International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Hybrid Deep Learning Model-Based Approach for Sentiment Classification

Usha G. R.[*1], **Dr. J. V. Gorabal**[2]

Submitted: 17/11/2022 **Revised:** 16/01/2023 **Accepted:** 14/02/2023

Abstract: The sentiment analysis task is more complex considering the lack of relevant information in brief texts. Deep neural networks, like as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have been widely employed to extract information from data sentiment in recent years, with surprisingly good results. Though CNN can efficiently retrieve comparatively high features employing convolution and max-pooling layers, it cannot understand relationships' sequences. Parallelly bidirectional RNN models can extract contextual information and fail to extract local features. In this paper, integrated CNN and RNN models for sentiment analysis are examined to have the advantages of CNN's coarse grain local feature extraction and long-distance dependencies of RNNs. Particularly bidirectional LSTM and GRU networks associated with the convolution and max-pooling layer are used for sentiment analysis in SST-2 and movie review datasets. Two pre-trained word embedding techniques glove and word2vec are used. Experimental findings show that max performance is achieved at 93.44% for SST-2 and 95.42% for the movie review dataset using CNN BiGRU word2vec and CNN BiGRU glove, respectively.

Keywords: *CNN, LSTM, GRU, BiGRU, Glove, Word2vec*

1. Introduction

In the accelerated digital world, the commercialization of electronic commerce and social media platforms produces numerous online texts generated by the consumers, for example, customer reviews on products, services, and blogs. There are many advantages of online buying compared to physical such as one can buy anything from anywhere and anytime. Nevertheless, there are various issues for products offered on online platforms, for example, inconsistencies among description and actual products, inadequate quality of products, etc. [1]. Hence it is very important to understand and analyze the opinions given by the product purchased individuals. The assessment of these opinions or reviews helps customers in terms of selecting products and vendors to provide quality products and services to meet customer satisfaction. These online texts are comprised of independent and semantic information.

Therefore, it is very significant to assess the opinions/reviews provided by consumers and is commonly termed as sentiment analysis (SA) or opinion mining.

Sentiment analysis automatically assesses the individual's opinions expressed in terms of texts and recognizes the feelings towards the products/service [2]. It is of major direction for future research to properly discriminate and categorize the semantic relationship of such texts. Because constructing a model for sentiment analysis is difficult due to the following concerns. The feature extraction is the key parameter that decides the performance of the entire model. The conventional design methods of feature extractions are typically based on knowledge or statistical approaches [3]. But these conventional methods invariably neglect perspective and semantics with which opinions are provided and deteriorate the model's overall accuracy.

SA can be evaluated at different levels such as aspect, an ambiguous text, spam, subjectivity categorization, and so on [4-7]. By experiencing the past research work in SA, it can be broadly classified as lexicon-based [8], machine learning-based (ML) [9], and hybrid approach [10]. The hybrid approach is not explored due to its inevitable high complexity. ML-based SA performance is better than the lexicon even though it has fast training merit. ML-based algorithms are available like Naïve Bayes, Support Vector Machine, Maximum Entropy, neural network (NN), etc. The difference between traditional and NN architecture is as shown in Fig.1.

[1] *Assistant Professor, Department of Information Science and Engineering, Sri Dharmasthala Manjunatheshwara Institute oj Technology, Ujire-574240, Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India,*
ORCID ID: 0000-0002-3557-1060

[2] *Professor, Department of Computer Science and Engineering*
ATME College of Engineering, Mysuru, Karnataka, India
Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India
Corresponding Author Email: ushagr2@gmail.com

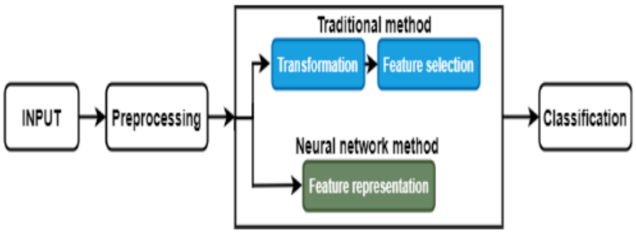


Fig.1. Traditional and NN architecture for SA

In recent years NNs gained more popularity due to their inherent properties such as being nonlinear, which helps to represent complex relations flexibly, being robust to noisy input environments, less execution time, easily predicting the probabilities, and so on [11-13]. Popular deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have achieved tremendous improvements in computer vision, audio, and text recognition since the inception of deep learning. Authors in [14] suggested the CNN model for SA using CNN and the training model by temporal and spatial information. The proposed method utilizes the weight shifting technique to reduce the computational complexities and train the design metrics. Nevertheless, CNN is well suited for applications that depend more on spatial information. As a result, the CNN model's effectiveness is heavily reliant on the selection of proper window sizes [15]. On the other hand, RNN is well suited for applications that depend more on temporal information (sequential data). RNN has poor performance in feature- extraction. RNN stores sequential information over time and suffers from scaling and fading problems.

The text categorization uses vectors to demonstrate the text, typically spatial information. RNN retrieves contextual information from features but cannot emphasize the most important information. Conversely, the convolution layer in CNN helps extract vectors' features and decreases their dimension. This research work aims to project a unique solution for SA employing a deep learning model that combines both CNN and RNN models to have the advantages of both deep learning models. The proposed structure uses CNN to extract features from the input at several locations of sentences and decreases its dimension. The extracted features are fed to the RNN models. Specifically, LSTM, BiLSTM, and GRU to retrieve the relative information to carry out SA. This work intends to study the effectiveness of different word embedding techniques combined with the CNN and RNN models. The paper is organized as follows- section 2 demonstrates earlier work carried out in SA using deep learning. CNN, different models of RNN, and word embedding techniques are described in section 3. In section 4, the experimental setup and results are presented. At last concluding remarks of this work are shown.

2. Related work

This section delves into the study on deep learning and the SA. In machine learning, deep learning is an emerging research area. It intends to imitate the knowledge and reasoning of the human mind. In 2006, the initiation of deep learning was proposed to create a fast-training algorithm for deep neural networks [16]. Deep learning has progressively been applied to research linked to natural language processing (NLP), like SA, thanks to the rapid development of deep learning. Word2vec [17] was presented in 2013 to get decentralized word vectors. The strategy has been proven in various NLP applications, and when paired with deep learning, it can generate even better results. Following that, GloVe, ELMo, and BERT [18-20] were proposed one after the other, demonstrating that deep learning for sentiment analysis has progressed significantly.

For SA, many generic machine learning procedures can be utilized. A basic and efficient baseline strategy is to characterize a sentence structure as a bag-of-words and then learn a linear classifier, for example, a logistic regression. This approach directly neglects syntax and semantic information from the document [21]. A further common way to express a sentence is via N-gram frameworks [22]. This model is best suitable for shorter sentences, and it takes into account the order of words, but it lacks in gathering spatial information. One can introduce low-rank matrices to represent linear classifiers [23] or employ multilayer NNs [24-25] to overcome these issues. Deep learning neural networks (DNNs) have grown more popular in NLP applications, where most of the effort entails training word vectors using neural language concepts [26] and then using the acquired word vectors to create a classification component. For text classification, DNNs combine feature extraction and classification. DNN-based techniques often begin with an input text that is represented as a sequence of words and then reflect every word in the sequence onto an incessant vector space. This is accomplished by scaling this with a weight matrix, resulting in a series of complex, real, weighted vectors. This sequence is then fed into a DNN that analyses it in many layers and produces a predictive probability. This process has been fine-tuned to maximize the accuracy of the model. RNNs reduce temporal complexity by analyzing texts word by word and preserving the semantics of all prior texts in a resolved hidden layer [27]. In RNN, the importance of recent words is greater than that of earlier words. The key elements could appear throughout the document, not just at the end. When it is used to capture the semantics of an entire document, this may reduce efficiency. To challenge this issue, the LSTM model was developed [28].

Originally, the CNNs were developed for computer vision [29], and they operate layers in conjunction with convolution filters employed to local characteristics. Many researchers have been inspired to use neural networks in different applications after examining the CNN-based computer vision outcomes. In recent decades, this field has been investigated, primarily through the use of multi-convolution and pooling layers in CNNs and the separation of classified representations of the input in order. CNN NLP models have generated exceptional results in sentence modeling, search query recovery, semantic parsing, and other NLP applications [30-32].

3. Proposed Approach

By integrating the CNN and RNN models, this work seeks to propose a novel strategy for improving sentiment classification on various data sets. The general structure of the proposed model is as depicted in Fig. 2.

This segment describes the proposed model in detail. The entire model is divided into the following stages

Word vector-matrix

The model takes the raw text input and separates it into words or tokens one by one in this step. Every token is transformed into a numeric vector. A word2vec and Glove, pre-trained word embedding models are used for converting the text into a numeric vector. Consider every text has n-words, then it is described as in (1)

$$Z = \{W_1, W_{2,}W_3, \ldots\ldots, W_n\} \in X^{n*d} \qquad (1)$$

where d is the word vector's dimension.

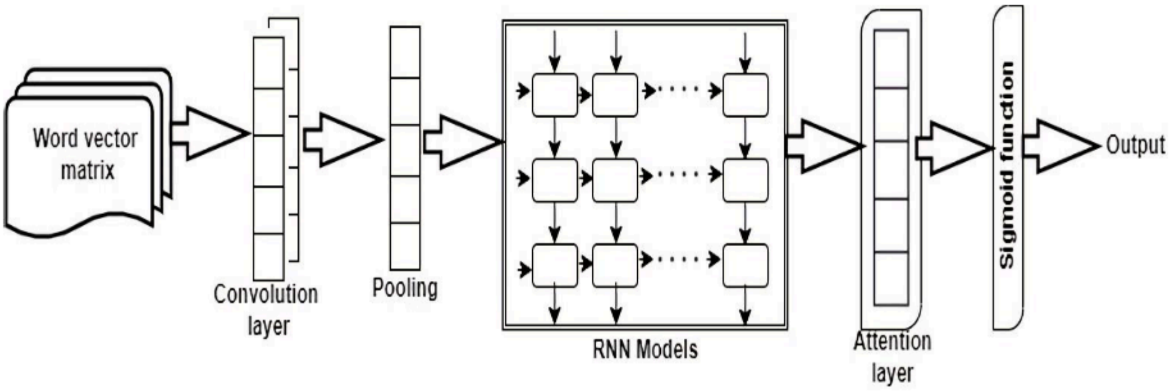


Fig. 2. A general architecture for text classification using a hybrid CNN and RNN models.

Zero padding and truncation techniques are used to form a uniform length of texts. Different texts may have different lengths resulting in a non-uniform length of texts. The uniform length texts are generated using zero appending and truncation based on the text size. Zero padding is used in texts smaller than l; otherwise, truncation. Therefore, every text will have the same dimension of l as defined in (2).

$$Z = \{W_1, W_{2,}W_3, \ldots\ldots, W_n\} \in X^{l*d} \qquad (2)$$

1D Convolution and max-pooling layer

Convolution layers are similar to the moving window across the matrix. Several such convolution layers having nonlinear functions like tan h, the rectified linear unit forms the CNNs. The output of CNNs is computed using convolutions over the input layer. Each layer is then applied to several kernels to aggregate their results, and local connections compute the output across the input layer. In the process of pooling the layers and at the training level, CNN realizes the size of the filter based on the task performed. SA receives input in words and documents structured as a matrix. Furthermore, every row of the matrix corresponds exactly with one token, which is generally a phrase. So, every row is a word-representation vector. This vector is often a low-dimensional, word-embedded form that indexes the word into a lexicon as an input. Filter glides over the matrix's complete text. As a result, the width of the filters and input matrix are the same.

Data from (2) is fed into a 1D convolution layer. To develop local features of n-gram (3) is used for convolution operation and filters whose range varies from 1 to N.

$$C_i^n = f(w^n \otimes R_{i:i+w-1} + b^n) \qquad (3)$$

where f is the nonlinear function, w^n is the weight matrix of the filter F_n given by $w \in X^{m*d}$ and bias function b^n. The text with length l generates the following feature map

$$c = [c_1, c_2, \ldots\ldots\ldots, c_l] \qquad (4)$$

The pooling layer captures the most significant information after the convolution procedure produces the feature maps for determining the local statistics. 1D max-pooling reduces or down-samples the input by converting individual kernel dimensions into a separate output of the determined number, and this is how the CNN model efficiently minimizes the total of features to avoid fitting problems while simultaneously reducing time and parametric complexities.

RNN models

LSTM (Long Short-Term Memory): The LSTM network is a type of RNN that has the ability to retain long-term dependencies. The network has hidden layers b_t varying with the time step t and is determined using a nonlinear transformation function [27]. This function operates on the present input a_t and previous hidden state b_{t-1}. The output of the network is computed using b_t. In addition, input, forget, and output gates are used to form memory cells in the LSTM. The input gate decides how much the new input will affect the memory cell; The forget gate controls how many prior data in the memory cell is lost, whereas the output gate controls how the memory cell affects the current hidden state. These gates operate on present input and b_{t-1}. LSTM network is realized as follows

$$i_t = \sigma(W_i\, b_{t-1} + U_i a_t + y_i) \tag{5}$$

$$f_t = \sigma\left(W_f\, b_{t-1} + U_f a_t + y_f\right) \tag{6}$$

$$\bar{C_t} = \tanh\,(W_c\, b_{t-1} + U_c a_t + y_c) \tag{7}$$

$$C_t = \ f_t \odot C_{t-1} + \ i_t \odot \ \bar{C_t} \tag{8}$$

$$O_t = \sigma(W_o\, b_{t-1} + U_o a_t + y_o) \tag{9}$$

$$b_t = \ o_t \odot \tanh C_t \tag{10}$$

where σ and $\odot$ are element-wise sigmoid and product function respectively, a_t and b_t are the input and hidden state vector at time t, respectively, different weight matrices of the gates for input is U, W and y are weight matrices for hidden state and bias vectors, respectively. The input vectors $(a_1, a_2, \dots \dots .\, a_n)$ are initialized randomly.

Bi-LSTM: Unlike LSTM, Bi-LSTM uses two hidden states to allow information to move in both directions: backward and forward, as depicted in Fig. 3. This may aid Bi-LSTM in better handling the problem. The input data comprising together previous and upcoming data will be kept by using these two-way routes, but the typical RNN model requires decay to include future information. The basic concept of Bi-LSTM is that two opposite pathways of an LSTM network are coupled to a single output. The forward LSTM stage obtains previous information, while the reverse LSTM state obtains subsequent information. This architecture aids the network's ability to remember information that came before and after it. In Bi-LSTM, the first layer's sequence output becomes the second layer's input, while the second layer's sequence output is the concatenation of the forward and backward layers' final unit outputs. The final output is given by

$$b = \left[b_{forward}, b_{backward}\right] \tag{11}$$

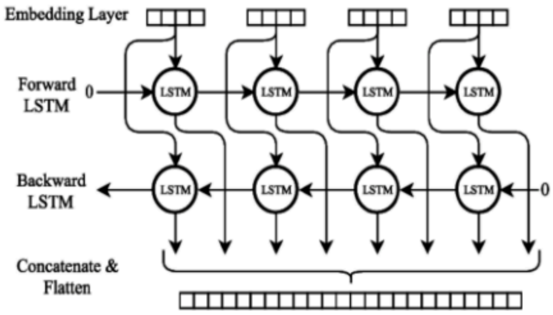


Fig. 3. Design of BiLSTM [33]

GRU (Gated Recurrent Unit): The GRU network is simpler and has fewer parameters than the LSTM. GRU uses an update gate to control how much information is updated in the hidden state and how much the previous hidden state influences the current hidden state, as well as a reset gate to control how information is updated in the hidden state and how much the previous hidden state influences the current hidden state. The GRU network is realized as follows

$$z_t = \sigma(W_z\, x_t + U_z h_{t-1} b_z) \tag{12}$$

$$r_t = \sigma(W_r\, x_t + U_r h_{t-1} b_r) \tag{13}$$

$$\widehat{h_t} = \tanh(W_h\, x_t + r_{t-1} \odot (U_h h_{t-2}) b_h) \tag{14}$$

$$h_t = (1 - z_t) \odot h_{t-1} + \ z_t \widehat{h_t} \tag{15}$$

where W_z, U_z, and b_z are update parameters and W_r, U_r, and b_r are reset parameters.

BiGRU: A forward GRU and a reverse GRU make up the BiGRU which are used to process both forward and backward data, respectively, as illustrated in Fig. 4 [34]. The forward and reverse GRU with the input x_t varying with time step t is given by

$$\overrightarrow{b_t} = \overrightarrow{GRU}\left(x_t, \overrightarrow{b_{t-1}}\right) \tag{16}$$

$$\overleftarrow{b_t} = \overleftarrow{GRU}\left(x_t, \overleftarrow{b_{t-1}}\right) \tag{17}$$

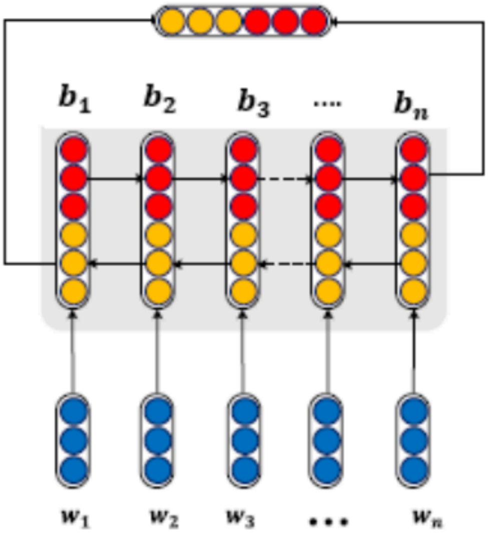


Fig. 4. Design of BiGRU [34]

Attention layer

The model employs a dense layer to connect each input to each output via weights. The output is produced using the sigmoid function in the last layer. The mean of the randomized findings is converted into 1 and 0 forms. To determine the positive (1) and negative (0) sentiment (18) is used.

$$\alpha = sigmoid\ (wx + \alpha) \tag{18}$$

where α is the offset.

Experiment

The proposed approaches are assessed in this segment for text sentiment categorization. The dataset used for this analysis is SST-2 and movie review.

Dataset

To validate the suggested technique, it was trained on two datasets. A binary labeled version of the Stanford Sentiment Treebank (SST-2) dataset is one of the datasets [35]. The SST-2 dataset was also used to compare the model's performance to earlier sentiment classification research. The second dataset is movie reviews (MR). Each review has one sentence describing a positive or negative review about the movie [36].

Metrics

As explained below, precision, recall, and F1-score are the standard performance metrics used for proposed model evaluations.

$$Precision = \frac{True\ positive}{True\ positive + false\ positive} \tag{19}$$

$$Recall = \frac{True\ positive}{True\ positive + false\ negative} \tag{20}$$

$$F1 - score = \frac{(2)(Precision)(Recall)}{Precision + Recall} \tag{21}$$

4 Experimental results

The experiment includes different models such as CNN, LSTM, BiLSTM, CNN LSTM based, CNN GRU, and CNN BiGRU based on SST-2 and movie review datasets. Word2Vec and GloVe are two pre-train word vector approaches that have found pioneer use in word vectorization and are used to initialize word vectors. The effectiveness of several models is shown in Table 1 and Fig. 5. Further, the analysis considers other conventional algorithms for comparison using the SST-2 dataset. The models considered are k means, KNN, decision tree, random forest classifier, logistic regression, and associated performances are 60%, 63%, 66%, 71%, and 61%, respectively, as depicted in Fig. 6.

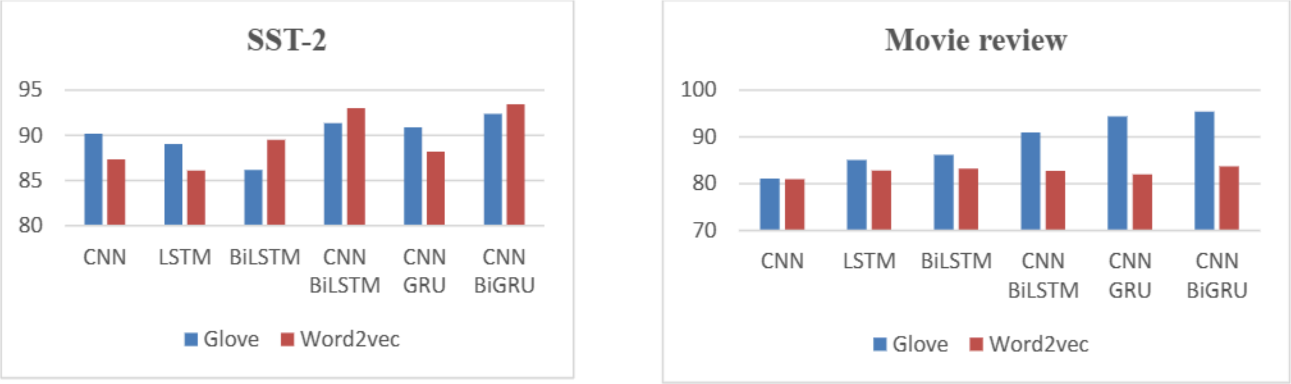
The performance slightly varies with the different word vector techniques. The findings of the experiments revealed that the CNN or RNN models alone would not improve the system's performance. The combination of CNN BiLSTM, CNN BiGRU with word2vec provides 93.02% and 93.44% F1-score, respectively, on SST-2 datasets. On the contrary, CNN BiGRU 95.42% on movie review datasets. By observing Table 1, it can be inferred that not all models show better performance on the same datasets. Experimentation of different models with the different with glove word embedding provides the highest performance of word embedding techniques will reveal the overall performance of that particular model combinations.

5. Conclusion

By combining the framework of CNN with improved RNN models, this paper attempts to present a unique sentiment classification model. The proposed approach supports the model in effectively classifying text sentiment by collecting both locally and globally dependencies in the framework of words. The models are trained and evaluated on SST-2 and movie reviews datasets. Therefore, the model effectively classified text sentiment over both datasets. The experiment's findings supported the model's effectiveness.

Table1. Performance of different models.

Dataset	Models	Glove			Word2vec		
		Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
SST-2	CNN	89.4	91.01	90.20	88.01	86.7	87.35
	LSTM	89.7	88.4	89.05	90.5	82.09	86.09
	BiLSTM	83.9	88.6	86.19	87.3	91.8	89.49
	CNN BiLSTM	90.9	91.8	91.35	94.7	91.4	93.02
	CNN GRU	89.9	91.9	90.89	87.5	88.9	88.19
	CNN BiGRU	91.3	93.5	92.39	92.6	94.3	93.44
Movie review	CNN	81.3	80.9	81.10	80.7	81.2	80.95
	LSTM	84.9	85.3	85.10	82.3	83.3	82.80
	BiLSTM	85.6	86.7	86.15	83.8	82.7	83.25
	CNN BiLSTM	90.32	91.65	90.98	82.4	83.1	82.75
	CNN GRU	94.9	93.9	94.40	83	81	81.99
	CNN BiGRU	95.14	95.7	95.42	84.5	82.9	83.69



(a)

(b)

Fig. 5. Performance analysis of different models with the dataset (a) SST-2, and (b) movie review

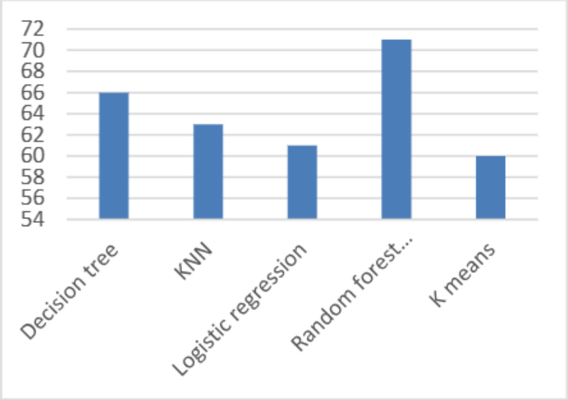


Fig. 6. Performance of conventional algorithms on SST-2 dataset

References

[1] Ji, P., Zhang, H. Y., & Wang, J. Q. (2018). A fuzzy decision support model with sentiment analysis for items comparison in e-commerce: The case study of http://PConline. com. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(10), 1993-2004.

[2] Zeng, D., Dai, Y., Li, F., Wang, J., & Sangaiah, A. K. (2019). Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism. Journal of Intelligent & Fuzzy Systems, 36(5), 3971-3980.

[3] Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (2017). Affective computing and sentiment analysis. In A practical guide to sentiment analysis (pp. 1-10). Springer, Cham.

[4] García-Moya, L., Anaya-Sánchez, H., & Berlanga-Llavori, R. (2013). Retrieving product features and opinions from customer reviews. IEEE Intelligent Systems, 28(3), 19-27.

[5] Reyes, A., & Rosso, P. (2012). Making objective decisions from subjective data: Detecting irony in customer reviews. Decision support systems, 53(4), 754-760.

[6] Weichselbraun, A., Gindl, S., & Scharl, A. (2014). Enriching semantic knowledge bases for opinion mining in big data applications. Knowledge-based systems, 69, 78-85.

[7] Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. arXiv preprint cs/0409058.

[8] Cho, H., Kim, S., Lee, J., & Lee, J. S. (2014). Data-driven integration of multiple sentiment dictionaries for lexicon-based sentiment classification of product reviews. Knowledge-Based Systems, 71, 61-71.

[9] Chen, L. S., Liu, C. H., & Chiu, H. J. (2011). A neural network based approach for sentiment classification in the blogosphere. Journal of Informetrics, 5(2), 313-322.

[10] Dang, Y., Zhang, Y., & Chen, H. (2009). A lexicon-enhanced method for sentiment classification: An experiment on online product reviews. IEEE Intelligent Systems, 25(4), 46-53.

[11] Moraes, R., Valiati, J. F., & Neto, W. P. G. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. Expert Systems with Applications, 40(2), 621-633.

[12] Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. Advances in neural information processing systems, 28, 649-657.

[13] Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1253.

[14] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[15] Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. (2019). Sentiment analysis of comment texts based on BiLSTM. Ieee Access, 7, 51522-51532.

[16] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554.

[17] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[18] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

[19] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[20] Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning (pp. 160-167).

[21] Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. Expert Systems with Applications, 57, 117-126.

[22] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[23] Huang, F., Li, X., Yuan, C., Zhang, S., Zhang, J., & Qiao, S. (2021). Attention-emotion-enhanced convolutional LSTM for sentiment analysis. IEEE

Transactions on Neural Networks and Learning Systems.

[24] Wu, O., Yang, T., Li, M., & Li, M. (2020). Two-Level LSTM for Sentiment Analysis With Lexicon Embedding and Polar Flipping. IEEE Transactions on Cybernetics.

[25] Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820.

[26] Elman, J. L. (1990). Finding structure in time. Cognitive science, 14(2), 179-211.

[27] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[28] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.

[29] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.

[30] Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014, April). Learning semantic representations using convolutional neural networks for web search. In Proceedings of the 23rd international conference on world wide web (pp. 373-374).

[31] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of machine learning research, 12(ARTICLE), 2493-2537.

[32] Tam, S., Said, R. B., & Tanriöver, Ö. Ö. (2021). A ConvBiLSTM Deep Learning Model-Based Approach for Twitter Sentiment Classification. IEEE Access, 9, 41283-41293.

[33] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, 45(11), 2673-2681.

[34] Zhou, J., Huang, J. X., Chen, Q., Hu, Q. V., Wang, T., & He, L. (2019). Deep learning for aspect-level sentiment classification: survey, vision, and challenges. IEEE access, 7, 78454-78483.

[35] https://www.kaggle.com/atulanandjha/stanford-sentimenttreebank-v2-sst2.

[36] https://www.cs.cornell.edu/people/pabo/movie-review-data/.

Authors' Profiles

Usha G R received the BE degree in Information Science and Engineering from SJMIT, Chitradurga in the year 2007, and MTech in Computer Science and Engineering from UBDTCE, Davanagere in 2010. She has 11 years of teaching experience. She organized various workshops in the department and also published many papers in reputed journals and international conferences. Her subjects of interest include Big Data Analytics, Machine Learning and Artificial intelligence.

J V Gorabal received BE in Computer Science & Engineering from BEC, Bagalkot, MTech in Computer Science & Engineering, KBNCE, Kalburgi and PhD in Computer Science & Engineering, JNT University Anantapuram respectively. He had 25 years of teaching experience. Currently, working as Professor in the Department of Computer Science & Engineering, ATME College of Engineering, Mysuru. He published many papers in national and international journals and attended conferences and workshops. His current research interests include Image Processing and Artificial Intelligence.