Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 10, No. 2, October : 2023

THE COMPARATIVE ANALYSIS OF DIFFERENT DEEP LEARNING MODELS FOR THE SARCASM DETECTION TASK IN SENTIMENT ANALYSIS

Adepu Rajesh, Research Scholar Department of Computer Science and Engineering, School of Engineering and Technology, Sardar Patel University, Balaghat, Madhya Pradesh, India
Tryambak Hiwarkar, Professor and Dean Department of Computer Science and Engineering, School of Engineering and Technology, Sardar Patel University, Balaghat, Madhya Pradesh, India

Abstract
During the previous few years, the textual with opinions are utilized over different social media platforms through Internet. Sentiment Analysis (SA) is exploited to investigate the opinionated text. It assist to identify (textual) the sentiments or emotions in the text expressed by the author. Various challenges are witnessed, and detection of Sarcasm fall into the main challenge. In Sarcasm, the message posted on particular topic persist the conflicts related to the context, and lead to ambiguity. Recently, we witness lots of advancements in deep learning technologies, and especially in the field of sentimental analysis. Moreover, the contributors highlighted the improvements in detection of sarcasm. Among various types of neural networks, utilization of convolution neural networks (CNNs), Long Term Short Memory (LSTM) and other models, namely Attention, Gated Recurrent Unit (GRU), Bi-directional LSTM (BiLSTM) by attention or no attention, achieved good performance metrics in the task of sarcasm detection. Here, we propose the comparative analysis based on various recent deep learning models i.e. an application of deep models for detection of sarcasm and comparative analysis among their performances.

Keywords: Sentiment analysis, deep models, comparison, sarcasm detection, text analysis

1. Introduction

Sentiment analysis comprises the study of various data generated on the social media related to different peoples expressed in the form of text, images, and videos consisting different sentiments or emotions. To extract, process, and analyze the various hidden facts in the data or viewpoints, natural language processing (NLP) is the most suitable technique to adopt. The analysis of text, linguistic information, context, affective states, and subjectivity can be easily identified and quantified using NLP and deep learning (DL). In NLP, textual data and data within the images is the active area of research.

Sarcasm detection (SD) is one of the identified challenge in SA, and attracted many researchers and academicians. SA is often mislead due to the existence of some words, having strong polarity but utilized sarcastically, i.e. other polarity was anticipated. Sarcasm may introduce uncertainty, while sarcasm isn't essentially ironic. Mostly, sarcasm is noticed in spoken words, sarcasm is characterized primarily by spoken intonation, and mostly depend on the context. Because of the inherent ambiguity of sarcasm, it can be difficult to even determine whether a statement is sarcasm. It is difficult for humans to take the decision for the utterance to categorize into sarcastic or normal.

SD is narrowed research branch in NLP, i.e. SA wherein sentiment detection is secondary task but more attention to determine the sarcasm in the input data. Detecting sarcasm automatically is useful for opinion mining and reputation management. SD benefits in SA by developing the system for automatic detection of sarcasm, can be considered a research problem.

The correct identification of sarcasm is the important aspect in sarcasm detection. In daily communication, peoples use sarcastic word regularly while arguing and venting. Thus, it is vital to determine the context of conversation by identifying the sarcasm. To detect the sarcasm in people's regular conversation it is necessary to process the words and tone. Hence, it is two stage process of sarcasm detection.

In this work, we investigate words in the sentence to detect the sarcasm. Precisely, we chosen online forum namely, Reddit [1] wherein users of the system create posts or comments and usually reply to another user's comments or posts. Input supplied to proposed model is a set of reply comments which leads to result comment, recurrent neural network (RNN) predicts the comments is sarcastic or non-sarcastic. The dataset is unique consisting massive amount of sarcastic comments with suitable annotation. The comparison using different DL models for SD is explored in this work. As per the literature, the authors targeted various machine learning (ML) algorithms or designed the specific DL model for SD. But, in this paper, we adopted different DL algorithms to study the behavior for SD. This is the first study to perform the comparative analysis of different DL models for SD.

The contributions targeted in the proposed work are listed as follows;

1. To study various DL models to determine the sarcasm in the input data.
2. To implement and test the different deep models on different datasets for sarcasm detection.
3. To perform the performance comparison among the various deep models.

Paper organization are as - In section 2, we explained the literature in the field of SA specifically, SD. Section 3, provides the information of dataset used and proposed methodology for the detection of sarcasm. Section 4, highlights results. Finally, paper is concluded in section 5.

2. Literature Review

Identification of figurative language expressions in short texts is the difficult and unresolved task in NLP, primarily due to nature of content are metaphorical and contradictory, and sarcasm is one of the main figurative expressions. A CNN network using embeddings technique is proposed in [2] wherein model captures both the relevant aspects of the content and the relevant contextual information about the author from the textual data. To represent the content, pre-trained word embeddings was adopted, further this is supplied to a convolutional layer to extracts the high-level features. Soujanya Poria [3] developed a model consisting of various models for personality detection, emotion and sentiment detection. Here, each model was trained on different benchmarking datasets, further, all these pre-trained models helped to extract sarcasm features associated with sarcasm datasets.

Aniruddha Ghosh [4] further combined CNN with LSTM as Deep Neural Networks (DNNs) applied to map features to more independent space. A DNN (fully connected) layer is added after LSTM network, the better classification results are obtained by mapping i.e. features are mapped to output space. In [5], neural network model is built on pre-trained transformer-based model, further enhanced by employing a recurrent CNN (RCNN). Here, authors achieved minimum data preprocessing.

In [6], a CNN-LSTM based neural network with word embeddings on word-vector encodings for headlines of news is developed to identify the headline is genuine or sarcastic. Further, the same neural network is applied to test dataset of real and sarcastic news headlines, and achieved an accuracy of 86.16%. Currently, Multimodal functionality fails to influence and extract the context from different modalities while conversation. Aman Shenoy [7] proposed RNN-based network that avoid various drawbacks, and preserves conversation context, speaker's emotions and interlocutor states, and outperforms the other models on benchmarking datasets with good accuracies.

In [8], various methods such as rule based, statistical, and deep learning are surveyed for the task of sarcasm detection. The detailed features utilization for the different sarcasm detection techniques are presented from the past to recent trends. In [9], the ML techniques are also investigated for SD task. In [20], RNN is employed for the microblog data in Chinese language to identify the sentiments. Here, word-vector from the sentence is constructed using distributed word, consisting the word's sequence features and semantic features. Thus, good performance is achieved for the prediction of sentiments.

3. Methods and Materials

3.1 Dataset and Characteristics

In this work, a twitter dataset by 'Ghosh' which is widely used for sarcasm detection is utilized [4]. In Twitter, hashtags (#) are utilized to express user's intentions or emotions. The users can self-declare

the sarcasm which becomes the hint for retrieval, i.e. #sarcasm, to gather such tweets, we crawled on Twittersphere. Here, simple heuristic omitted such users which persist sarcasm but it is not mentioned explicit (i.e. #sarcasm), thus, LSA-based method could be adopted to cover listed hashtags (e.g. #sarcastic, #sentiment, etc.). Ghosh collected tweets based on the profile of user with strong prejudice related sincerity or sarcasm. In order to develop dataset for sarcastic tweets, authors can accumulated different tweets consisting one or more positive indicators of sarcasm, moreover, indicators are removed in tweets, when tweets not containing positive indicators of sarcasm, i.e. tweets are categorized as non-sarcastic.

This particular dataset consists of around 55k tweets and 47% of the tweets are labelled as sarcastic. The dataset is divided into three parts - 70% dataset is utilized to train the models, 15% dataset is utilized to test the models, and 15% dataset is utilized used for validation.

3.2 Deep Learning Models

3.2.1 CNN

CNN [10] is a type of artificial neural network (ANN), utilizing perceptrons for learning from the input data. Like, ANNs, a CNN consist of various layers such as input layer, hidden layers, and output layer. In addition, CNNs considered to be a regularized forms of multilayer perceptrons (MLP) wherein MLP usually dense layers or networks, i.e. every neuron in previous layer acquire the connections in next layer with all neurons. This "fully-connectedness" effect on the networks could be overfitting of data. Thus, regularization is confirmed by adding thresholding mechanism corresponding to weights to reduce the losses by using appropriate loss function. In CNNs, various methods are utilized for regularization – 1) hierarchical pattern extraction from data, and 2) accumulate complex patterns from simpler and smaller patterns. Therefore, CNNs are less complex in terms of complexity and connectedness, as compared with ANNs.

It is identified that CNNs apply less pre-processing in comparison with different image classification methods i.e. network learns filters, hand-engineered in conventional algorithms. This is the major advantage in CNNs for feature engineering.

In CNN, x represent input layer consisting n entries. Every entry is characterised by a vector of d-dimension. Convolution Layer represent features learning when sliding w-grams. Let, $x_1, x_2, \ldots, x_n$, be the sequence of n entries, and concatenated embeddings of w ($x_{i-w+1}, \ldots, x_i$) be a vector $c_i \in R^{wd}$, here, $0 < i < s+w$ and w be a filter width. Padding (zero pad) is performed on embeddings, x_i , $i < 1$ or $i > n$. Further, $q_i \in R_d$ is generated for w-gram, $x_{i-w+1}, \ldots, x_i$ by applying convolution weights, $W \in R^{d \times wd}$.

$$q_i = tanh\,(W \times c_i + bais) \qquad (1)$$

where, bias $\in R^d$.

In max pooling - all w-grams q_i (i = 1 to $s+w-1$), utilized to produce the depiction of x (input sequence) by max pooling: $x_j = max\,(q_{1,j}, q_{2,j}, \cdots)$ (j = 1 to d).

3.2.2 LSTM

Long short-term memory (LSTM) [11] is a type of RNN utilized under DL models wherein it consisting feedback connections. It process the sequence of data points such as text, speech, video, etc. LSTM could be applicable for various tasks – 1) anomaly detection, speech recognition, weather prediction, stock prediction, and handwriting recognition, etc. The LSTM can be applied for classification, processing and performing predictions on temporal data, sometime lags can be identified in temporal data in the duration of important events. Further, LSTMs are characterized to handle the problem of vanishing gradient, the drawback encountered in basic RNN. Thus, LSTM are preferred over hidden Markov models (HMM) and RNN in varieties of applications.

A LSTM module comprises – 1) input gate, 2) output gate, and 3) forget gate. Generally, this module is called as cell. The data flow in the cell is fully controlled by these gates wherein cell stores values for arbitrary intervals while processing the input. Figure 1 depict the construction of LSTM cell. The main responsibility of LSTM cell is to track dependencies amongst the elements present in input sequence. Input gate controls the flow of new value, forget gate determines the availability of data as per the context, and output gate produce the value utilized to compute final outcome based on

activation applied on LSTM unit. Activation function such as tanh and logistic sigmoid are the regular choice in LSTM gates.

In LSTM, various connections can be seen and mostly all are recurrent connections. The gates operations are visualised in the form of weights calculation amongst these connections during learning phase i.e. training. The gates computations are represented as follows;

$$inp_t = \sigma(W_i[h_t-1, x_t] + b_i)$$
$$frg_t = \sigma(W_f\,[h_t-1, x_t] + b_f\,)$$
$$q_t = tanh(W_q[h_t-1, x_t] + b_q)$$
$$out_t = \sigma(W_o[h_t-1, x_t] + b_o)$$
$$cout_t = f_t \odot c_{t-1} + i \odot_t q_t$$
$$hid_t = o_t \odot tanh(c_t)$$

Where, x_t = input vector, hid_t = hidden state vector, inp_t = input gate vector, frg_t = forget gate vector, out_t = output gate vector, q_t = cell input vector, $cout_t$ = output of cell, $\odot$ = element-wise product, σ = sigmoid function.

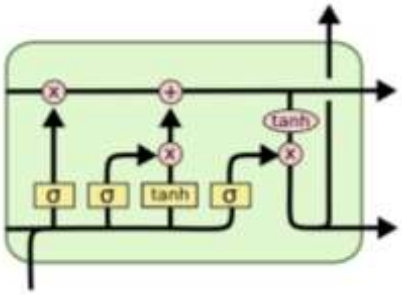

Figure 1: LSTM Cell

3.2.3 BiLSTM

Typical RNNs have a problem of only taking previous context into account when it is very useful to also look into future context as well. Thus, Bi-directional RNNs are adopted and can travel in all inputs (i.e. past and future) to extract more features for the prediction at output layer. To do so, Bi-directional LSTM model is utilized because it has all the advantages of BiRNNs, and vanishing gradient issue [12] is easily handled. BiLSTM is shown in Figure 2 for more detailed understanding.

In a Bi-directional LSTM, one LSTM processes the text sequence from start to end of text. Another LSTM processes the text sequence in backward direction from end to start. There is no interaction between the two different types of neurons [13].

$$y_t = \sigma(hid\,\overrightarrow{}_t, hid\,\overleftrightarrow{}_t)$$

y_t = output, $hid\,\overrightarrow{}_t$ = forward layer output, $hid\,\overleftrightarrow{}_t$ = backward layer output

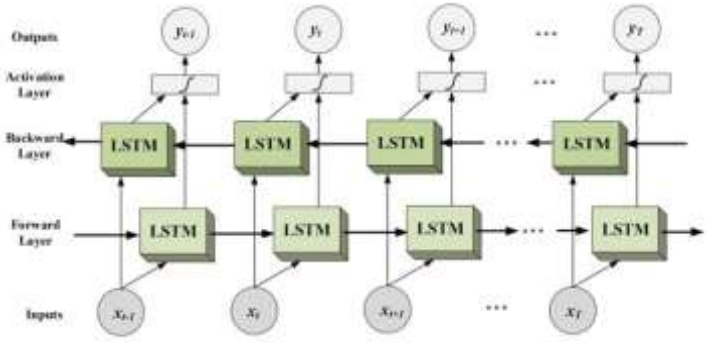

Figure 2: Bi-directional LSTM Model

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 10, No. 2, October : 2023

3.2.4 Attention

Attention utilizes a context vector (CV) for mapping inputs in source and target [14]. The CV preserves information pertaining to all hidden states belongs to input cells and maps to current targeted output (see Figure 3). Thus, model is capable to "attend to", and identifies complex relationship amongst source inputs and target, during learning phase. Attention mechanism produce an attention score $as_{i,t}$ for each t word in sentence i, represented as follows;

$$as_{i,t} = \sigma(Wh_t\ c)$$

σ = activation function

Further, weight probability $a_{i,t}$ is calculated of each h_t as;

$$a_{i,t} = exp(as_{i,t}) / \sum^{T}_{K=1} exp(as_{i,k})$$

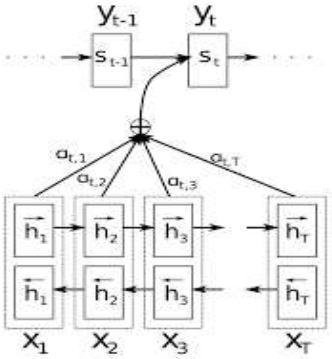

Figure 3: Attention Model

3.2.5 CNN with LSTM Model

The idea behind this model is that CNN advantageous due to hierarchical pattern extraction from data and accumulate complex patterns from simpler and smaller patterns. Hence, CNNs are less complex in terms of complexity and connectedness. We conclude that CNNs are effectively model hierarchical patterns corresponding to local features towards to determine global features, further, which helps in identifying the context. The word vectors stores sentence representation, learned in training further, max pooling is employed to extract the prominent features for features generation.

A sarcastic text assumed to be a sequence of word combination or textual signals. Here, RNN is most favourable model for temporal data with temporal memory, helps in storing temporal contextual details. Various implementations based on RNNs and LSTMs witnessed the efficient training and handling of exploding or vanishing or gradient issue using backpropagation through time.

Subsequently, performance of LSTM can be enhanced by allowing better features. The convolution networks reduces frequency variation by applying different convolutional filters, and composite feature map is generated from discriminating words in sequence, further, it is supplied to LSTM layer. The LSTM identifies long term dependencies among the sentences and accomplished by computing gate equations (i.e. input gate, forget gate and output gate) [4]. We implemented [15] for the task of sarcasm detection to evaluate the performance among other models.

3.2.6 CNN with BiLSTM Model

The CNN and BiLSTM is developed based on advantages of a BiLSTM model but with an added layer of convolution filters before the BiLSTM in order to generate efficient feature map from discriminating words in sequence. To determine the performance of aforementioned model for sarcasm detection with the other models, we designed the model presented in [16].

3.2.7 BiGRU with Attention Model

This model uses a Bi-directional GRU with attention mechanism. The GRU is like a LSTM wherein forget gate is available along with fewer parameters compared to LSTM, due to non-availability of output gate. Further, it is identified that performance of GRU and LSTM is similar for certain tasks. In addition, GRU shows the better performance for datasets with smaller and less frequent data. The composition of GRU for input (x) is represented [17] as follows:

$$r = \sigma(x_t U^r + h_{t-1} W^r)$$
$$z = \sigma(x_t U^z + h_{t-1} W^z)$$
$$s_t = tanh(x_t U^s + (h_{t-1} \circ r) W^s)$$
$$ht = (1 - z) \circ s_t + z \circ h_{t-1}$$

Where, r and z are gates, $U \in R^{d \times h}$, $W \in R^{h \times h}$ are parameters, $\circ$ is Hadamard product, $x_t \in R^d$ is the token in x, hidden state $h_t \in R^h$, history encoded in $x_1, \cdots, x_t$.

Attention mechanism augments into the network, an additional component or circuit, thus, learning is enhanced during training the model via back-propagation. Deterministic attention namely soft attention, applied when weights are computed using softmax, moreover, attention module produces output, the weighted sum corresponding to each location. The word-level attention is applied over words having closer semantics related to meaning of sentence. Here, output computed is weighted combination corresponding to input states, but not the output of last state [12].

3.2.8 Hybrid Model

This is a very recent model which is a combination of convolution filters, BiLSTM and attention layer model. A BiLSTM is two network consisting forward as well as backward LSTMs. So, forward LSTM works in forward direction of input data whereas backward LSTM process the data in reverse direction. The annotated words are obtained by concatenating the hidden states (h_i) in forward and backward direction. Thus, annotation h_j consisting summarization of forward words and preceding words.

At particular time step, LSTM is able to represent information efficiently, here, annotation performed at time step focused around the words in input. Here, hidden state consisting details of complete input text wherein strong correlation is established amongst the word in the text. The CV c is calculated as a weighted sum of all these annotations.

$$c = \sum \alpha_i h_i$$

Here, α_i - weight/attention corresponding to h_i calculated using Softmax for all h_i. The each h_i score is calculated in forward direction and output score is generated. Further, this CV c is concatenated with the CNN model output. Then, all feature vectors are supplied to MLP, thus, binary classification in terms of probability is produce and sentence or text identified as sarcastic or non-sarcastic. In [18], the hybrid model is discussed wherein it utilize the CNN, LSTM and attention for sentiment analysis. Here, we adopted this model to detect the sacrum detection to evaluate the behaviors in comparison with other DL models.

4. Results and Discussion

The dataset is preprocessed using keras (TF 2.0) library [19], keras is most popular library for neural network design implemented in Python. The fast experimentation and calculations are achieved in keras for DNNs, thus, offers user-friendly, modular and extensible implementation. The keras not only support ANN but has full support for CNN and RNNs. The raw textual data cannot be directly fed into any deep learning models, thus, we need to apply the suitable encoding technique. Moreover, embedding layer could be employed for textual data in keras. The data preparation in keras is performed via Tokenizer API. The word embedding (e.g. GloVe) is performed by assigning random weights and embedding is learned for all words in training dataset. Model training is accomplished on aggregated global word-word co-occurrence informations obtained from the corpus, and resultant word vector space contains interesting linear features. We utilized GloVe pretrained vectors, which is trained on 400k vocabulary collecting 6B tokens, in which each word is characterized as a 300 dim vector.

The preprocessing modules for data cleaning are – 1) Tokenize - creates a vocabulary using words, and 2) Lemmatization - group together different forms of words. After the dataset is processed, it is fed to each of the deep learning models. The Callbacks tools of keras is useful to view - internal states and model statistics, during training. Other applications of callbacks includes the ability to dynamically change the learning rate (LR) of the optimizer during the course of training [11], model checkpoint and early stopping at minimum loss. Reduce LR on plateau is a callback feature that reduces LR when

no improvement in metric. Reducing LR benefits models, once no learning is observed, thus, callback monitors metric and if no significant improvement is identified for chosen number of epochs, then LR is reduced.
The parameters utilized for the performance measurement are as follows:
Precision - The fraction of correct positive classification to correct positive classification and incorrect positive classification. Precision can be expressed as;

$$Precision = TP / (TP + FP)$$

Where, *TP* – true positive, *FP* – false positive.
Recall - The fraction of correct positive classification to correct positive classification and incorrect negative classification. Recall is also known as sensitivity. Recall can be expressed as;

$$Recall = TP / (TP + FN)$$

Where, *TP* – true positive, *FN* – false negative.
F1-score – It is the harmonic mean corresponding to precision and recall.

$$F1\text{-}score = 2*(precision*recall)/(precision + recall)$$

Accuracy - Indicates, among all the test datasets, how many of them are captured correctly by the model compared to their actual value.

The comparative analysis based on the various models are presentation in Table 1. Table 1 highlighted the accuracy for CNN, BiLSTM, and combination of CNN + LSTM / BiLSTM achieved the same accuracy of approximately 80%. Moreover, the accuracy (81%) is little improved by using attention mechanism with CNN and RNNs. This is the first of kind comparative analysis based on DL models for the sentiment analysis to detect the sarcasm from the textual data.
There are numerous vital hyper-parameters in proposed models, and suitable fine-tuning values. For all models, initial LR, $\alpha = 0.01$. The size of word vectors is 300, hidden units size in GRNNs is 256, and size of filters in Convolutional layer is 256. We trained all the proposed models on batch size = 256 and epochs = 50. The optimizer used is 'adam', it is an optimization technique to utilize to update weights iteratively while training data. The 'adam' accumulates best properties from RMSProp and AdaGrad to offer optimized weight, handling noisy problem through sparse gradients.
The loss function is 'Categorical cross entropy' is employed for loss calculation. It compares the distribution of predictions with true distribution, here, true class probability is set to 1, and 0, otherwise. Simply, one-hot encoding (vector) is applied for true class, loss is lower when model's output is closure to vector.
We also applied dropout = 0.4, wherein neurons are randomly selected and overlooked during training i.e. neurons "dropped-out" randomly. Thus, contribution of neurons is downstream temporally during forward pass and no weight update is performed on such neurons during backward pass. Further, network becomes non responsive to weights of dropped neurons. Hence, we can achieve network generalization and overcome the problem of overfit on training dataset.

Table 1: Comparison of Various Models

Model	Precision	Recall	F1-score	Accuracy
CNN [10]	0.778	0.799	0.788	0.798
BiLSTM [13]	0.779	0.823	0.800	0.805
CNN+BiLSTM [16]	0.806	0.773	0.789	0.804
CNN+LSTM [15]	0.800	0.769	0.784	0.800
BiGRU+Attention [17]	0.784	0.820	0.802	0.808
Hybrid Model [18]	0.810	0.804	0.807	0.814

5. Conclusions

Sarcasm detection researched significantly over preceding few years, demanding that to assess the individual work to identify the suitable combination of models for the task of SD. Currently, SD investigates the discovery of efficient features along with contextual information and its effects. This work studies and employs the deep learning models that have utilized in recent years to for sentiment analysis. We observe the performance for different models, moreover, the performance is evaluated based on accuracy, recall, precision and F1-score. It is found that the hybrid model outperform all other models followed by the Bi-directional GRU with attention model. As the hybrid model concatenates both CNN and BiLSTM features, thus, contains huge feature vector which is utilized for classification. Hence, hybrid model performs better compared with the state-of-the-arts.

Declarations

Conflict of interest

Authors declare no conflict of interest.

Informed consent

Informed consent is obtained from different individual, if required or necessary.

Data availability

Data is publically available on the Internet.

Funding

No funding was provided for this work.

6. References

[1] Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., & Blackburn, J. (2020, May). The pushshift reddit dataset. In Proceedings of the international AAAI conference on web and social media (Vol. 14, pp. 830-839).

[2] Amir, S., Wallace, B. C., Lyu, H., & Silva, P. C. M. J. (2016). Modelling context with user embeddings for sarcasm detection in social media. arXiv preprint arXiv:1607.00976.

[3] Poria, S., Cambria, E., Hazarika, D., & Vij, P. (2016). A deeper look into sarcastic tweets using deep convolutional neural networks. arXiv preprint arXiv:1610.08815.

[4] Ghosh, A., & Veale, T. (2016, June). Fracking sarcasm using neural network. In Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis (pp. 161-169).

[5] Potamias, R. A., Siolas, G., & Stafylopatis, A. G. (2020). A transformer-based approach to irony and sarcasm detection. Neural Computing and Applications, 32, 17309-17320.

[6] Mandal, P. K., & Mahto, R. (2019). Deep CNN-LSTM with word embeddings for news headline sarcasm detection. In 16th International Conference on Information Technology-New Generations (ITNG 2019) (pp. 495-498). Springer International Publishing.

[7] Shenoy, A., & Sardana, A. (2020). Multilogue-net: A context aware rnn for multi-modal emotion detection and sentiment analysis in conversation. arXiv preprint arXiv:2002.08267.

[8] Kumar, L., Somani, A., & Bhattacharyya, P. (2017). Approaches for computational sarcasm detection: A survey. ACM CSUR.

[9] Chimote A.K., Tandan S.R. (2019). Research on Detection of Sarcasm using Machine Learning Techniques. International Journal of Recent Technology and Engineering, 8:2S11.

[10] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics, 36(4), 193-202.

[11] Tang, D., Qin, B., Feng, X., & Liu, T. (2015). Effective LSTMs for target-dependent sentiment classification. arXiv preprint arXiv:1512.01100.

[12] L. H. Son, A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar and M.Abdel-Basset, Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model With Convolution Network, IEEE Access, vol. 7, 2019

[13] Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078.
[14] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
[15] Wang, J., Yu, L. C., Lai, K. R., & Zhang, X. (2016, August). Dimensional sentiment analysis using a regional CNN-LSTM model. In Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers) (pp. 225-230).
[16] Rhanoui, M., Mikram, M., Yousfi, S., & Barzali, S. (2019). A CNN-BiLSTM model for document-level sentiment analysis. Machine Learning and Knowledge Extraction, 1(3), 832-847.
[17] Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. arXiv preprint arXiv:1702.01923.
[18] Kumar, A., Sangwan, S. R., Arora, A., Nayyar, A., & Abdel-Basset, M. (2019). Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. IEEE access, 7, 23319-23328.
[19] Tensor Flow, https://www.tensorflow.org/guide/keras/custom_callback
[20] Zhang, Y., Jiang, Y., & Tong, Y. (2016). Study of sentiment classification for Chinese micro blog based on recurrent neural network. Chinese Journal of Electronics, 25(4), 601-607.