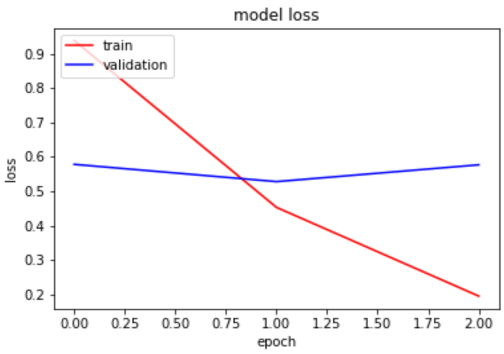


Fig 16. Model loss plot

Loss of the proposed model declined dramatically in the third epoch for the training data and showed optimistic loss while validating the model.

5. Conclusion

This study provides the classification of emotions from the text using the deep learning technique BERT. That classifies the 5 types of emotions from the text. BERT model is used for training the data. From the comparison with the previously existing methods, our proposed method gave better accuracy.

References

[1] P. S. Sreeja and G. S. Mahalakshmi, ‘‘Emotion recognition from poems by maximum posterior probability,’’ Int. J. Comput. Sci. Inf. Secur., vol. 14, pp. 36–43, 2016.

[2] C. Diamantini, A. Mircoli, D. Potena, and E. Storti. 2015. Semantic disambiguation in a social information discovery system. In 2015 International Conference on Collaboration Technologies and Systems, CTS 2015. 326–333.

[3] Md. 2018. Sentiment analysis of Arabic Tweets using deep learning. Procedia Computer Science 142 (2018), 114–122.

[4] Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8, 4 (2018), e1253.

[5] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove:

Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 1532–1543.

[6] Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019. Exploiting BERT for end-to-end aspect-based sentiment analysis. ArXiv preprint arXiv: 1910.00883 (2019).

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. ArXiv preprint arXiv: 1706.03762 (2017).

[8] J. Kaur and J. R. Saini, ‘‘Punjabi poetry classification: The test of 10 machine learning algorithms,’’ in Proc. 9th Int. Conf. Mach. Learn. Comput. (ICMLC), 2017, pp. 1–5.

[9] G. Mohanty and P. Mishra, ‘‘Sad or glad? Corpus creation for Odia poetry with sentiment polarity information,’’ in Proc. 19th Int. Conf. Comput. Linguistics Intell. Text Process. (CICLing), Hanoi, Vietnam, 2018.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/ N19-1423

[11] A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, J. Barnden, and A. Reyes, ‘‘SemEval-2015 task 11: Sentiment analysis of figurative language in Twitter,’’ in Proc. 9th Int. Workshop Semantic Eval. (SemEval), 2015, pp. 470–478.

[12] G. Rakshit, A. Ghosh, P. Bhattacharyya, and G. Haffari, ‘‘Automated analysis of Bangla poetry for classification and poet identification,’’ in Proc. 12th Int. Conf. Natural Lang. Process., Dec. 2015, pp. 247–253.

[13] K. Bischoff, C. S. Firan, R. Paiu, W. Nejdl, C. Laurier, and M. Sordo, ‘‘Music mood and theme

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924

classification-a hybrid approach,'' in Proc. ISMIR, Oct. 2009, pp. 657–662.

[14] O. Alsharif, D. Alshamaa, and N. Ghneim, ''Emotion classification in Arabic poetry using machine learning,'' Int. J. Comput. Appl., vol. 65, p. 16, May 2013.

[15] A. Zehe, M. Becker, F. Jannidis, and A. Hotho, ''towards sentiment analysis on German literature,'' in Proc. Joint German/Austrian Conf. Artif. Intell. Cham, Switzerland: Springer, 2017, pp. 387–394.

[16] L. Barros, P. Rodriguez, and A. Ortigosa, ''Automatic classification of literature pieces by emotion detection: A study on Quevedo's poetry,'' in Proc. Humaine Assoc. Conf. Affect. Comput. Intell. Interact, Sep. 2013, pp. 141–146.

[17] W. Li and H. Xu, ''Text-based emotion classification using emotion cause extraction,'' Expert Syst. Appl., vol. 41, no. 4, pp. 1742–1749, Mar. 2014.

[18] TensorFlow Text Classification Using Attention Mechanism. Accessed: Jan. 25, 2020. [Online].

Available: https://androidkt.com/tensorflow-text classification-attention-mechanism/

[19] Keras Documentation: Embedding. Accessed: Jan. 2, 2020. [Online].

Available: https://keras.io/layers/embeddings/

[20] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, ''Dimensional sentiment analysis using a regional CNN-LSTM model,'' in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, 2016, pp. 225–230.