2020 International Conference on Computer Communication and Informatics (*ICCCI* -2020), Jan. 22 – 24, 2020, Coimbatore, INDIA

Sentiment Analysis on Customer Reviews using Convolutional Neural Network

Mr. Srikanth Tammina
Hyderabad, Telangana, India
Srikanth.tammina95@gmail.com

Mr. Sunil Annareddy
Hyderabad, Telangana, India
Sunilkumarreddy15@gmail.com

***Abstract*— 21st century is touted as a data revolution where tons of data silos being created every second. These data silos are stored across various servers and creating authentic data mines across multiple locations. Business leaders across the world realized the significance of the information residing in these data silos and constantly trying to extract relevant information which can be used to strategize and make savvy decisions. Prodigious salient information out there concealed in customer reviews, tweets, social media comments, and the challenge is to extract hidden insights from it. It is an arduous process to extract meaningful information from these data silos manually. One of the imperative aspects in Natural Language Processing is sentiment Analysis - Extracting emotions, opinions, sentiments, fervor from the corpus of data and showcasing general opinion of the reviewers about the product. All previous works in NLP rely on traditional classification techniques like Support Vector Machines, Naïve Bayes, Maximum Entropy, Random forest, etc. With the advent of deep learning models, there is a reliable improvement in accuracy in Natural Language Processing. This paper introduces a deep learning technique namely Deep Convolutional Neural Network which captures a sentiment in the text corpus. The performance of this model will be evaluated on the Amazon product reviews and IMDB movie reviews.**

Keywords*— *Sentiment analysis, Convolutional neural network, Machine learning, word2vec, Feature extraction

I. Introduction

Since the last decade, social media platforms like Twitter, Facebook, Reddit, 9gag, Instagram and so on are getting proliferated with stupendous amounts of data. As many people logging into these social media services more text information is being generated on these platforms. To put it into a perspective, according to recent statistics [1] – As of June 2019, there are 4.4 billion internet users, 1,209,600 new data is generating every day from social media, 474,000 tweets are tweeted per minute and 293 billion emails are sent daily. At the same time, E-commerce is widely used by many people for buying and selling products. Information about the product feedback and follow up comments are generating at a vast scale. Companies are very much inquisitive about the general opinion about the specific product from these huge datasets of reviews from customers. One of the renowned E-commerce company Amazon sells around 100 million products and they had 310 million active customers. IMDB has 462.80 million visits to its website every month and gets around 2 million movie reviews every week. Analyzing these reviews to understand the sentiment around the product or movie submitted by millions of people manually is an impossible task. To effectively study the emotion in these user reviews, sentiment analysis has gained widespread keenness among researchers. Sentiment analysis has a wide range of applications like recommending products pertaining to the likes of a customer, predict the result in political election depending on the user reviews on a political party generated from twitter, Facebook or any other social media. Sentiment analysis is achieved on three hierarchical levels: Sentence level, attribute or aspect level and Document-level

A. *Sentence Level analysis* – As the name suggests, Sentiment analysis is performed at the sentence level. If a document comprises of 7 sentences, SA classifies each sentence into either positive, negative or neutral bins. Subsequently, analysis can be performed to classify the sentence as objective or subjective sentence depending on the polarity.

B. *Attribute or aspect Level analysis* – This level is used to analyze different features/aspects of a specific product. Sometimes, it is not sufficient to classify a product as positive or negative. Customers may want to know what aspects of the product are positive and negative. Example: Analyzing emotions for features on a smartphone like a camera, battery, build and processor. Each aspect has different sentiment depending on user review.

C. *Document Level analysis* – The task at this level is to determine the general sentiment of the document as a whole. The general assumption at the document level is it considers the whole document as a single entity. Many algorithms like Naïve Bayes, decision trees, Max entropy are employed at the document level to classify the sentiment.

Sentiment analysis is the subset within the scope of NLP and computational linguistics which comprises pre-processing, classification and apprehension of the mood and opinion of the customers expressed across the internet.

978-1-7281-4514-3/20/$31.00 ©2020 IEEE

Typically, Sentiment analysis is performed by two main techniques: Lexicon based approach and Machine learning approach

A. *Lexicon based approach* – This approach is unsupervised learning where labeled input data is not required for training and testing. This method scrutinizes for intrinsic aspects to find an association between elements in the dataset. Lexicon based learning leverages the use of sentiment lexicons which are dictionaries of words annotated with their sentiment score and strength. Each word in a document is scanned through a dictionary of positive and negative lexicons and associated polarity value is assigned to each of the words and combined function is applied to calculate the final prediction for the document. The main advantage of a lexicon-based approach is not relying on an accurately labeled training dataset.

B. *Machine learning approach* – This approach requires a training data set with preassigned labels. This is fed to a supervised learning algorithm to train the classifier with labeled training data set. Once the data is fed into a supervised algorithm, features are extracted from the input data to simplify the learning between input and output. Features are a subset of a training dataset that merits attention while averting the rest. Few methods to extract word embeddings are Global vectors, word2vec, Latent Semantic Analysis and so on. Once the features are extracted, they are fed into classifiers that train on the feature data set and classifies the output into bins. Some notable machine learning classifiers are Support Vector Machines, Naïve Bayes, Maximum entropy, decision trees.

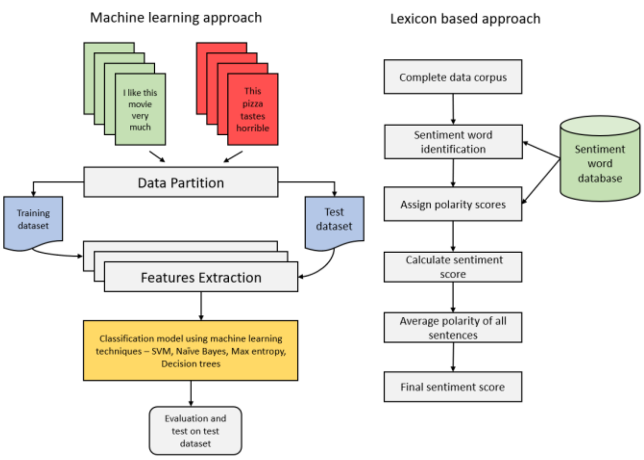

Fig. 1. Comparative Architecture of approach between Machine learning approach and Lexicon Based approach.

Researchers have been developing complex machine learning models to capture accurate emotion behind the text and in this process, they developed many successful algorithms like decision trees, Naive Bayes, Max entropy, SVM. As data compounding every day at an enormous scale there are few challenges that researchers are facing like complexity in the data and computational deficiency to tackle big data corpus. Since the past decade, with advance developments in deep learning, various deep learning models like auto-encoder-decoder, Unidirectional and Bidirectional Long Short-Term memory model, Recurrent neural network and convolutional neural network have been utilized to carry out the task of handling huge information of data. These kinds of models effectively try to identify features or relevant information in the input dataset intrinsically and by heuristic learning it learns representations and correlations of intricacies in the data to tackle various NLP tasks. In this paper, we recommend an approach to understand the emotion in the customer reviews in the real world by unleashing the power of the Convolutional Neural Network model (CNN) to perform sentiment analysis.

In recent developments, while tackling pain points in sentiment analysis, deep learning methods are introduced to handle high intricacies in the data. There are two imperative machine learning models for handling opinion mining, which is hierarchical- structured model and sequence base structured model. Convolutional neural networks come under the hierarchically structured model and Recurrent neural networks (RNN), gated recurrent units (GRUs) and Long Short-Term memory model (LSTM) come under sequence-based model. In the Convolutional neural network an input layer, hidden layer, and output layer are in hierarchical structure. The hidden layer comprises the convolutional layer as a feature extractor, Max pooling layer as a condenser and a fully connected layer which acts as a classifier. The convolutional layer in the network leverages the use of kernels which are filters of various matrix sizes to construct disparate feature maps. In image processing, these kernels of fixed sizes are applied to the input image pixels by convolution operation and feature maps are extracted from the image. By application of relevant kernels, spatial and temporal dependencies are successfully captured without any loss of information. In term of natural language processing, the use of kernels of different sizes is deemed as various N-gram matrices which analyze the combination of multiple sentences phases and interpret each N-gram differently. These feature maps from applying kernels on training dataset play a paramount role in classifying polarity across sentences. For example, let us posit a sentence "this chicken golden delight pizza from dominos would have taste best if the crust is properly baked", In this sentence if we use size two kernel to get feature maps which are bi-gram words then one of the bigram words "taste best" gives positive connotation whereas if we use convolution operation of size four to get relevant feature maps which are 4-gram words and while considering one 4-gram word like "would have taste best" this emanates sort of negative connotation. By the conclusion, employing right kernels in permutations and combinations elucidate a better understanding of features and enhancing model accuracy. This paper predominantly explicates one of the deep learning methods based on the Convolutional neural network to perform sentiment analysis. Section II describes related background work in the field of sentiment analysis. Section III elucidates the architecture of the Convolutional neural network. In Section IV we propose our model and methodology. Section V and VI describe the dataset and show the performance metrics of this model performed on IMDB movie reviews and Amazon customer reviews. Section VII shows the conclusion and future scope of our work.

II. Background Work And Literature Survey

In 2008, Bo Pang and Lillian Lee [1] from the Yahoo research team introduces opinion-oriented information-seeking systems in their paper *Opinion mining and sentiment analysis*. They used unsupervised lexicon induction to determine the degree of positivity determined by the lexicons and observed that unigrams performed better than bigrams and trigrams while classifying sentiment in movie reviews. On the contrary Dave et al. in 2003 observed in his experiment that bigrams and trigrams as better feature extractors than unigram for product reviews. In 2009, Alec Go, Richa Bhayani and Lei Huang [2] introduce a novel approach for automatically classifying the sentiment of Twitter messages. They considered a training set of twitter messages with emoticons. Instead of hand labeling the data set which merits many hours, they leverage the use of distant supervision where the extracted tweets populated with emoticons. By applying rules on the emoticons, they labeled training data set and used machine learning algorithms Naïve Bayes, Max entropy and SVM to train the input data. They achieved an accuracy of 80 %. In January 2010, Alexander Pak and Patrick Paroubek [3] showcased how to automatically collect a huge set of corpora for sentiment analysis and built sentiment classifiers on top of it to determine positive, negative and neutral sentiment at sentence and aspect level.

Ilya Sutskever and Kai Chen [4] in 2003 discussed in their paper *Distributed Representations of Words and Phrases and their Compositionality* about learning of word vector representations through deep learning models. In 2011, Jason Weston and Leon Bottou [5] proposed a new coalesced neural network architecture that can be employed to disparate NLP tasks. Their learning algorithm learns internal representations based on the enormous amount of unlabeled training data. In 2014, Yoon Kim [6] in his paper implemented a layer convolutional neural network on top of word vectors which are trained by Mikolov in 2013 on billions of words from unsupervised learning. In 2010, Luciano Barbosa and Junlan Fang explore some features of how tweets messages are written and exploit sources of noisy labels as their training data. In 2014, IBM researchers Cicero Nogueira dos Santos and Maira Gatti [7] proposed a new DCNN that leverages character-to-sentence level information to employ sentiment analysis of short texts.

III. Convolutional Neural Network

Over the past few decades, deep learning techniques are employed in a wide range of verticals in businesses like Manufacturing, Supply chain, Banking, Health care, etc. Few applications of deep learning in these domains are predicting credit default customers, demand forecasting of a product in upcoming days, spam email detection and so on. As data booming every second these advance deep learning models are gaining a lot of attention. One of the successful variants in a deep neural network is Convolutional Neural Network. The fundamental and renowned usage of CNNs are used in the image processing field where it is proven effective and successful in predicting images into their respective classes. The same mechanism of identifying features in image data set and predicting the image class is extrapolated into a new field of Natural Language Processing. Here the input is a set of sentences in a document that needs to be segregated into different sentiment classes namely positive, negative and neutral.

In 1962, Dowling and Werblin are working on the visual cortex of the human brain. In this process, they discovered receptive field structures in the visual cortex. The fundamental role of visual cortex is to process the visual information. There are multiple levels of receptive field structures inside the cortex such as Primary visual area, Visual association area, and Higher visual association area. The visual information perceived by the human eye is passed to these hierarchical layers of receptive field structures to comprehend the information and makes sense of what the eye sees. Similarly, these structures act as a layer in CNNs. In 1998, the first convolutional neural network was introduced in a paper by Bengio, Le Cun, Bottou and Haffner. This CNN was named LeNet-5 and its main function is to classify digits from handwritten numbers.

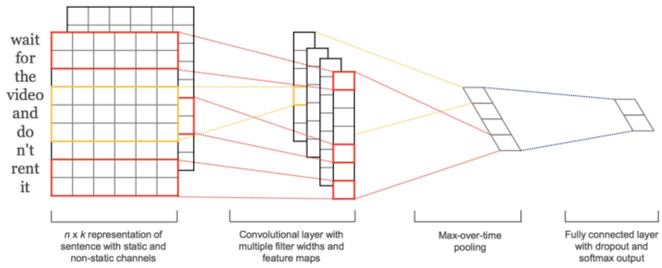

Fig. 2. Kim, Y . Convolutional neural network [6] for sentence classification

The above figure shows vanilla one layer convolutional neural network architecture. This network has two key components namely Feature extraction layer to extract relevant features from the input and the Classification layer to classify the input into various classes by learning non-linear combinations of the features from the previous component. Finally, The CNN model is trained using backpropagation or gradient descent to minimize the loss function and accurately predict the classes for each sentence in the corpus. The CNN comprises of:

A. *Convolutional Layer* - The first layer in CNN is convolutional layer. Convolutional layer consists of sets of kernels or filters or feature extractors of various sizes that transmute input matrix into output through a mathematical operation called Convolution. The matrix size of kernels is comparatively smaller in dimension than those of the input matrix i.e $I_w > F_w$ where I_w represents the dimension matrix of input and F_w represents the size of kernel filters. The imperative characteristic of convolution operation is weight sharing across all input matrix. Sharing of weight reduces the number of parameters for efficient learning of model and accurate generalization. These kernels capture the spatial information like edges, corners, etc. of the input matrix which act as relevant features. The kernel matrix comprised of shape (i, j) where i is the width and j are the height of the matrix. A Linear operation called convolution is calculated where kernel matrix of shape (i, j) runs on the input matrix of shape (m, n) to get one feature map. i and j are the parameters that can be customized by the user. This convolution operation is carried out by using multiple filter kernels on input matrix to form different feature maps. These feature maps contain disparate characteristics of the input data. The output size of convolutional layer is computed as follows:

$$O = \frac{W-K+2P}{S} + 1$$

Where O is the output height, W is the input height, K is the filter size, P is padding, and S is stride. Stride is the amount by which the kernel filter is moved by as the kernel hovers over the image. During convolution, kernel precludes the center of kernel to cover the outermost elements in the input matrix and this reduces the size of the outer feature map compared to the remaining feature maps. To counter this problem, zero padding is used where zeros are added to outermost layer of the input matrix.

B. *Nonlinear Activation Function* – The output of the convolutional layer is fed into nonlinear activation function to introduce non-linearity into the network. Typically, convolutional operation (element-wise matrix multiplication) is a linear transformation of input which is inadequate to create a universal function approximator as many real-world problems are non-linear in nature. Commonly used activation functions are Sigmoid or logistic, Hyperbolic tangent and Rectified linear unit (ReLU). By employing these activation functions after the convolutional layer, the network churns out a nonlinear decision boundary via a non-linear combination of weights and inputs.

C. *Pooling Layer* – This layer is placed after the convolutional layer and it operates on each feature maps discriminately [8]. The pooling layer is a down sampling operation that minimizes dimensions of the feature maps, preserves spatial dependencies and avoids overfitting. It also reduces the number of parameters and computational complexity in the network. The two most common pooling layers used are the average pooling and max pooling layer. Max pooling takes the maximum feature value among feature maps and average pooling calculates the average values of all feature maps and outputs a condensed version of one-dimensional feature map.

D. *Fully Connected Layer* – The output of the convolutional layer and subsequent pooling layer is condensed into a one-dimensional vector of numbers and each number in this array is connected to one or multiple fully connected layers also called dense layers. Features drawn out from the convolutional layer and down sampled by pooling layer are mapped to a fully connected layer to learn the nonlinear combinations of the high level features. Following fully connected layer, an activation function is applied which normalizes real values in the output from fully connected layer to target class probabilities which range between (0,1). Different set of tasks requires particular set of activation function.

Task	**Activation Function**
Regression to continuous values	Identity
Multi class output values	Sigmoid / Softmax
Binary output (two classes)	Sigmoid

IV. Proposed Model and Methodology

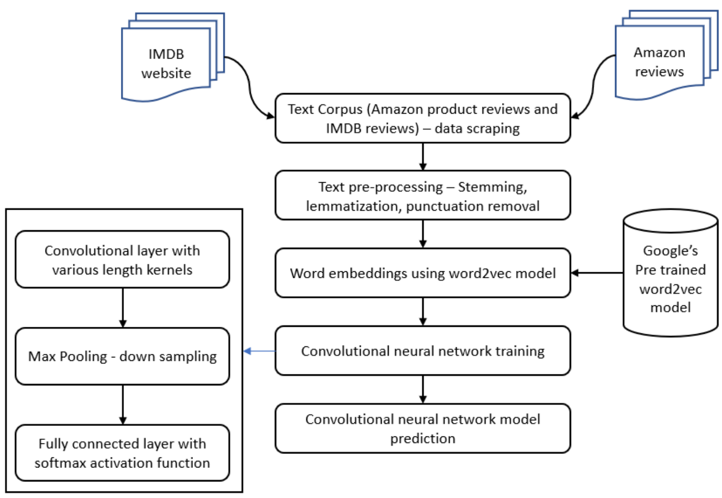

Fig. 3. Proposed approach using word2vec and CNN to classify opinions

A. Leveraging Google's Pretrained Vectors – The project word2vec was created by Google in 2013 on Apache. As deep learning models take numerical data into inputs to perform training whereas in sentiment analysis the input data is in text format. So, we use word embeddings which are texts converted into numbers. Instead of randomly initializing weight vectors onto each word and train them using machine learning models we leverage the use of word embeddings which are pre-trained vectors created by Google [9]. These vectors are encoded using two techniques – CBOW (continuous bag of words) and the Skip-gram model. Google trained about 100 billion words and published these pre-trained vectors online. The model contains 300-dimensional vectors for 3 million words and phrases. In this paper, we converted the sentences into word vectors of n-dimensional space and feed these pre-trained vectors as input into CNN.

B. Model – The word2vec pretrained vectors for each word in sentence act as input layer for CNN architecture. Let us assume Corpus (C) $\mathbf{C_i \in R^h}$ where R^h is the h-dimensional pre-trained word vector (each word in sentence creates one word vector with dimension h) for i-th word in input corpus C. If length of corpus is Z which comprises of many words and sentences, then

$$C_{1:z} = C_1 \oplus C_2 \oplus C_3 \oplus \ldots\ldots \oplus C_z$$

$\oplus$ is concatenation operator. Let us take a filter (F) with dimension $\mathbf{F \in R^{kh}}$ where R^h is the h-dimensional pre-trained vector and k is size of each kernel filter and also the number of words in input matrix which will be involved in convolution operation. The input word vectors in corpus C on which the convolutional operation of first kernel filter of size k will be applied are

$$C_{x:x+y-1} = C_x \oplus C_{x+1} \oplus C_{x+2} \oplus \ldots\ldots \oplus C_{x+y-1}$$

The first feature map that churns out from the convolution operation between word vectors $\mathbf{C_{x:x+y}}$ and kernel filter **F** is [10]:

$$M_i = f(\, F \bullet C_{x\,:\,x+y-1} + \text{bias}\,)$$

Now this filter F of size k glides across all the possible combinations of words in the corpus **C** $\{ C_{1:x}, C_{2:x+1}, C_{3:x+2}, C_{4:x+3} \ldots\ldots C_{n-x+1:n} \}$ to give different features maps **M** :

$$M = [\, M_1, M_2, M_3, \ldots\ldots, M_{n-x+1} \,]$$

The output of feature maps is fed into max-pooling layer where max-pooling operation takes the maximum value of M (M_{max}) which is the most important feature with highest value as the feature equivalent to its particular kernel $F \in R^{kh}$. This whole process is derived for one filter F and to extract one feature from max-pooling layer. Multiple kernels with disparate kernel sizes are used to attain numerous feature maps. These features are fed as input into fully connected layer with softmax activation function to get the probabilities of classes between (0,1) and perform classification.

The first step in our proposed model is Data cleansing. By using regex expressions, we removed punctuation marks. Then text preprocessing namely, tokenization, lemmatization and stopwords removal are performed on two datasets. Before sending these structured sentences into our CNN model for training, we need to convert string into word vectors. We used Google pre-trained vectors - word2vec to represent these sentences into word vectors. Word2vec comprehends and vectorizes the meaning of words in a text based on the hypothesis that words with similar meanings in a particular context display adjacent distances. This word2vec matrix act as an input layer for CNN. These vectorized values of each word is sent to first layer namely, Convolutional layer where features are extracted and were used to create a feature map. Then these feature maps are sent to max pooling layer and extracts the maximum value out of these feature maps. This process is repeated by changing the kernel size and number of kernels to extract different feature maps. These feature maps are sent to final layer namely, fully connected layer which classifies the given text into positive or negative polarity.

V. DATASET

We use two datasets to perform sentiment analysis. One of the corpuses is IMDB dataset that contains the movie reviews and the other is Amazon product reviews dataset. The rating distribution of IMDB dataset comprised of 2 classes (0 for negative review and 1 for positive review) and the distribution of Amazon reviews have 5 classes of rating from 1 to 5 (1 being negative , 2 being somewhat negative, 3 being neutral, 4 being somewhat positive and 5 being positive). Below table shows the detail statics of the datasets.

Data	No of text data (T)	Dist. (+, -)	Train:Val:Test
IMDB	50000	50:50	35000:5000:10000
Amazon	34,000	95:5	23800:3400:6800

Above table shows the statistics of data where two different datasets are IMDB and Amazon reviews, T is the number of text reviews overall, Distribution shows the percentage distribution of positive and negative texts. For our model we categorized the data into train, test and validation. The respective ratios are shown in the table.

VI. Results

The two datasets described above are applied onto CNN model and also, we test this dataset on traditional machine learning models namely Random forest and Naïve Bayes. The parameters that we initialized into CNN are dropout of 0.2 and we leverage the usage of Adam optimizer instead of stochastic gradient descent to update network weights during training. Dropout is a generalization technique used in deep learning models to prevent over-fitting.

Comparison of Accuracy for different Machine learning methods and CNN algorithm on test data set

Classifier Model	Accuracy %	
	IMDB	Amazon
Random Forest	55.8	44.3
Naive Bayes	54.3	46.3
Word2Vec + CNN	74	68

The initial corpus of IMDB and Amazon reviews are segregated into 70% training dataset, 10% into validation dataset and 20% into test dataset. Each sentence consists of words are converted into vectors by leveraging the usage of Google's pretrained word vectors. We converted the sentences into word vectors of n-dimensional space and feed these pre-trained vectors as input into CNN. Comparatively, Convolutional Neural Network algorithm achieved highest accuracy of 68% on Amazon dataset and 74% on IMDB dataset whereas, Random Forest scored 55.8% on IMDB and 44.3% on Amazon dataset. Naïve Bayes achieved an accuracy of 54.3% on IMDB and 46.3% on Amazon. These results implies that deep learning model namely, CNN indeed learned better compared to Naïve Bayes and RF. CNN efficiently leveraged the information extracted from the feature maps and classified better than the others. It took 2 hours to train on both datasets on an i7 intel processor with 8 gb of RAM.

VII. Conclusion And Future Work

In this paper, we have explored one of the deep learning techniques and how to employ this deep learning algorithm - CNN on to text dataset and extract general opinion from the customer reviews. We have also leveraged the use of google pre-trained vectors word2vec to extract vector data from text corpus in order to feed as an input into our network. The experimental results conclude that convolutional neural network when properly trained can outstrip and supplant traditional machine learning models. With little tuning of the hyperparameters like the size of the filter, the number of kernels, dropout probability can lead to better accurate generalization of the trained model. In future sentiment analysis would be key to all the e-commerce and many business verticals to understand the opinion of the customers towards their product or services. Ever-increasing of online user data also merits accurate sentiment analysis. There is an imperative and expedite the need for understanding these sentiments and take insightful decisions. A lot of hidden insights and valuable information can be extracted from sentiment analysis. The application of sentiment analysis is a wide range like a product recommendation, public sentiment analysis, predict the political results of a party based on people's opinion and so on. We believe our model and results can further galvanize other researchers to use more deep learning models like Recurrent neural network, Long Short-Term memory model [10] [11] [12] and understand the emotion in customer reviews even more accurately.

References

[1] Pang, Bo & Lee, Lillian. (2008). Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval. 2. 1-135. 10.1561/1500000011.

[2] Go, Alec & Bhayani, Richa & Huang, Lei. (2009). Twitter sentiment classification using distant supervision. Processing. 150.

[3] Pak, Alexander & Paroubek, Patrick. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Proceedings of LREC. 10.

[4] Tomas Mikolov and Ilya Sutskever and Kai Chen and Greg Corrado and Jeffrey Dean. (2013). Distributed Representations of Words and Phrases and their Compositionality. arXiv

[5] Ronan Collobert and Jason Weston and Leon Bottou and Michael Karlen and Koray Kavukcuoglu and Pavel Kuksa. (2011). Natural Language Processing (almost) from Scratch, arXiv: 1103.0398

[6] Kim, Y. Convolutional neural networks for sentence classification. arXiv 2014, arXiv:1408.5882.

[7] C. N. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in Proceedings of 25th International Conference on Computational Linguistics, 2014

[8] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," Journal of Machine Learning Research, vol. 12, pp. 2493–2537, 2011

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Proceedings of 27th Annual Conference on Neural Information Processing Systems, 2013, pp. 3111–3119.

[10] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in Proceedings of 2011 Conference on Empirical Methods in Natural Language Processing, 2011, pp. 151–161.

[11] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in Proceedings of 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2012, pp. 1201–1211

[12] F. A. Gers, J. Schmidhuber, and F. A. Cummins, "Learning to forget: Continual prediction with LSTM," Neural Computation, vol. 12, no. 10, pp. 2451–2471, 2000.

[13] S. Dhar, S. Pednekar, K. Borad and A. Save, "Sentiment Analysis Using Neural Networks: A New Approach," *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, 2018, pp. 1220-1224.

[14] W. Li, Z. Li and X. Fang, "Short text sentiment analysis based on convolutional neural network," *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Limassol, 2018, pp. 291-295.

[15] I. Santos, N. Nedjah and L. de Macedo Mourelle, "Sentiment analysis using convolutional neural network with fastText embeddings," *2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, Arequipa, 2017, pp. 1-5.