IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SENTIMENT ANALYSIS USING DEEP NEURAL NETWORK ON MOVIE REVIEWS

P.Likitha[1], P.Bhavya[2], S.Sravani[3], P.Lahari[4], M.Pallavi[5]

1Student, 2Student, 3Student, 4Student, 5Assistant Professor

1Vignan's Institute of Engineering for Women,

2Vignan's Institute of Engineering for Women,

3Vignan's Institute of Engineering for Women,

4Vignan's Institute of Engineering for Women,

5Vignan's Institute of Engineering for Women

Abstract:

In this era of social networking, sentiment analysis became a challenging task due to an abundance of opinions and words in the same comment. Measuring the emotional tendency and taking specific measures is useful to satisfy the public. Artificial Intelligence technology made it easy to calculate the sentiment of the comment. There are many Neural Networks to process this task. But in this project, we have used a hybrid Deep CNN-BiLSTM Neural Network model for obtaining better comment vectors. Distributed word representations are widely used for sentiment classification. Only semantic information of words are considered in word embedding‘s.

We have proposed the traditional TFIDF algorithm in addition to Word2Vec embedding in order to extract the features of the word to create a weighted word vectors. These weighted word vectors are sent as an input to the Convolution Neural Networks (CNN) along with the max pooling layer. The output generated is then transferred to the Bi-LSTM neural network in order to obtain the accurate results of sentiment classification. This Word2Vec and TF-IDF with hybrid CNN-BiLSTM model is applied on the Stanford IMDB movie reviews dataset and it proved to be productive with high accuracy. It proposed based on a hybrid CNN-BILSTM Deep Neural Network model.

Keywords: Word2Vec, hybrid CNN-BILSTM Deep Neural Network, TFI

INTRODUCTION

Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces. The term Deep Learning was introduced to the machine learning community by Rina Dechter in 1986, and to artificial neural networks by Igor Aizenberg and colleagues in 2000, in the context of Boolean threshold neurons.

The first general, working learning algorithm for supervised, deep, feed forward, multilayer perceptrons was published by Alexey Ivakhnenko and Lapa in 1967. A 1971 paper described already a deep network with 8 layers trained by the group method of data handling algorithm.

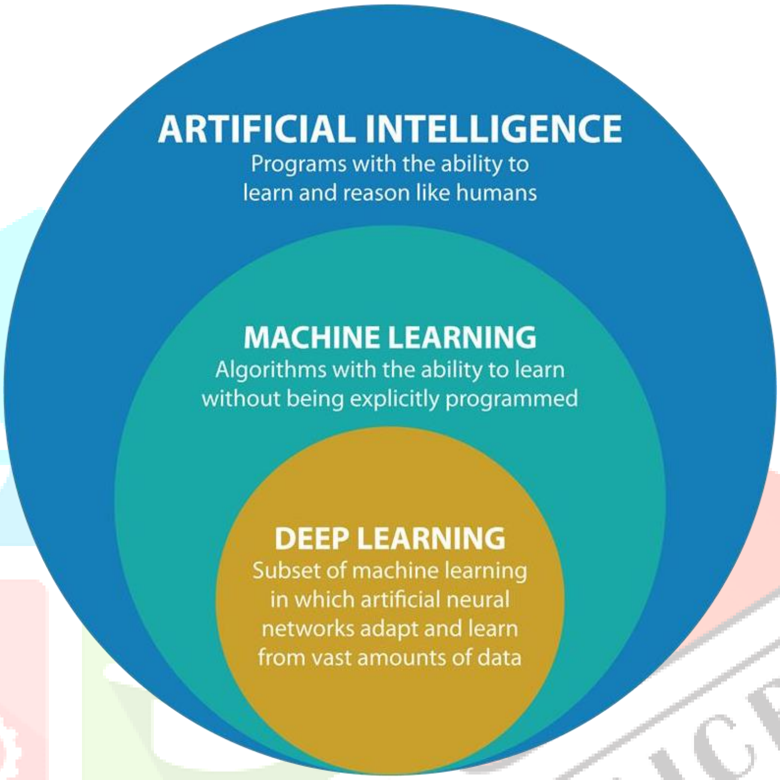

Figure: Deep Learning

PROBLEM STATEMENT

Users have written over hundreds of reviews for each movie. The reviews are expressed in the natural language, along with a self-annotated score describing the overall sentiment of that review. To make a better informed decision, user has to go through each of them, which is a time consuming activity that user is highly unlikely to invest time in.

LITERATURE SURVEY

Review Literature of Sentiment Analysis of Online Papers:

Sentiment analysis or opinion mining is used to automate the detection of subjective information such as opinions, attitudes, emotions, and feelings. "Sentiment Analysis of Online Papers" (SAOOP). SAOOP aims at supporting researchers and saving their time and efforts by enabling them to report the total evaluation for the papers. SAOOP includes main two evaluations for each research paper: Sentiment score and System score. Sentiment score which is an evaluation for the paper based on analyzing online sentiment reviews. System score is an evaluation for the paper based on topic domain parameters. SAOOP employs several techniques including natural language processing, text analysis and opinion mining in the sentiment analysis evaluation process. SAOOP is a new technique that introduces an enhancement for the bag-of-words (BOW) model in sentiment analysis. It improves accuracy and solves several sentiment evaluation challenges.

Review Literature of Sentiment Analysis of Comment Texts based on BiLSTM:

With the rapid development of Internet technology and social networks, a large number of comment texts are generated on the Web. In the era of big data, mining the emotional tendency of comments through artificial intelligence technology is helpful for the timely understanding of network public opinion. The technology of sentiment analysis is a part of artificial intelligence, and its research is very meaningful for obtaining the sentiment trend of the comments. The essence of sentiment analysis is the text classification task, and different words have different contributions to classification. In the current sentiment analysis studies, distributed word representation is mostly used. However, distributed word representation only considers the semantic information of word, but ignore the sentiment information of the word. In this paper, an improved word representation method is proposed, which integrates the contribution of sentiment information into the traditional TF-IDF algorithm and generates weighted word vectors. The weighted word vectors are input into bidirectional long short term memory (BiLSTM) to capture the context information effectively, and the comment vectors are better represented.

. EXISTING SYSTEMS

The previous research on Twitter Sentiment Analysis was implemented to analyze the lexicon features of the tweets by using Machine Learning algorithms such as SVM. This approach ended by assigning sentiments with an accuracy of 80%. The second approach by Xi Ouyang and Pan Zhou on Sentiment Analysis was proposed by processing it using Word2Vec .The framework used is CNN. Normalization and dropout are also used to increase the accuracy. The test accuracy obtained was 45.4% for the movie review dataset. The other approach by Subarno Pal, reported the work on Sentiment analysis of the movie reviews. This work mainly focused on recurrent neural networks. Different models were used and the accuracy was noted. Firstly they implemented it using Conventional LSTM and obtained an accuracy of 80.92%. Next they have used deep LSTM, where the accuracy had been increased by 1%. Finally they concluded their work by using Bidirectional LSTM and the accuracy obtained was 83.83%. The researchers of the fourth paper proposed their work on Sentiment Classification of the movie reviews dataset which contains 44,617 reviews. Word2Vec was used to create the word embedding‘s. They have used the hybrid CNN-LSTM neural network and have tested the model over 4000 reviews and stated that the model yields best results in terms of accuracy and runtime.

PROPOSED SYSTEM

As mentioned earlier, we tend to propose the model based on Deep Neural Networks on the ensemble of Convolution Neural Networks (CNN) and Bidirectional – Long Short Term Memory(Bi-LSTM). Using the hybrid model helps us to enhance the accuracy over a single model. To cypher the model first of all the dataset is loaded and is processed using Natural Language Processing techniques. Then these words are embedded using different word representations and also the weighted word vector has been passed as an input to a Convolution Neural Network Layer, thereby Max Pooling has been applied. Next, the Bidirectional-LSTM layer has been processed and also he output is obtained.

5.1 Preprocessing

Preprocessing is an essential step in Natural Language Processing. It converts the text into chunks and machine readable form in order to make algorithms perform better. There are different steps in preprocessing:

Tokenization: Tokenization is the process of splitting the sentence into words referred to as tokens. The sentence is divided into tokens whenever there‘s a whitespace or any other punctuations. A paragraph can also be converted into a sentence by tokenizing it whenever there is a full stop. There are several tools to implement tokenization suck as Natural Language Toolkit (NLTK), TextBlob, Spacy, Gensim etc.

Stop Word Removal: The most commonly used words (―the‖, ―a‖, ―is‖) in a language are known as Stop Words. These words do not affect the meaning of the sentence. Removal of these words helps in minimizing the data. Using NLTK we can remove stop words.

Normalization: Normalization is the process of removing punctuations, html tags, etc. Normalization also converts the uppercase characters to lowercase.

Stemming: Stemming is the process of removing the affixes. Stemming produces the morphological variants of the base word and reduces the words. For English language, stemming can be done either by Porter Stammer or Lancaster Stammer. The tokenized words are passed as the input for the stemmer.

Lemmatization: The canonical form of the base word is known as Lemma. In lemmatization all the words are converted into their respective lemmas. NLTK provides Word Net Lemmatizer which checks the lemmas using Word Net Database.

5.2 Word Embedding:

Most popular representation of document vocabulary is known as word embedding‘s. In word embedding‘s, words are represented in the form of hot coded vectors. Word Embedding‘s capture the context of the word and they find the semantic and syntactic similarity of the words. Word2Vec, Doc2Vec, GloVe are some of the techniques to learn word embedding‘s using Neural Networks. In word embedding‘s one hot coded vector of length _N‘ is created for each word. Every word is represented by 1 at its index position and zeros at remaining positions of a vector. Vector is represented in the form V= [w1, w2,..,wN].These embedding‘s are visualized in a N-Dimensional space, where each dimension is occupied by a single word. The main objective of word embedding is similar words have to occupy close spatial positions. Mathematically the angle between the vectors must be close to 0 and the cosine value must be equal to 1. Word2Vec embedding can be obtained using two methods: Skip Gram and Continuous Bag of Words (CBOW).

CBOW Model: This method predicts the words from a given context. The input is one hot encoded vector of length _N‘. This input is mapped to a hidden layer that contains _P‘ hidden neurons and the output is valued layer of size _N‘.

Skip Gram Model: Skip Gram model is an exact flip of CBOW model where the target word is taken as input and the probability distributions of the input word are obtained as the output of the model. CBOW is applied to determine more frequent words and is considered as the fastest technique. Skip Gram technique is more accurate for small amount of data

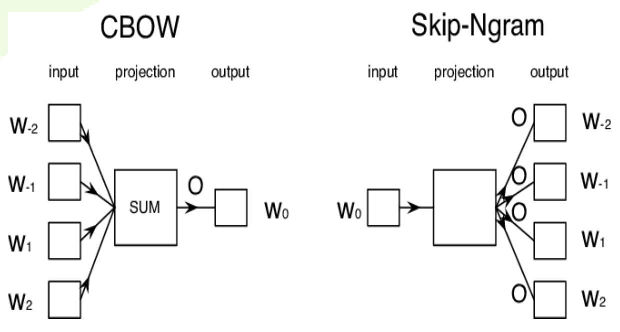

Figure : Model of CBOW and Skip-Ngram

Feature Extraction

Term frequency- inverse document frequency (TF used as a weighing factor in text mining. It reflects the importance of a word in a corpus. Term frequency is the number of times each term occurs in a document. Term frequency increases with the occurrence of the term in a corpus. Inverse document frequency decrements the count of frequently used terms in a document. TFmathematically computed as the product of term frequency and inverse document frequency. If a term occurs frequently in a document then TF-IDF to be zero. For term i in document j , weight is calculated as

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$

n_{in}: number of input features
n_{out}: number of output features
k: convolution kernel size
p: convolution padding size
s: convolution stride size

Convolution Neural Network (CNN) is a deep neural network with feed- forward technique and is the building block of any neural network. CNN‘s are generally applied for image processing tasks and now a days it is being widely used for various other tasks such as sentence or text classification. CNN produces accurate results in feature extraction. To extract features CNN uses two techniques: convolution and pooling. Convolution Neural Network is a multilayer neural network where the outputs of the sequenced layers are fully connected.

Max Pooling Layer

Convolution layers apply filters to the embedded vectors in order to extract features. The results of the convolution layers are precise and they give the exact position of the features. Therefore, small change in the input may result in an error. To address this problem pooling layer is added to the convolution layer after a non-linearity activation has been applied to the output of the convolution layer. The main work of the pooling layer is to order the layers in the neural network after every feature map. Pooling layer is applied on every feature map and a new set of pooled feature maps are created. The size of the feature map must always be greater than the pooling size. Pooling layer always decreases the size of the feature map. Max Pooling calculates the maximum value of the feature map. Max Pooling layer overcomes the drawbacks occurred in the convolution layer when there are slight variations in the input.

Bi-directional Long Short Term Memory (Bi-LSTM)

Bi-directional Long Short Term Memory (Bi-LSTM) is typically a LSTM with an additional feature. Bi-LSTM is used to enhance the performance of the model. Two LSTM‘s are trained on the input data in a Bi-LSTM neural network. A forward layer is applied on one LSTM and a backward layer is applied on the other. Due to this, the performance of the BI-LSTM model is faster. By providing the original copy and the reversed copy of the input to the network, it overcomes to drawback of the LSTM model. The merge mode can be selected depending on the problem. Different merge modes are available in Bi-LSTM model such as sum, multiplication, concatenation, and average. The concatenation operation is done instead of applying time distributed layer. This doubles the output of the model in a single time step.

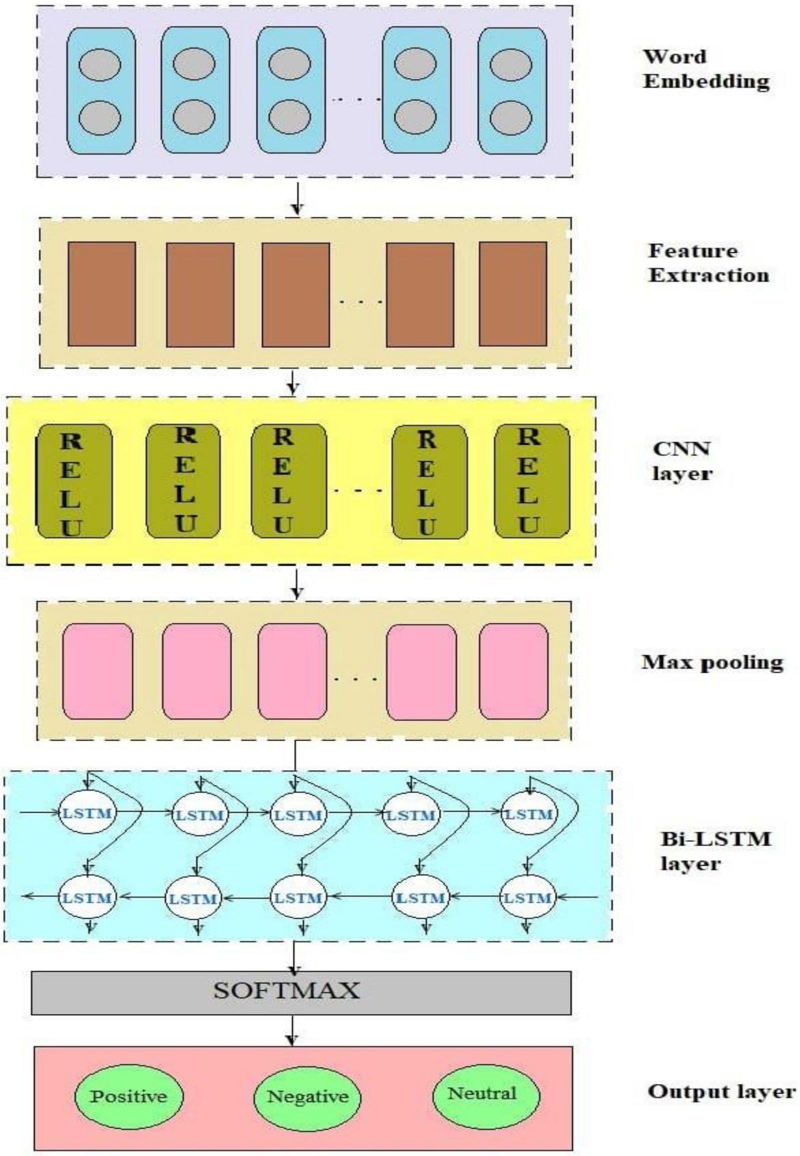

Figure: Proposed model architecture in this paper

VISUALIZATION OF RESULTS

#accuracy of both testing and training data

Train on 44500 samples, validate on 500 samples
Epoch 1/10
44500/44500 [==============================] - 40s 898us/step - loss: 0.5107 - accuracy: 0.7439 - val_loss: 0.3740 - val_accuracy: 0.8160
Epoch 2/10
44500/44500 [==============================] - 37s 827us/step - loss: 0.4164 - accuracy: 0.8096 - val_loss: 0.3647 - val_accuracy: 0.8480
Epoch 3/10
44500/44500 [==============================] - 37s 828us/step - loss: 0.3994 - accuracy: 0.8206 - val_loss: 0.3687 - val_accuracy: 0.8280
Epoch 4/10
44500/44500 [==============================] - 36s 814us/step - loss: 0.3918 - accuracy: 0.8249 - val_loss: 0.3445 - val_accuracy: 0.8460
Epoch 5/10
44500/44500 [==============================] - 37s 821us/step - loss: 0.3897 - accuracy: 0.8254 - val_loss: 0.3468 - val_accuracy: 0.8460
Epoch 6/10

44500/44500 [==============================] - 37s 821us/step - loss: 0.3831 - accuracy: 0.8288 - val_loss: 0.3353 - val_accuracy: 0.8500
Epoch 7/10
44500/44500 [==============================] - 36s 818us/step - loss: 0.3761 - accuracy: 0.8319 - val_loss: 0.3358 - val_accuracy: 0.8540
Epoch 8/10
44500/44500 [==============================] - 36s 820us/step - loss: 0.3762 - accuracy: 0.8322 - val_loss: 0.3168 - val_accuracy: 0.8560
Epoch 9/10
44500/44500 [==============================] - 37s 830us/step - loss: 0.3714 - accuracy: 0.8362 - val_loss: 0.3214 - val_accuracy: 0.8540
Epoch 10/10
44500/44500 [==============================] - 37s 830us/step - loss: 0.3670 - accuracy: 0.8380 - val_loss: 0.3123 - val_accuracy: 0.8600

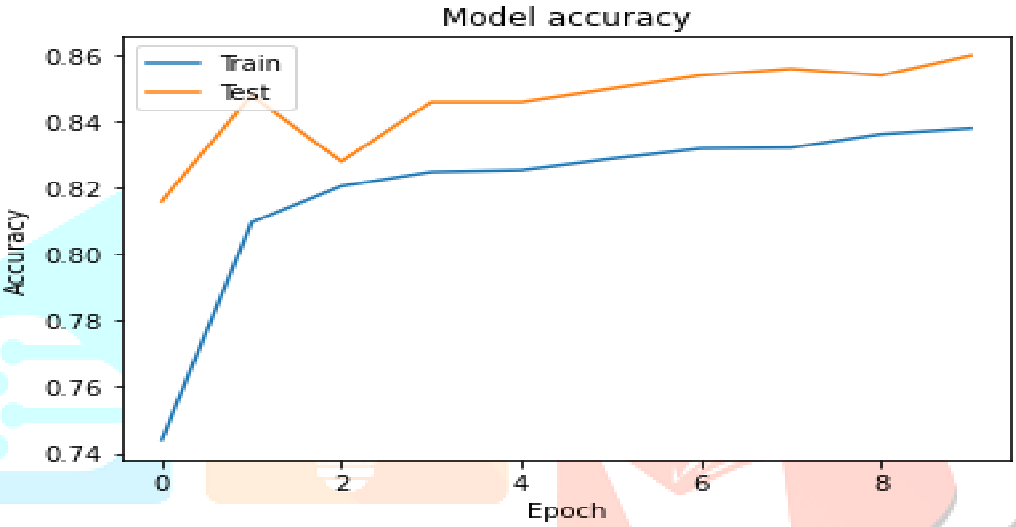

Fig: Training & validation accuracy values

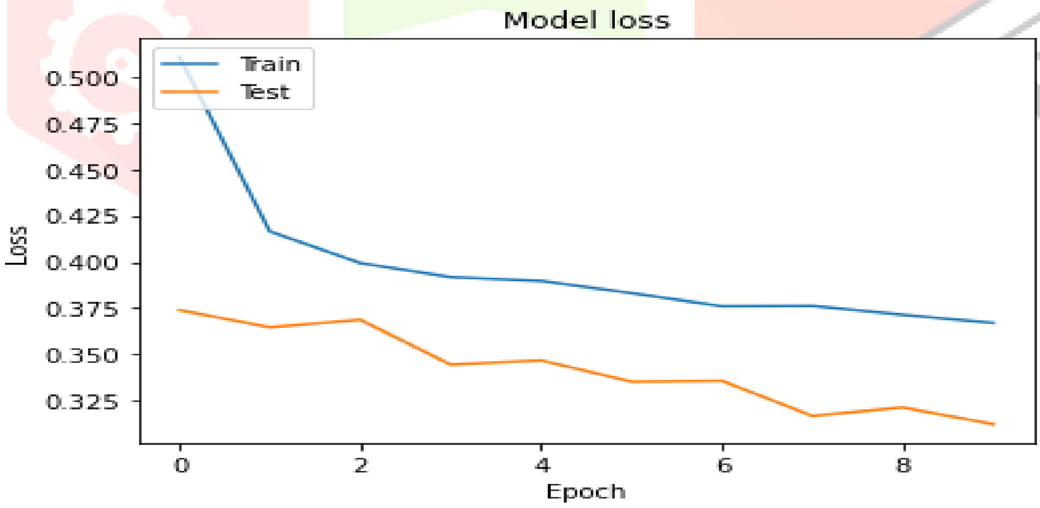

Fig: Training & validation loss values

```
model.summary()
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_5 (Conv1D)            (None, 32, 90)            270
_________________________________________________________________
conv1d_6 (Conv1D)            (None, 32, 90)            24390
_________________________________________________________________
conv1d_7 (Conv1D)            (None, 32, 90)            32490
_________________________________________________________________
conv1d_8 (Conv1D)            (None, 32, 90)            40590
_________________________________________________________________
dropout_4 (Dropout)          (None, 32, 90)            0
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 16, 90)            0
_________________________________________________________________
dropout_5 (Dropout)          (None, 16, 90)            0
_________________________________________________________________
bidirectional_2 (Bidirection (None, 256)               224256
_________________________________________________________________
dropout_6 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 257
=================================================================
Total params: 322,253
Trainable params: 322,253
Non-trainable params: 0
_________________________________________________________________
```

CONCLUSION

IMDB Sentiment classification using deep neural networks comes under the category of opinion mining. It targets on analyzing the sentiments of the author by preprocessing the text. Once the text is preprocessed, Word2Vec algorithm is used for word embedding‘s and it converts the word into a encoded vector which is further used for feature extraction. Using the hybrid model helps us to enhance the accuracy over a single model. The model based on Deep Neural Networks on the ensemble of Convolution Neural Networks (CNN) and Bidirectional – Long Short Term Memory(Bi-LSTM). It will predict the words that is both forward and backword words of the particular expression. Feature extraction is done by TF-IDF algorithm in order to remove the inconsistencies. Further the data is fed into the hybrid CNN_BILSTM model. The hybrid model excelled over the other deep networking models and other machine learning algorithms.

REFERRENCES

1. S. Poria And A. Gelbukh, ―Aspect Extraction For Opinion Mining With A Deep Convolutional Neural Network,‖ Knowledge-Based Syst., Vol. 108, Pp. 42–49, Sep. 2016.
2. K. Kim, M. E. Aminanto, And H. C. Tanuwidjaja, ―Deep Learning,‖ Springer, Singapore, 2018, Pp. 27–34.
3. L. Wang, D. Miao, and Z. Zhang, ‗‗Emotional analysis on text sentences based on topics,'' Comput. Sci., vol. 41, no. 3, pp. 32–35, Mar. 2014.
4. L. Kang, L. Xu, and J. Zhao, ‗‗Co-extracting opinion targets and opinion words from Online reviews based on the word alignment model,'' IEEE Trans. Knowl. Data Eng., vol. 27, no. 3, pp. 636–650, Mar. 2015.
5. D. Tang, F. Wei, and B. Qin, N. Yang, T. Liu, and M. Zhou, ‗‗Sentiment embeddings with applications to sentiment tanalysis,''IEEE Trans.Knowl. Data Eng., vol. 28, no. 2, pp. 496–509, Feb. 2016.
6. B. Ay Karakuş, M. Talo, İ. R. Hallaç, And G. Aydin, ―Evaluating Deep Learning Models For Sentiment Classification,‖ Concurr. Comput. Pract. Exp., Vol. 30, No. 21, P. E4783, Nov. 2018.
7. B. Pang, L. Lee, And S. Vaithyanathan, ―Thumbs Up?,‖ In Proceedings Of The Acl-02 Conference On Empirical Methods In Natural Language Processing - Emnlp '02, 2002, Vol. 10, Pp. 79–86.
8. N. Kalchbrenner, E. Grefenstette, And P. Blunsom, ―A Convolutional Neural Network For Modelling Sentences,‖ Apr. 2014.
9. Y. Goldberg And O. Levy, ―Word2vec Explained: Deriving Mikolov Et Al.'S Negative- Sampling Word-Embedding Method,‖ Feb. 2014.
10. M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, And E. Muharemagic, ―Deep Learning Applications And Challenges In Big Data Analytics,‖ J. Big Data, Vol. 2, No. 1, P. 1, Dec. 2015.
11. M. V. Mäntylä, D. Graziotin, And M. Kuutila, ―The Evolution Of Sentiment Analysis— A Review Of Research Topics, Venues, And Top Cited Papers,‖ Computer Science Review. 2018.
12. J. Einolander, ―Deeper Customer Insight From Nps-Questionnaires With Text Mining - Comparison Of Machine, Representation And Deep Learning Models In Finnish Language Sentiment Classification,‖ 2019.
13. Y. Ming Et Al., ―Understanding Hidden Memories Of Recurrent Neural Networks,‖.
14. B. Pang, L. Lee, and S. Vaithyanathan, ‗‗Thumbs up?: Sentiment classification using machine learning techniques,'' in Proc. Conf. Empirical Methods Natural Lang. Process. (ACL), Grenoble, France, Jul. 2002, pp. 79–86.
15. B. Pang, L. Lee, and S. Vaithyanathan, ‗‗Thumbs up?: Sentiment classification using machine learning techniques,'' in Proc. Conf. Empirical Methods Natural Lang. Process. (ACL), Grenoble, France, Jul. 2002, pp. 79–86.