International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 12, No.1 (Dec-2022)

https://dx.doi.org/10.12785/ijcds/1201126

RNNCore: Lexicon Aided Recurrent Neural Network for Sentiment Analysis

Nikita Taneja[1], Hardeo Kumar Thakur[2]

[1] *Computer Science and Technology, Manav Rachna University, Faridabad, India*
[2] *Computer Science and Technology, Manav Rachna University, Faridabad, India…*

Received 05 Jul. 2021, Revised 19 Jul. 2022, Accepted 24 Aug. 2022, Published 31 December 2022

Abstract: Sentiment Analysis (SA) or Opinion Mining can help in identifying subjective information conveyed by user reviews for various automation tasks such as building better recommendation systems, identifying user trends, monitoring, and customer support. This paper focuses on sentiment score detection. Traditional SA algorithms suffer from low accuracies in identifying true user intents. However, with the advent of Deep Learning many NLP (Natural Language Processing) tasks including Sentiment Analysis have become feasible with accuracies comparable to that of human experts. An additional advantage of Deep Learning in contrast to supervised learning is that in deep learning a manually tuned feature set is not required. Deep Learning algorithm such as Convolution Neural Networks (CNN), Long Short Term Memory (LSTM), Recurrent Neural Networks (RNN)and various other have successfully been applied to SA. RNN, in particular, is well suited for this task, however, most of the works done over RNNs require large supervised training sets which are usually not available for all domains. This work proposes a new method called RNNCore which can make use of the pre-trained word embeddings from Stanford Core NLP in conjunction with RNN to improve accuracy and reduce computation cost. Comparison between the results of RNNCore, RNN, and OneRmethod on the IMDB review dataset suggests that RNNCore yields 92.60%F1-measure which is a marked improvement of 17.74% as compared with a simple RNN approach for Sentiment Analysis.

Keywords: Deep learning, Sentiment Analysis, Natural Language Processing, Recurrent Neural Networks.

1. Introduction

With the great progression of social media, user reviews are constantly generated from all over the internet every second which are easily accessible for further analysis. Increasingly companies have been trying to use this information to evaluate client satisfaction, preferences and provide users with recommendations [1][2]. As a consequence, there is a need to build computational models to analyze these data. These models are needed to detect the user opinions concerning products or services under consideration. The Online user can provide readily available textual information which can be used for various NLP tasks [3], sentiment analysis being one of them. This information is usually subjective expressions that describe the reviewer's feelings toward given products and services. The user provides a claim about the products and services associated and often associates a sentiment, which can be 'positive' or 'negative' or even neutral toward the topic; Sentiment always involves the user's desires and intents. The Goal of the Sentiment analysis task is to deal with the computational understanding of the provided reviews using textual analysis. It tries to determine the mindset of a user towards certain products and services. The user's mindset can reflect his state of mind and the intended emotional communication requirements. The SA (Sentiment analysis) can be used for identifying critical beliefs of the user about products or services by mining online user reviews. Also, it can be applied in tracking the shifting attitudes and interest [4][5]of the user toward products, services, public topics, etc through mining reviews which is also quite useful. This extracted information can then be used to notify other customers about their emotional attitude (positive or negative) towards the product. Tracking user trends of product reviews is also gathering research support, as the track record can be utilized for the changing consumer preferences. The detection of

E-mail: nikita.taneja@gmail.com, hkthakur@mru.edu.in

"flames" [6], Sentiment classification will also benefit too heated or aggressive language in e-mails or on social networking platforms. Monitoring newsgroups and forums, where quick and automatic flaming detection is required, will likewise see significant gains. Sentiment analysis can also aid the creation of new types of search engines and recommendation systems, as these systems should not propose something that has received unfavorable feedback. However, ambiguity [7] is one of the most serious issues in the field of Computational Linguistics. This difficulty can only be solved if computational systems have some type of world knowledge or at the very least a rudimentary dictionary or any artificial intelligence-based decision support becomes possible. The Semantic ambiguity present in the user reviews is very strongly related to vagueness, and can never have well-defined meanings in all senses. Another problem is inference [8] i.e. the process to find and identify implicit relations. SA becomes even more complicated when its task also is to identify neutral sentiments. The additional problem of SA is the availability of large datasets, large supervised training sets which are usually not available for all domains. This work tries to improve on the SA task by reducing the ambiguity using a pre-trained lexicon i.e. the Stanford CoreNLP [9] and inference by using an efficient deep learning architecture of RNN [10] on the IMDB review dataset [11]. Most existing deep learning algorithms need manually labeled data sources for the training of the algorithm, this data is required to ascertain that the algorithm can deliver good accuracy. Thus a huge text/review corpus is needed for training of the deep learning algorithm for human-like prediction of the input reviews. However, collection and then manually labeling such large-corpus requires a lot of resources in terms of time and cost as well. Also, all deep learning models require a very large no. of parameters or the weight adjustments of the underlying neural network layers [12], the time complexity of these models is quite high. Furthermore, deep learning algorithms require very high-end hardware, multiple GPUs, a CPU, and large RAM. Initializing the neural networks using already trained word vectors taken from a supervised language model can be a good method for improving the existing neural networks such as RNNs performance when a large corpus is unavailable. Hence, this work uses the pre-trained word embeddings from Stanford CoreNLP in conjunction with RNN. This method can capture both the semantic and syntactic information from the user reviews, this information is very important for the generalization of the sentiment analysis task. Simply put: RNN requires structured information about the language, which is difficult to get from the raw data. This work combined the RNN with the Word embedding model from CoreNLP to get better results faster.

The motivation of the work comes from the need for linguistic resources, such as lexicons, are required to accomplish the sentiment analysis goal. The word sentiment polarity can be determined using a variety of lexicons. SentiWordNet is the most impressive of these. These lexicons, however, have several drawbacks. SentiWordNet, for instance, these lexicons assigns sentiment to words rather than real-world concepts, making it very difficult to identify and score different meanings of the same word in the lexicon. In addition, there is uncertainty, making it difficult to determine which is more suited in each scenario. Even while basic sentiment annotation is sufficient for some applications, further learning based on words is required for efficient emotion analysis. When examining consumer perceptions of a product, also from a customer point of view he wants to know more about the customer service. In this situation, it is important to understand the emotional meaning of the sentences that make up the review collectively not just words. To this end, lexicon-based training is needed over the reviews. To overcome such limitations, we propose usage of pre-trained lexicon with sequence learning neural networks which are RNNs to further improve on the simple lexicon-based sentiment analysis.

Contributions of the work are as follows

1. Utilization of existing pre-trained word embedding to increase the efficiency of the sentiment analysis, for this end the work makes use of the Stanford Core NLP pipeline [42].
2. Development of an efficient RNN based sequence classifier, where the sequences of input words are used for training instead of a single word.

In addition to this general introduction and motivation, this present paper is organized into five sections. Section 1 provides the Introduction to Sentiment Analysis problem. Section 2 gives a detailed Literature Survey of Sentiment Analysis methods which includes the current knowledge of Sentiment Analysis. Section 3 focuses on the proposed methodology of the RNNCore method for sentiment analysis. The result of the proposed RNNCore algorithm is discussed in Section 4. Based on the results of experimentation and its analysis, concrete conclusions have been derived in these results, and analysis is shown in section 5.

2. RELATED WORK

The most basic method of sentiment classification is to employ a thesaurus [13] that has information about which words and phrases are good but which are negative. This thesaurus can be individually compiled or automatically obtained. Annotating corpus is normally done manually, and subsequently, methods are used to discover sentiments about new batches of words or phrases using enormous collections of data. Instead of relying on the polarity of words, other techniques can focus on mining phrases or entire reviews. In Lexicon Based approaches [14][15][16] the lexicon-based method employs a sentiment thesaurus comprised of opinion terms. By comparing these terms to the remainder of the data, the polarity is established. To comprehend how a sentiment value or score is allocated to input text using neutral, negative, and positive terms from the thesaurus. The lexicon-based systems employ a language model, which is a collection of recognized and pre-compiled collection of sentiment terms and concepts. The Lexicon-Based methods are divided into two subcategories: Dictionary-Based and Corpus-Based. The words that are routinely gathered and then manually annotated are used [17][18]. The number of synonyms for a given term in the dictionary is steadily increasing. WordNet [19] is just an illustration of a lexicon that may be used to create a lexicon corpus called SentiWordNet [20]. However, these approaches cannot handle subject-specific (domain) and context-based texts, which is their biggest flaw. The corpus-based method [21][22] provides a dictionary relevant to a certain domain. These dictionaries generate a collection of initial judgment phrases based on the search for appropriate words utilizing quantitative or linguistic methodologies. Latent Semantic Analysis (LSA) [23] and similar methods are based on semantics using statistics [24].The more recent classification based techniques uses training and testing of text. Generate a training set by hand and categorize it as neutral positive, or negative. After that, a test is run upon the test data to ensure that the method is reliable. The fresh reviews may then be classified using this approach. The most extensively employed machine learning algorithms for sentiment categorization are Naive Bayes (NB) [25], Maximum Entropy (ME) [26], and Support Vector Machines (SVM) [27][28]. Although providing a collection of labelled reviews to train the classifier is implausible, semi-supervised and unsupervised algorithms are meant to work around this. Also different kinds of ensemble classifiers [29][30] are also being developed for sentiment analysis. For performing features selection voting rule is used, it is to create an ensemble classifier depending upon the output of larger parts of classifiers, their classification is done. Ensemble classifiers, on the other hand, are not necessarily superior. New observations can cause the ensemble classifiers to get confused. Ensembles, in other words, cannot compensate for unknown disparities between the samples and the newer inputs.

With the advent of deep learning many NLP tasks including sentiment analysis have become feasible with accuracies comparable to that of human experts. Likewise, the advantage of deep learning in contrast to the supervised learning approach need manually tuned features. Deep learning methods such as Convolution Neural Networks (CNN) [31][32], Recurrent Neural Networks (RNN) [33][34][35], Long Short Term Memory (LSTM) [36] and various other have successfully been applied to SA.

Figure 1: Overall workflow of the proposed sentiment analysis method.

3. METHODOLOGY

The RNNCore method consists of various parts from data processing to the RNN based classification, the Overall workflow of the proposed sentiment analysis method is described in Figure 1 and explained later in this section. Also, the proposed RNN architecture as in figure 2 is trained over words embedding, which are numeric representations of the text. The RNN does not require manual selected features, it can identify its own set of features from the word embedding. Also RNN can utilize pre trained word embedding, which are provided by CoreNLP [37]. Thus the proposed RNN can take input corpora and word embedding (pre-trained) as inputs to generate a sentiment score as shown in Figure 2 below. RNN is trained to generate a sentiment score in the range of (1-5) instead of polarity (-1, 1). This is done to generalize the RNN output so that it can be used in other applications such as recommendation systems as input.

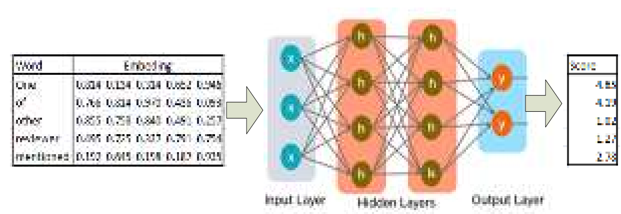

Figure 2: Using word embeddings for Sentiment analysis in RNNCore

Recurrent networks take into account not only the present input but also what they have seen in the past. There is an input layer, a hidden layer containing activations, and an output layer in the generalized architecture of a neural network illustrated in Figure 2. It is possible to expand the number of layers in an RNN by having one input layer take input from another, i.e., after receiving information from the input layer, the first hidden layer activates the next hidden layer, and so on. Finally, the information arrives at the output layer, which generates the result (sentiment score). The weights of each hidden layer in RNN are trained separately in each epoch of training when varying lengths of reviews arrive at the input layer to generate sentiment scores at the output layer. Before sentiment analysis is done using the RNN, pre-processing is one of the vital methods to get efficient results. Following steps are followed in pre-processing stages

3.1 Data preprocessing

There is a certain amount of irrelevant data available within the data which is collected from the user reviews. All arbitrary characters (such as special characters) or unreadable information is filtered out from the reviews. An NLP pipeline is applied for the removal of this useless data. All kind of grammatical associations which can exist in between the reviews is outputted by the NLP pipeline. The reviews that contain meaningful information are recognized using this pipeline. The outcomes aren't aided through facilitating filtering. Nouns, adjectives, and verbs are also discovered using this NLP pipeline. The NLP pipeline after preprocessing has further tasks as described as following:

3.1.1 Vectorization

The data has been changed to vector form by using the TF-IDF [38] function and find out the unique words from all the reviews which have been provided as input. It provides a single text file containing all the data needed for the tweet classification. Vector space model is the most widely used method in tweet representation. Vectorization model uses feature entries with associated weights for expressing the review information. To represent document object vector space model have been introduced. Vector $d = w_1, w_2, w_3, \ldots w_m$ means that there view words and its weight in review*ed*, the number of all word vectors. w_i $(i = 1 \ldots m)$ is the weight of the entry i in review d. The review vector set is the pattern or data object of the review clustering.

3.1.2 Part-of speech Tagging (POS)

Part-of-speech (PoS) tagging lets the system identify that what Part-of-speech each word in the text belongs to; it may be out of following noun, pronoun, adjective, verb, and interjection and so on. The purpose of PoS tagger is to uncover patterns in reviews using relative frequency assessment of the current segment of review.

3.1.3 Stemming and lemmatization

The process of stemming usually involves words with their roots also known as word stems. As the stem-related words for example "speak," "speaker," and "speaking" are rooted in one single word, "read". The bag of word dimensionality however is decreased. While using stemming, nevertheless, care must be exercised because it may exacerbate prejudice. Whenever the words "experiment" and "experience" are combined into one word, the skewed impact of stemming is visible.

3.1.4 Stops-Word removal

Prepositions, articles, and other words that serve as connectors in a sentence are examples of stop words. While there is no definitive list of stop words, several search engines employ several of the most popular, short word forms, like "the," "is," "at," "which," and "on." These words can be eliminated from the review before categorization since they appear often in the review and have not much effect on the sentence's final emotion.

3.1.5 Tokenization into N-grams

Tokenization is the technique of extracting a collection of words from the reviews. Words as well as other items are separated from the input string. Whitespace is a typical divider for recognizing separate words. Because user review data comprises multiple emoticons, URL links, and acronyms that can be easily distinguished as full entities, tokenization of social media data is far more challenging than tokenization of normal text. Combining adjacent words into phrases or n-grams, which can be unigrams, bigrams, trigrams, and so on, is standard procedure.

3.2 RNN Based Classification

The unique aspect of Sentiment analysis and related data is its temporal aspect associated with it. Each word in a sentence depends greatly on what came before and what comes after it. In order to account for this dependency, we need to use a sequence processing neural network. Recurrent Neural Networks in particular are well suited for this task. RNN are the feed-forward neural networks rolled out over time as such they deal with sequence data where the input has some defined ordering which gives rise to several kinds of architectures [39]. One of which is, vector to sequence models- these neural nets take in a fixed size vector as input and outputs a sequence of any length in image captioning like the image's vector representation can be input and its output sequence is a sentence that explains the image. Now Sequence to vector model- is the second type of neural network in which neural nets take a sequence as input and spits out a fixed-length vector. As in sentiment analysis, the film evaluation is considered as an input and the output is a fixed size vector representing how good or bad this person thought the movie was. The next and third one, sequence to sequence models- it is the more famous model among all and these neural networks take input as a sequence and outputs another sequence. Thus Applied to Sentiment Analysis the input could be a sentence user query and the output can be the agent response.

4. RESULTS AND ANALYSIS

The implementation of the RNNCore is done using MATLAB 2019a[42], the CoreNLP [43] is used for pre-trained lexicon provider, for the Analysis of the proposed RNNCore method IMDB review dataset [11] is utilized containing 50,000 reviews with equal polarities. The IMDB review dataset is in the area of movie reviews which is ideal for testing as there are enormous online sources of such user reviews, typically with metadata that offers easily extractable class labels, i.e. polarity (positive or negative). It is crucial to note that our approaches are not domain-specific and therefore can be easily transferable to other domains if adequate training data is available. Furthermore, movie reviews were shown to be more challenging to categorize than some other product reviews, with movie reviews being simpler to categorize than book and user reviews. For the analysis of the RNNCore, the dataset was divided into three sizes, 10,000 reviews (10K), 20,000 reviews (20K), and 50,000 reviews (50K) randomly. These sub-datasets were further evaluated against OneR Classifier, Simple RNN without word embedding, and the proposed RNNCore, RNN with pre-trained word embedding. The OneRClassifier [40] simply assigns the most frequent (i.e. the class with a maximum no of instances) class to the instance without any learning. The OneR acts as a minimum baseline for the classifiers. The simple RNN based classifier uses a weighted sum of output vector for estimating the sentiment.

4.1 Validation Accuracy

During the training, validation is done simultaneously to estimate the validation accuracy using loss per epoch (iteration). Mean Squared Error loss function is used to optimize a machine learning algorithm. This Validation Accuracy can be used to compare the models on various subsets of the datasets.

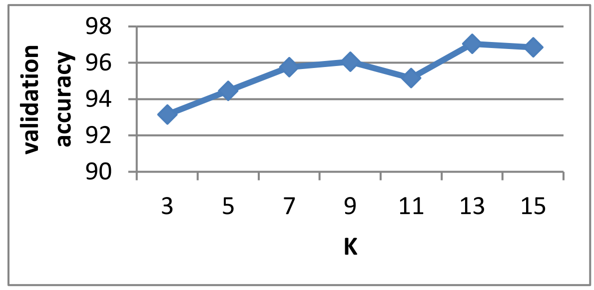

Figure 3: K-fold cross validation results for the proposed RNNCore for the values of K.

Cross-validation is used to validate the RNNCore model's generalization capabilities and also to avoid over-fitting. The RNNCore method is tested using K-fold cross-validation, with K values ranging from 3 to 15. Figure 3 shows the comparable findings. From Figure 3 it can be seen that the highest performance of RNNCore is attained when K is approaching 10.

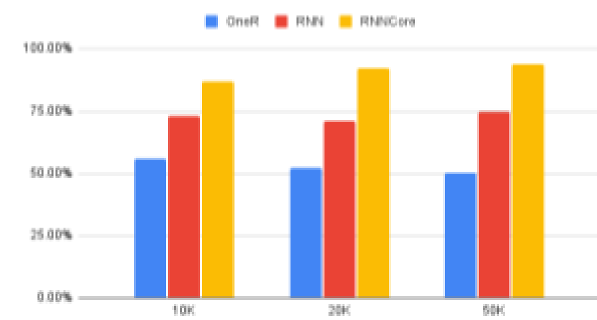

Figure 4: Validation accuracy of the OneR, RNN and RNNCore classifiers

As we can see from the results in figure 4, that the RNNCore outperforms both baselines in all three subsets of the datasets (10K, 20K, and 50K). RNNCore works well for all the subsets providing 86.80%, 91.80%, and 93.70% validation accuracy respectively which is higher than both OneR (56.10%, 52.07%, and 50.00%) and Simple RNN (73.10%, 71.10%, and 74.90%). On Average RNNCore provides 90.77% accuracy compared to 52.67% by OneR and 73.03% by RNN. RNNCore is 38.1% better than OneR and 17.74% better than RNN implementations.

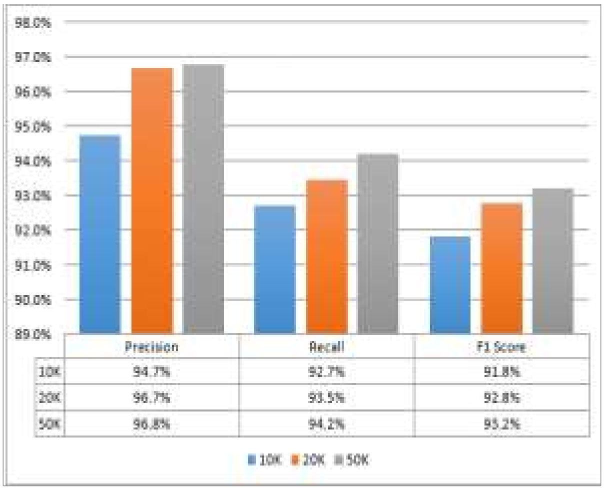

Figure 5: Precision, Recall and F1-Score of the RNNCore classifier with 10K, 20K and 50K instances

Experiments were further done to estimate the Precision, Recall and F1-Score [41] of the RNNcore algorithm as in Figure 5. In all the metrics (Precision, Recall, and F1-Score) RNNCore performed certainly well providing on average 96.07% Precision, 94.10% Recall, and 92.60% F1-Score on all three datasets.

4.2 Computation Cost Analysis

In terms of Computation Cost, RNNCore outperforms the simple RNN with twice as much efficiency. Table 1 shows the comparison between the RNN and RNNCore over the epochs used for training over the IMDB dataset. From 10 epochs to 100 epochs RNNCore works very well against the RNN implementation on average the RNN used around 19.5 minutes to train whereas the RNNCore took around 10.46 minutes, making RNN 46.5% slower than the RNNCore implementation.

The chart of data as in Table 1 is used to show the difference in performance and the shape of the curves. The shape of the curves for both RNN and RNNCore are not exponential as seen in Figure 6 and are reaching towards a peak, which is important because it shows that the algorithms take lesser amount of time as they achieve good accuracy. RNNCore however does this faster than the RNN implementation.

Table 1: Comparison of RNN and RNNCore sentiment analysis algorithms in terms of Computation Cost (minutes)

EPOCHS	RNN	RNN CORE
10	10.02447	3.6168
20	12.27535	6.1665
30	15.39875	7.8672
40	18.77193	10.3005
50	20.64936	11.346
60	22.02725	12.4299
70	23.09473	12.8586
80	23.78683	12.3231
90	24.10644	13.2915
100	25.3592	14.4135
Mean	**19.54943**	**10.46136**

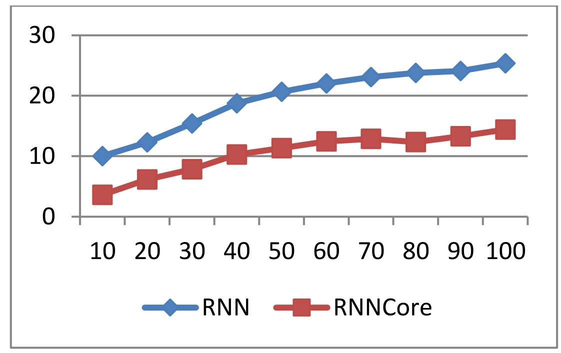

Figure 6: Computation Cost of RNN and RNNCore sentiment analysis algorithms

5. CONCLUSION

In this paper, an improved RNN based sentiment analysis method (RNNCore) was developed which utilized word embedding (pre-trained) to increase the efficiency of the traditional RNN network. This paper is focused on a novel method that can make use of such word embedding from Stanford Core NLP in conjunction with RNN to improve accuracy and reduce computation cost. The RNNCore is applied to achieve better sentiment analysis and finds sentiment cues more accurately than the conventional RNN up to 17% more. For upcoming development, we would like to incorporate RNNCore for aiding the recommendation models, especially cross-domain recommendation systems (CDRS). The movie review information can be used to fill the sparse cross-domain recommendation matrix for increasing the efficiency of the traditional CDRS.

6. REFERENCES

[1]. Yadav, Ashima, and Dinesh Kumar Vishwakarma. "Sentiment analysis using deep learning architectures: a review." Artificial Intelligence Review 53.6 (2020): 4335-4385.

[2]. Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. Electronics, 9(3), 483.

[3]. Maulud, D. H., Zeebaree, S. R., Jacksi, K., Sadeeq, M. A. M., & Sharif, K. H. (2021). State of art for semantic analysis of natural language processing. Qubahan Academic Journal, 1(2), 21-28.

[4]. Seo, S., Kim, C., Kim, H., Mo, K., & Kang, P. (2020). Comparative study of deep learning-based sentiment classification. IEEE Access, 8, 6861-6875.

[5]. Habimana, O., Li, Y., Li, R., Gu, X., & Yu, G. (2020). Sentiment analysis using deep learning approaches: an overview. Science China Information Sciences, 63(1), 1-36.

[6]. Agüero-Torales, M. M., Salas, J. I. A., &López-Herrera, A. G. (2021). Deep learning and multilingual sentiment analysis on social media data: An overview. Applied Soft Computing, 107373.

[7]. Jouravlev, O., & Jared, D. (2020). Native language processing is influenced by L2-to-L1 translation ambiguity. Language, Cognition and Neuroscience, 35(3), 310-329.

[8]. Heinz, J., De la Higuera, C., & Van Zaanen, M. (2015). Grammatical inference for computational linguistics. Synthesis Lectures on Human Language Technologies, 8(4), 1-139.

[9]. Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., &McClosky, D. (2014, June). The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations (pp. 55-60).

[10]. Ahmed, A., &Yousuf, M. A. (2021). Sentiment Analysis on Bangla Text Using Long Short-Term Memory (LSTM) Recurrent Neural Network. In Proceedings of International Conference on Trends in Computational and Cognitive Engineering (pp. 181-192). Springer, Singapore.

[11]. Liu, J., Zheng, S., Xu, G., & Lin, M. (2021). Cross-domain sentiment aware word embeddings for review sentiment analysis. International Journal of Machine Learning and Cybernetics, 12(2), 343-354.

[12]. Pathak, A. R., Pandey, M., &Rautaray, S. (2021). Topic-level sentiment analysis of social media data using deep learning. Applied Soft Computing, 108, 107440.

[13]. Esuli, A., &Sebastiani, F. (2007). SentiWordNet: a high-coverage lexical resource for opinion mining. Evaluation, 17(1), 26.

[14]. Hamouda, A., &Rohaim, M. (2011, January). Reviews classification using sentiword net lexicon. In World congress on computer science and information technology (Vol. 23, pp. 104-105). sn.

[15]. Kumar, K. N., & Uma, V. (2021). Intelligent sentinet-based lexicon for context-aware sentiment analysis: Optimized neural network for sentiment classification on social media. The Journal of Supercomputing, 1-25.

[16]. Eng, T., Nawab, M. R. I., &Shahiduzzaman, K. M. (2021). Improving Accuracy of The Sentence-Level Lexicon-Based Sentiment Analysis Using Machine Learning. International Journal of Scientific Research in Computer Science Engineering and Information Technology, 3307, 57-68.

[17]. Khoo, C. S., &Johnkhan, S. B. (2018). Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons. Journal of Information Science, 44(4), 491-511.

[18]. Fellbaum, C. (2010). WordNet. In Theory and applications of ontology: computer applications (pp. 231-243). Springer, Dordrecht.

[19]. Baccianella, S., Esuli, A., &Sebastiani, F. (2010, May). Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In Lrec (Vol. 10, No. 2010, pp. 2200-2204).

[20]. Rice, D. R., & Zorn, C. (2021). Corpus-based dictionaries for sentiment analysis of specialized vocabularies. Political Science Research and Methods, 9(1), 20-35.

[21]. Kumar, A., & Sebastian, T. M. (2012). Sentiment analysis on twitter. International Journal of Computer Science Issues (IJCSI), 9(4), 372.

[22]. Mihalcea, R., Corley, C., &Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In Aaai (Vol. 6, No. 2006, pp. 775-780).

[23]. Moreno-Ortiz, A., &Fernández-Cruz, J. (2015). Identifying polarity in financial texts for sentiment analysis: a corpus-based approach. Procedia-Social and Behavioral Sciences, 198, 330-338.

[24]. Govindarajan, M. (2013). Sentiment analysis of movie reviews using hybrid method of naive bayes and genetic algorithm. International Journal of Advanced Computer Research, 3(4), 139.

[25]. Appel, O., Chiclana, F., Carter, J., & Fujita, H. (2016). A hybrid approach to the sentiment analysis problem at the sentence level. Knowledge-Based Systems, 108, 110-124.

[26]. Ren, R., Wu, D. D., & Liu, T. (2018). Forecasting stock market movement direction using sentiment analysis and support vector machine. IEEE Systems Journal, 13(1), 760-770.

[27]. Xia, H., Yang, Y., Pan, X., Zhang, Z., &An, W. (2020). Sentiment analysis for online reviews using conditional random fields and support vector machines. Electronic Commerce Research, 20(2), 343-360.

[28]. Dashtipour, K., Ieracitano, C., Morabito, F. C., Raza, A., & Hussain, A. (2021). An ensemble based classification approach for persian sentiment analysis. In Progresses in Artificial Intelligence and Neural Systems (pp. 207-215). Springer, Singapore.

[29]. Rani, S., & Gill, N. S. (2020). Hybrid Model using Stack-Based Ensemble Classifier and Dictionary Classifier to Improve Classification Accuracy of Twitter Sentiment Analysis. International Journal, 8(7).Convolution Neural Networks

[30]. Giménez, M., Palanca, J., &Botti, V. (2020). Semantic-based padding in convolutional neural networks for improving the performance in natural language processing. A case of study in sentiment analysis. Neurocomputing, 378, 315-323.

[31]. Huang, M., Xie, H., Rao, Y., Liu, Y., Poon, L. K., & Wang, F. L. (2020). Lexicon-Based Sentiment Convolutional Neural Networks for Online Review Analysis. IEEE Transactions on Affective Computing.

[32]. Kurniasari, L., &Setyanto, A. (2020, February). Sentiment Analysis using Recurrent Neural Network. In Journal of Physics: Conference Series (Vol. 1471, No. 1, p. 012018). IOP Publishing.

[33]. Sachin, S., Tripathi, A., Mahajan, N., Aggarwal, S., &Nagrath, P. (2020). Sentiment analysis using gated recurrent neural networks. SN Computer Science, 1(2), 1-13.

[34]. Goud, A., & Garg, B. (2021). Sentiment Analysis Using Long Short-Term Memory Model in Deep Learning. In 2nd EAI International Conference on Big Data Innovation for

Sustainable Cognitive Computing (pp. 15-23). Springer, Cham.

[35]. Song, M., & Chambers, T. (2014). Text mining with the Stanford CoreNLP. In Measuring scholarly impact (pp. 215-234). Springer, Cham.

[36]. Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. Information Processing & Management, 39(1), 45-65.

[37]. Hussain, S., Sianaki, O. A., &Ababneh, N. (2019, March). A survey on conversational agents/chatbots classification and design techniques. In Workshops of the International Conference on Advanced Information Networking and Applications (pp. 946-956). Springer, Cham.

[38]. Radhakrishnan,P, Introduction to Recurrent Neural Network (2017,20August)TowardsDataScience,https://towardsdatascience.com/introductionto-recurrent-neural-network-27202c3945f3

[39]. Understanding LSTM networks (2017, 27August) Colah's blog, https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[40]. Jason Brownlee, (2020), One-Class Classification Algorithms for Imbalanced Datasets, https://machinelearningmastery.com/one-class-classification-algorithms/

[41]. Goutte, C., &Gaussier, E. (2005, March). A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In European conference on information retrieval (pp. 345-359). Springer, Berlin, Heidelberg.

[42]. Knight, A. (2019). Basics of MatLab® and beyond. Chapman and Hall/CRC.

[43]. Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014, June). The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations (pp. 55-60).

Nikita Taneja is working as an Assistant Professor at Department of Computer Science and Technology of Manav Rachna University (MRU), Faridabad. She has more than 13 years of teaching and industry experience in reputed institutions and organizations of India. She is pursuing her Ph.D. (Computer Engineering) from Manav Rachna University in the field of Cross Domain Recommender systems.

Dr. Hardeo Kumar Thakur is working as an Associate Professor at department of Computer Science and Technology of Manav Rachna University (MRU), Faridabad. He has more than 12 years of teaching and research experience in leading institutions of India. He has earned his Ph.D. (Computer Engineering) from University of Delhi in 2017 in the field of data mining. Dr. Thakur has published 14 research papers in international journal of repute, 13 papers in international conferences and 1 book chapter. His current research interests are Data Mining, Dynamic Graph Mining, and Machine Learning and Big Data analytics. He is an active referee for many international Journals and Conferences.