

Sarcasm Detection Using Multi-Channel Attention Based BLSTM On News Headline

Azika Syahputra Azwar

Bina Nusantara University

Suharjito (✉ suharjito@binus.edu)

Bina Nusantara University

Research

Keywords: Sarcasm, News Headline Detection, BLSTM sentence classification, Attention Based, Multi-Channel

Posted Date: September 4th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-63423/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Sarcasm Detection Using Multi-Channel Attention Based BLSTM On News Headline

Azika Syahputra Azwar^{1,*} and Suharjito¹

*Correspondence

suharjito@binus.edu

¹Computer Science Department,
Binus Graduate Program – Master
Computer Science, Bina Nusantara
University, Jakarta, Indonesia,
11480

Abstract

Sarcasm is often used to express a negative opinion using positive or intensified positive words in social media. This intentional ambiguity makes sarcasm detection, an important task of sentiment analysis. Detecting a sarcastic tone in natural language hinders the performance of sentiment analysis tasks. The majority of the studies on automatic sarcasm detection emphasize on the use of lexical, syntactic, or pragmatic features that are often unequivocally expressed through figurative literary devices such as words, emoticons, and exclamation marks. In this paper, we introduce a multi-channel attention-based bidirectional long-short memory (MCAB-BLSTM) network to detect sarcastic headline on the news. Multi-channel attention-based bidirectional long-short memory (MCAB-BLSTM) proposed model was evaluated on the news headline dataset, and the results-compared to the CNN-LSTM and Hybrid Neural Network were excellent.

Keywords: *Sarcasm, News Headline Detection, BLSTM sentence classification, Attention Based, Multi-Channel.*

Introduction

Text has become a way to express something and convey information from one individual to another because of its certainty and completeness of expression. Text classification has been widely applied in several ways including news classification, emotional analysis and an automatic question and answer system [1].

However, text classification is a fairly broad field and is not only limited to positive or negative classifications as has often been done in recent years because there is one type of text that has the opposite meaning of what is written, namely sarcastic text or sarcasm. "Sarcasm is a specific type of sentiment where a person expresses his negative feelings using positive words in his speech or exaggerates the positive words"[2].

Therefore, to be able to classify texts, researchers must need a deeper understanding of language, system dialogue, and skills to understand the context along with its content. The problem is that in the presence of sarcasm the polarity of the user's expression is reversed the meaning of the misclassification error, thus affecting the performance of the System [3].

This is a difficult task because of the complexity and ambiguity of natural language, where a word may have a different meaning, or several phrases can be used to express the same idea [4]. Because of this, classification of sarcasm texts is still quite difficult in many natural language processing systems because even for humans it is quite difficult to properly verify the occurrence of sarcasm in a statement. As described by Pozzi and colleagues [5], "The difficulty in recognition of sarcasm causes misunderstanding in everyday communication and poses problems to many NLP systems."

Many research have tried to identify sarcasm and have proposed various models, in which they have found that sarcastic sentences have certain biases with regard to the sentence itself [6] , while others make claims that this phenomenon is non-literal and its identification requires pragmatic knowledge [7]. Based on existing research, there are 2 types of methods that are often used for research in the field of Natural Language Processing (NLP), namely Convolutional Neural Network (CNN) and Long-term and Short-term Memory Neural Network (LSTM) [8]. In this paper, we present a multi-channel attention based BLSTM sarcasm detection technique in the news headlines that weight in the advantages of multi-channel mechanism, attention mechanism, multi-channel and bidirectional long short-term memory layer (BLSTM). Moreover, we compared this approach with other deep learning algorithms to show the performance and accuracy our model has reached so far. We applied our approaches to analyze sarcasm of news headline in news headline dataset.

In the rest of this paper, a brief of related works is summarized in “Related works” section. In “Methods/experimental” section, we represent our methodologies that we have implemented to recognize sarcasm in news headline, with the dataset used. In “Results and discussion” section, a brief discussion of the results is addressed. At the end, insights for the future and a short summary are presented.

Related Works

A lot of researches have been done lately in the domain of natural language processing. In [9] Jain, Kumar, & Garg proposed an RNN approach with Feature-rich CNN to improve the accuracy of sarcasm detection using a dataset of 3.000 sarcastic tweets and 3.000 non-sarcastic tweets in Hindi and English. This model consists of a preprocessing stage where the data is normalized by removing punctuation marks, special characters and regular expressions, then the data is stemmed to return the word to its basic form after the data is tokenized using GloVe for English and Hindi-SentiWordNet for Hindi then the result is an English vector. processed using the BLSTM method then the results enter the soft attention layer stage to see the level of semantic closeness in each word with the intent of the sentence from the processed data then the results are processed again to see the conceptual relationship pragmatically then the results of the English and Hindi vectors enter the convolution stage the layer to produce a feature map then enters the RELU Layer stage to overcome non-linearity in the previous process results then enters the pooling layer stage where the feature map is reduced in dimensions and takes the most influential features then enters The representation layer stages of the previous process are fed into fully connected softmax to calculate the probability of each word and classify the tweets as being sarcastic or non-sarcastic

In [10] Mandal & Mahto proposed the CNN-LSTM model approach added with word embedding to improve the accuracy of sarcasm detection by using a data headline dataset that has been categorized manually as many as 26.709 data and divided into 11.725 sarcasm and 14,984 non-sarcasm. The model used consists of a preprocessing stage where the data is normalized by removing punctuation marks, special characters and regular expressions then the data is stemmed to return the word to its basic form after the data is tokenized using a word embedding dictionary that has been created based on the 10.000 words that appear most frequently in the data. sarcasm then the next stage of convolution where the vector is processed at 1-D convolution layer with 32 filters and 7 kernel sizes so that the filter will produce 7 word combinations then the results will go to the 1-D max pooling layer where the results of each kernel are converted into 1 output based on the highest value in each kernel then the results are processed again in the 1-D convolution layer stage after which the data is processed using the CNN-LSTM method with a dropout of 0,5 then in the final stage the results are processed with a binary cross-entropy loss function. The evaluation method used is to measure the level of accuracy where an accuracy of 86.16% is obtained.

Then [11] Mehndiratta & Soni proposed the RNN, CNN model approach, and a combination of the two added with word embedding and hyperparameter tuning to increase the accuracy of sarcasm detection using the Reddit dataset which contains comments that have been categorized manually into the sarcasm category of 1.35 million data from a total of 533 million data. The model used consists of a preprocessing stage where the data is normalized by eliminating punctuation marks, special characters and regular expressions, then the data is stemmed to return the word to its basic form after the data is tokenized using the word embedding dictionary GloVe or fastText and then entered into the classification method using CNN. , LSTM, CNN-LSTM, and LSTM-CNN in which a hyperparameter is added in the form of an epoch and a special dropout after which a comparison report of the accuracy of each method is generated.

In [12] Kumar, Sangwan, Arora, Nayyar, & Abdel-Basset proposed a deep learning model approach called sAtt-BLSTM convNet which is based on a combination of soft attention-based bidirectional long short-term memory (sAtt-BLSTM) and convolution neural network (convNet) to improve sarcasm detection accuracy with 40.000 random data tweets labeled 15.000 sarcastic tweets and 25.000 non-sarcastic tweets. The model used consists of a preprocessing stage where the data is normalized by removing punctuation marks, special characters and regular

expressions, then the data is stemmed to return the word to its basic form after the data is tokenized using the GloVe word embedding dictionary and then enter the BLSTM method, which includes soft-attention to combine the results of 2 backward and forward outputs into one then the results enter the attention layer stage to see the level of semantic closeness in each word with the intention of the sentence from the processed data then the results enter the convolution layer stage to produce a feature map then enter the RELU stage The layer to overcome the non-linearity in the results of the previous process then enters the pooling layer stage where the feature map is reduced in dimensions and takes the most influential features then enters the representation layer stage the results of the previous process are entered into fully connected softmax to calculate the probability of each word and classification of tweets as sarcastic or non-sarcastic results.

In [13] Hiai & Shimada proposed an RNN approach with a Relationship Vector to improve the accuracy of sarcasm detection using a dataset of 21.000 sarcastic tweets and 21.000 non-sarcastic tweets. This model consists of a preprocessing stage where the data is normalized by removing punctuation marks, special characters and regular expressions, then the data is stemmed to return the word to its basic form after the data is tokenized using word2vec then the results are reprocessed using the Role Pair Relation Vector method to see the linkage between features. vector based on the previous training data process after which the results are processed using the RNN method with the BLSTM type with vector dimensions 200, 150 hidden, and an epoch size of 30 is added.

In [14] Xiong, Zhang, Zhu, & Yang proposed the Self-matching Networks and BLSTM approach with Low-Rank Bilinear Pooling to improve the accuracy of sarcasm detection using a dataset of 91.717 data. This model consists of a preprocessing stage where the data is normalized by removing punctuation marks, special characters and regular expressions then the data is stemmed to return the word to its basic form after the data is tokenized using GloVe then the results are processed using Self-Matching Network and BLSTM to produce both feature maps. then enter the Low-Rank Bilinear Pooling process to combine and calculate the sarcastic and non-sarcastic probabilities of the incoming data.

In [15] Misra & Arora proposed a Hybrid Neural Network approach to improve the accuracy of sarcasm detection using a news headline dataset of 29.709 data. This model consists of a preprocessing stage where the data is normalized by removing punctuation marks, special characters and regular expressions, then the data is stemmed to return the word to its basic form after the data is tokenized using GloVe then the results are processed using the Hybrid Neural Network to produce both feature maps then combined then calculated sarcastic and non-sarcastic probabilities from the data entered using softmax.

In [16] Kumar, Narapareddy, Srikanth, Malapati, & Neti proposed a BLSTM-based Multi-Head Attention approach to improve the accuracy of sarcasm detection using a dataset of 110.914 sarcastic reddit comments and 173.003 non-sarcastic reddit comments. This model consists of a preprocessing stage where the data is normalized by removing punctuation marks, special characters and regular expressions, then the data is stemmed to return the word to its basic form after the data is tokenized using GloVe then the results are processed using the BLSTM method with 100 vector dimensions, 100 hidden units, and add a dropout value of 0,5 then the results are processed on the Sentence Level Multi-Head Attention Layer to measure the importance value of each word based on the semantic factor after that the results are processed again in Auxiliary Features Concatenation where the initial data is extracted semantic, sentiment, and punctuation and combined. with the result data from the previous process to create a new representation of the data then the processing result is entered into the softmax layer to calculate the probability of each word and classify the data as sarcastic or non-sarcastic.

Some of this work contributions is the use of an end-to-end network that comprises proposed model steps: preprocessing, token vectorization, *BLSTM*, *Attention Layer*, *Pooling*, *Relu Layer*, and *Representation Layer*. We compare the results with other two deep learning approaches.

Methods/experimental

In this section, we will present the dataset used, and our methodologies to recognize sarcasm on headline news using Multi-Channel Attention Based BLSTM in addition to other two deep learning algorithms, which are CNN-LSTM, and Hybrid Neural Network.

Used dataset

We used the News Headline dataset provided by [15], that classify news headline into sarcasm and non-sarcasm. The dataset consists of 56.418 news headline, that 25.846 represent sarcasm and 30.752 represent non-sarcasm. We split our dataset into 45.352 news headline in the training set and 11.066 news headline in the testing set. We used 80% of the dataset for training and 20% for testing. The training datasets were used to train the classifier and to

optimize the parameters, while the test dataset (unseen to the model) was reserved to test the built model, to provide an indication of how good the trained model is. Also, we tried to split the data 70% for training and 30% for testing it gave us the same results which is 97.84%.

Data preprocessing

Before data are transfer to input layer, data are pre-processed to clean and transform the data for feature extraction [17]. The steps we followed are:

1. Converting the entire text in a document into a standard form (in this case lowercase or lowercase).
2. Cutting a document into parts called tokens and remove certain characters that are considered punctuation.
3. Removing stop words that do not contribute much to the content of the text, such as the words “and”, “i”, “you”, “with”, “she”, “he”, and others.
4. Stemming process that the return of an affixed word to its root form.

Proposed Multi-Channel Attention Based BLSTM

The proposed deep learning model uses 6 layers: the input layer, embedding layer, BLSTM layer, attention layer, max pooling layer, ReLu Layer, concatenate layer, and representation layer. Fig. 1 depicts the architecture of the proposed multi-layer model. The details of each layer are provided in the subsequent subsections.

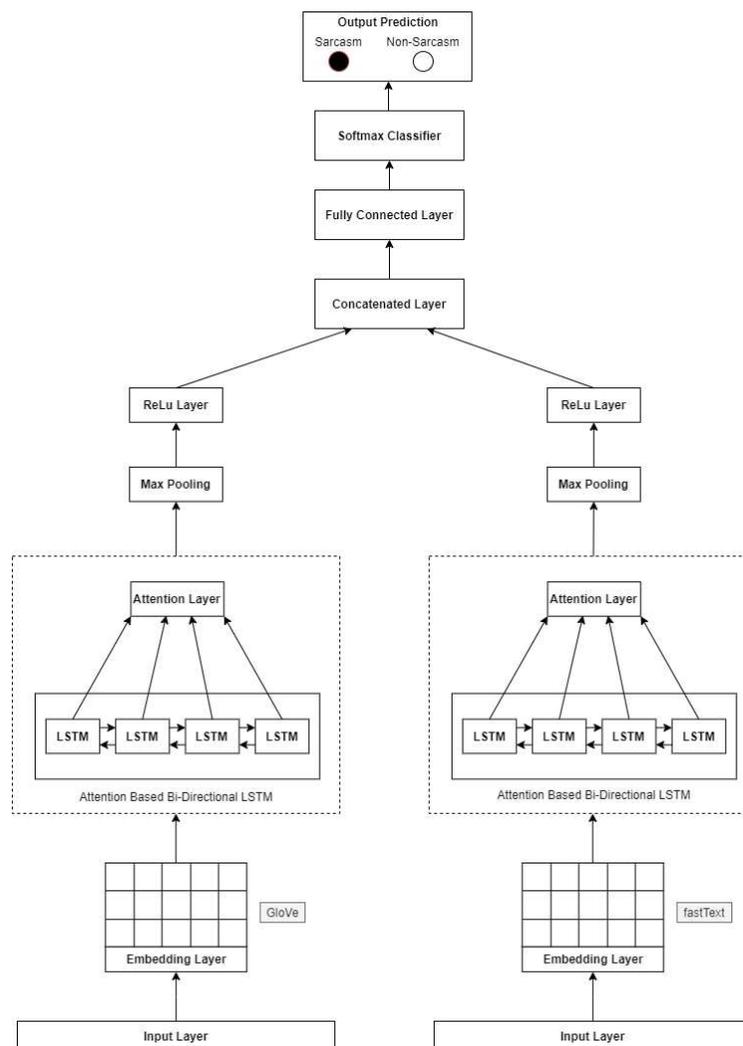


Figure 1. System architecture of the proposed Multi-Channel Attention Based BLSTM.

1. Input Layer

The news headline after pre-processing are fed to the input layer. The input layer is connected to the embedding layer, which builds word embeddings using GloVe and fastText.

2. Embedding Layer

The embedding layer maps the input into real-valued vectors using encoding from look-up tables. Word embeddings facilitate learned word representations. The benefits of extracting features based on word embedding to detect sarcasm have been recently reported [18]. In this study, to build word embeddings, GloVe and fastText, which generates a word vector table, is used. GloVe and fastText is a count-based model of representing words by feature vectors. This log-bilinear model studies the relationship of words by counting the number of times they co-occur. Thus, this model aids in mapping all the tokenized words in each news headline to their respective word vector tables. Proper padding is performed to unify the feature vector matrix. That is, if the total number of given tweets is Z and there is a tweet X with t tokens, generation of a word vector with dimension d of the word vectors is completed using GloVe and fastText. Thus, for all Z , each t in X is mapped to its respective V . After this mapping, each X is expressed as a vector of the word embeddings = concatenation (E). Thus, the feature vector matrix is obtained as shown in (1).

$$F = U + B + T + P + E \quad (1)$$

where C is the concatenation operator of the vector. The tweets are of varying length, so to unify the feature vector matrix representation of news headline, the news headline with the maximum length in the given corpus are used as a threshold value. This is done basically to fix the length of the news headline matrix. Hence, for all the news headline that were shorter than this threshold, zero padding was performed. This matrix was finally fed as input (i.e., F) to the BLSTM layer.

3. BLSTM Layer

LSTM is considered as one of most successful RNN variants, which introduces three additional gates. In the text mining domain, LSTMs have been involved in the task of sentiment analysis [19], sentence classification [20], etc. When using it for the whole long document, the training of LSTMs is not stable and underperformed traditional linear predictors as shown in [21]. Moreover, the training and testing time of LSTMs are also time/resource consuming for the long document. [21] illustrates that the training of LSTM can be stabilized by pre-training the LSTMs as sequence autoencoders or recurrent language models. However, the above problem is avoided when we use LSTMs for label sequence predictions, which is typically much shorter than a document. The label sequence here is the assignments of ordered labels to a text. Although several variants of LSTMs exists, the standard LSTM is used. An additional word embedding layer is also applied for the labels.

LSTM is an RNN that contains special units in the recurrent hidden layer called memory blocks [22]. There is an input, output, and forget gate for each memory cell. The hidden layer of LSTM is also called the LSTM cell. LSTM has the capability to plot long-term dependencies by defining each memory cell with a set of gates $<d$, where d is the memory dimension of the hidden state of LSTM [23].

Each word in the news headline F is independent of other words, when the words are represented by making use of word embedding E . In this layer, a new representation for each word is achieved by summarizing contextual information from both the directions in a news headline. The bidirectional LSTM is a combination of forward LSTM (2) and backward LSTM (3), which reads the comment from x_n to x_1 :

$$\overrightarrow{h}_t = \overrightarrow{LSTM}(w_t, \overrightarrow{h}_{t-1}) \quad (2)$$

$$\overleftarrow{h}_t = \overleftarrow{LSTM}(w_t, \overleftarrow{h}_{t-1}) \quad (3)$$

We concatenate forward hidden state and backward hidden state to obtain hidden state representation h_i for each word x_r . Then, h is calculated using (4) [24]:

$$h_i = \overrightarrow{h}_t \odot \overleftarrow{h}_t \quad (4)$$

Where, h_i is the output of the i -th word \odot function is a concatenation function used to combine the two outputs. Generally, different merge modes can be used to combine the outcomes of the Bi-LSTM layers. These are concatenation (default), multiplication, average, and sum. h_t represents the output sequence of the forward layer which is calculated iteratively using inputs in a positive sequence from time $t-n$ to time $t-1$, \overleftarrow{h}_t represents

the output sequence of the backward layer which is calculated using the reversed inputs from time $t-n$ to $t-1$. This process helps in capturing information of whole sentence around every word x_t . We denote all the hidden state of the words x_t as $H \in \mathbb{R}^{N \times 2p}$, where size of $-h$ forward and h backward be p .

$$H = (h_1, h_2, \dots, h_n) \quad (5)$$

4. Attention Layer

In the text analysis task, the attention model is used to represent the correlation between the words in the text sentence and the output result. The model is first applied to the task of machine translation. The feed-forward attention model [25] adopted in this paper is a direct simplification of the conventional attention model. The simplification method is to construct a single vector c from the whole sequence., constructed as follows:

$$e_t = a(h_t) \quad (6)$$

$$a_t = \frac{\exp(e_t)}{\sum_{k=1}^T \exp(e_k)} \quad (7)$$

$$c = \sum_{t=1}^T a_t h_t \quad (8)$$

Where a is a learning function, and now it is only determined by h_t . In the above formula, the attention mechanism can be considered as constructing a fixed length of the embedded layer c of the input sequence by calculating an adaptive weighted average of the sequence of states h . We obtain the final sentence-pair representation used for classification from:

$$h^* = \tanh(c) \quad (9)$$

5. Max Pooling

The output from the proposed convolutional layer then undergoes 1-dimensional max pooling. This layer converts each kernel size of the input into a single output by selecting the maximum value observed in each kernel. Pooling is used to reduce overfitting; this allows to add more layers to proposed architecture and in turn allows the neural network to extract higher-level features.

6. ReLu Layer

The activation or ReLU Layer [26] is applied for dealing with the nonlinearity in the model. It generates a rectified feature map, which is fed to the concatenated layer to combine 2 matrix GloVe and FastText.

7. Concatenated Layer

In the Concatenated Layer, the output results from the two ReLU Layers with different word-embedding will be combined to form a larger matrix S_{all}

$$S_{all} = [f_g, f_s] \quad (9)$$

Where f_g is the vector result of the relu process with the word embedding GloVe and f_s is the vector result of the relu process with the word embedding fastText, which is fed to the fully connected layer.

8. Representation Layer

The output layer is a fully connected layer that consists of the softmax activation function. The concatenated feature map is input to the fully connected softmax layer, which calculates the probability of any output word and classifies the news headline as sarcastic or non-sarcastic as an output. The output vector of the softmax layer P_i is as (10).

$$P_i = \text{Softmax}(W_c \cdot f + b_c) \quad (10)$$

Where P_i is the probability of whether the input is sarcastic or not, and W_c is the weight matrix and b_c is the offset value.

Results and discussion

For discussing the results, the empirical analysis has been broadly divided into two parts: (i) parameter setting for the proposed model and (ii) comparison with multiple baselines on the basis of classification accuracy

1. Parameter Setting

Optimal selection of parameters is imperative to achieve superlative performance results. We use the validation data to tune the hyper-parameters so as to obtain the best results. Table 1 lists the values used in this work.

Table 1 Hyperparameter values

Hyperparameter	Value
Dimension of GloVe and fastText vector	300
Hidden units of LSTMs (Forward, Backward)	64 each
Batch Size	128
Activation Function	ReLU
Regularization	Dropout Operation
Dropout Rate	0.5: word embedding; 0.2 BLSTM; 0.2: ReLu
Learning Rate	0.005
Epochs	100

2. Performance Result

The proposed model is evaluated to predict sarcasm in news headline using one datasets containing a total of 56,418 news headline. The results have been assessed using key performance indicators (accuracy, recall, precision, and F measure ([27][28]) Table 2 lists the results of the proposed MCAB-BLSTM model implemented on the datasets.

Table 2 Performance of the proposed MCAB-BLSTM

MCAB-BLSTM	Value
Accuracy	96.64%
Recall	97%
Precision	97%
F-Measure	97%

3. Comparison With Other Deep Learning Models Result

We compare the results of the proposed model with three other deep learning architectures, namely, CNN-LSTM, and Hybrid Neural Network. The word embedding was performed using GloVe for each baseline model and the evaluation was made for both datasets using the four key performance indicators. The results obtained for CNN-LSTM, and Hybrid Neural Network are shown in Tables 3 and 4, respectively

Table 3 Performance of the CNN-LSTM

CNN-LSTM	Value
Accuracy	86.16%
Recall	87%
Precision	85%
F-Measure	86%

Table 4 Performance of the proposed Hybrid Neural Network

Hybrid Neural Network	Value
Accuracy	89.7%
Recall	90%
Precision	88%
F-Measure	89%

It can be clearly observed that the proposed MCAB-BLSTM outperforms the other models with an accuracy of 96.64% achieved for the news headline dataset. CNN-LSTM shows the least accuracy of 86.16% for the news headline news datasets, respectively. The models, in order from lowest to highest accuracy, are CNN-LSTM < Hybrid Neural Network < MCAB-BLSTM. The best recall is also observed for the proposed MCAB-BLSTM for the news headline datasets with value of 97%. The best MCAB-BLSTM also model demonstrates the best

precision value of 97%. Table 5 summarizes the comparison of the accuracy results obtained by the above three models. Fig. 3 graphically illustrates these comparative results.

Table 5 Accuracy Comparison of models

Models	Value
CNN-LSTM	86.16%
Hybrid Neural Network	89.7%
MCAB-BLSTM	96.64%

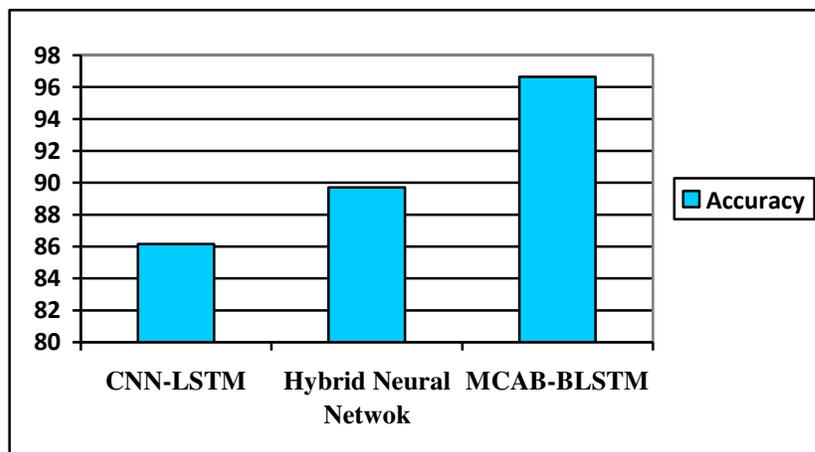


Figure 2. Accuracy Comparison of the Proposed Model with Baseline Models.

Conclusions

This paper proposes a sarcasm detection model Multi-Channel Attention Based BLSTM (MCAB-BLSTM). The model is based on the characteristics of text sentiment analysis. Firstly, the 2 words embedding is constructed and then input into the Attention Based BLSTM then process to ReLu Layer and we concatenate 2 output matrices for training, which effectively solves the long-term dependence and gradient dispersion of the other deep learning model in the training process. Experiments show that this method can significantly improve the sarcasm detection effect, and the accuracy rate reaches 96.64%. In the future, on the basis of this paper, exploring the influence of different forms of neural networks on the model, and trying to introduce more attention model structure and optimizing the sarcasm detection model will become the direction of further research.

Abbreviations

LSTM: Long Short-Term Memory; Bi-LSTM: Bidirectional Long Short-Term Memory; RELU: Rectified Linear Unit.

Acknowledgements

There are no acknowledgements.

Authors' contributions

AA designed and developed the system, interpreted the results and wrote the manuscript under the supervision of SH as an academic supervisor. SH also made contribution to the conception and analysis of the work. Both authors read and approved the final manuscript.

Funding

There is no funding.

Availability of data and materials Dataset.

Dataset from Kaagle

Competing interests

The authors declare that they have no competing interests

Reference

1. Peng, H., Li, J., He, Y., Liu, Y., Bao, M., Wang, L., & Yang, Q. (2018). Large-scale hierarchical text classification with recursively regularized deep graph-cnn. *Proceedings of the 2018 World Wide Web Conference*, (pp. 1063-1072).
2. Bharti, S. K., Vachha, B., Pradhan, R. K., Babu, K. S., & Jena, S. K. (2016). Sarcastic sentiment detection in tweets streamed in real time: a big data approach. *Digital Communications and Networks*, 2(3), 108-121.
3. Farzindar, A., & Inkpen, D. (2015). Natural language processing for social media. *Synthesis Lectures on Human Language Technologies*, 8(2), 1-166.
4. Bakshi, R. K., Kaur, N., Kaur, R., & Kaur, G. (2016). Opinion mining and sentiment analysis. In 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 452-455). IEEE.
5. Pozzi, F. A., Fersini, E., Messina, E., & Liu, B. (2016). Sentiment analysis in social networks. Morgan Kaufmann.
6. Kreuz, R., & Caucci, G. (2007). Lexical influences on the perception of sarcasm. *Proceedings of the Workshop on computational approaches to Figurative Language*, (pp. 1-4).
7. Gibbs Jr, R. W., Gibbs, R. W., & Colston, H. L. (Eds.). (2007). *Irony in language and thought: A cognitive science reader*. Psychology Press.
8. Wang, Z., & Song, B. (2019). Research on hot news classification algorithm based on deep learning. In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 2376-2380). IEEE.
9. Jain, D., Kumar, A., & Garg, G. (2020). Sarcasm detection in mash-up language using soft-attention based bi-directional LSTM and feature-rich CNN. *Applied Soft Computing*, 106198.
10. 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, Cham.
11. Mehndiratta, P., & Soni, D. (2019). Identification of sarcasm using word embeddings and hyperparameters tuning. *Journal of Discrete Mathematical Sciences and Cryptography*, 22(4), 465-489.
12. 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.
13. Hiai, S., & Shimada, K. (2019). Sarcasm Detection Using RNN with Relation Vector. *International Journal of Data Warehousing and Mining (IJDWM)*, 15(4), 66-78.
14. Xiong, T., Zhang, P., Zhu, H., & Yang, Y. (2019). Sarcasm Detection with Self-matching Networks and Low-rank Bilinear Pooling. In *The World Wide Web Conference*, (pp. 2115-2124).
15. Misra, R., & Arora, P. (2019). Sarcasm detection using hybrid neural network. *arXiv preprint arXiv:1908.07414*.
16. 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.
17. Guyon, I., & Elisseeff, A. (2006). An introduction to feature extraction. In *Feature extraction* (pp. 1-25). Springer, Berlin, Heidelberg.
18. Onan, A. (2019). Topic-enriched word embeddings for sarcasm identification. In *Computer Science On-line Conference* (pp. 293-304). Springer, Cham.

19. Tang, D., Qin, B., & Liu, T. (2015, September). Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 conference on empirical methods in natural language processing (pp. 1422-1432).
20. Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075.
21. Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. In Advances in neural information processing systems (pp. 3079-3087).
22. Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078.
23. Joshi, A., Bhattacharyya, P., Carman, M., Saraswati, J., & Shukla, R. (2016, August). How do cultural differences impact the quality of sarcasm annotation?: A case study of indian annotators and american text. In Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (pp. 95-99).
24. Graves, A., Jaitly, N., & Mohamed, A. R. (2013, December). Hybrid speech recognition with deep bidirectional LSTM. In 2013 IEEE workshop on automatic speech recognition and understanding (pp. 273-278). IEEE.
25. Raffel, C., & Ellis, D. P. (2015). Feed-forward networks with attention can solve some long-term memory problems. arXiv preprint arXiv:1512.08756.
26. Li, Y., & Yuan, Y. (2017). Convergence analysis of two-layer neural networks with relu activation. In Advances in neural information processing systems (pp. 597-607).
27. Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.
28. Bhatia, M. P. S., & Khalid, A. K. (2008). A PRIMER ON THE WEB INFORMATION RETRIEVAL PARADIGM. *Journal of Theoretical & Applied Information Technology*, 4(7).

Figures

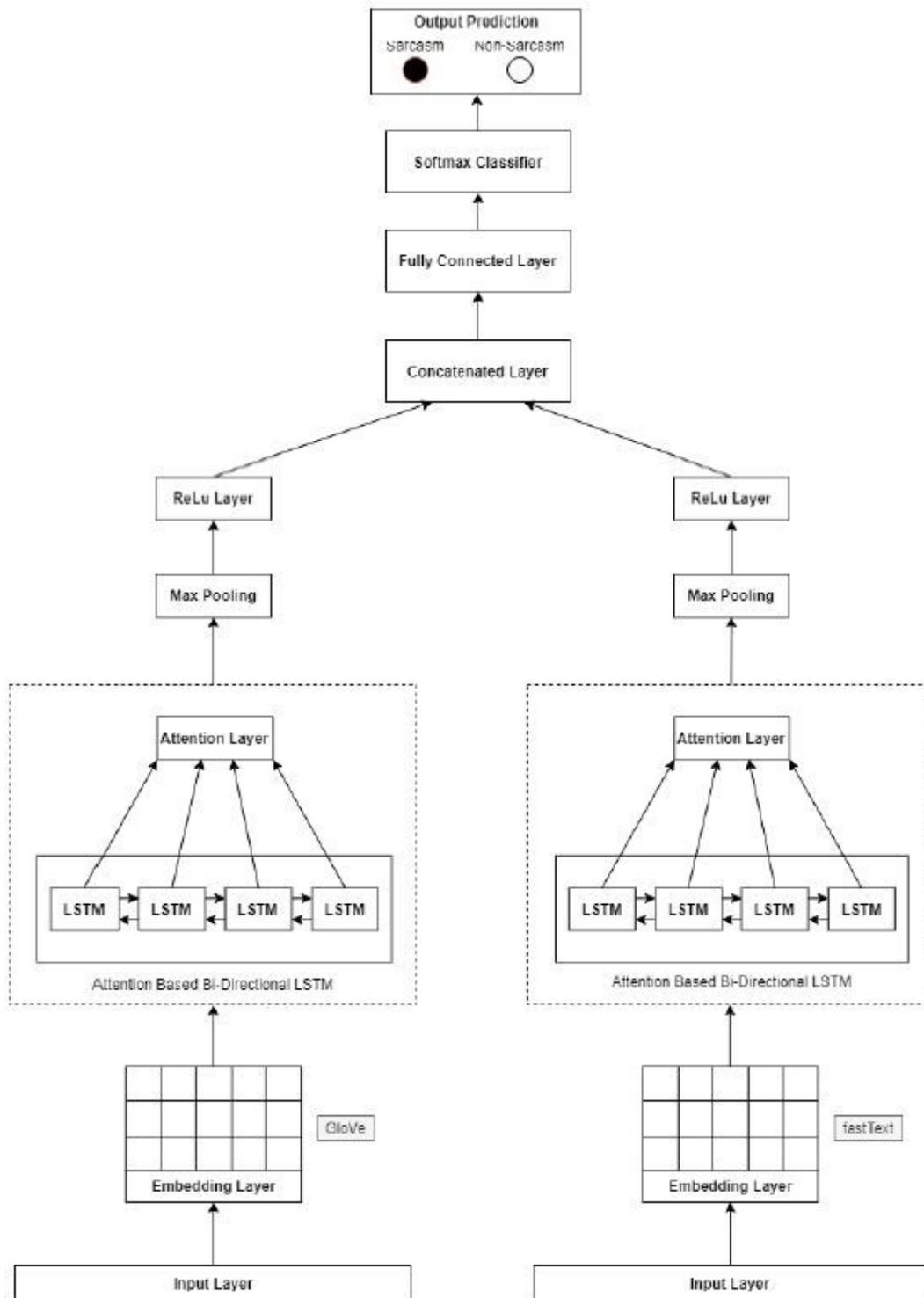


Figure 1

System architecture of the proposed Multi-Channel Attention Based BLSTM.

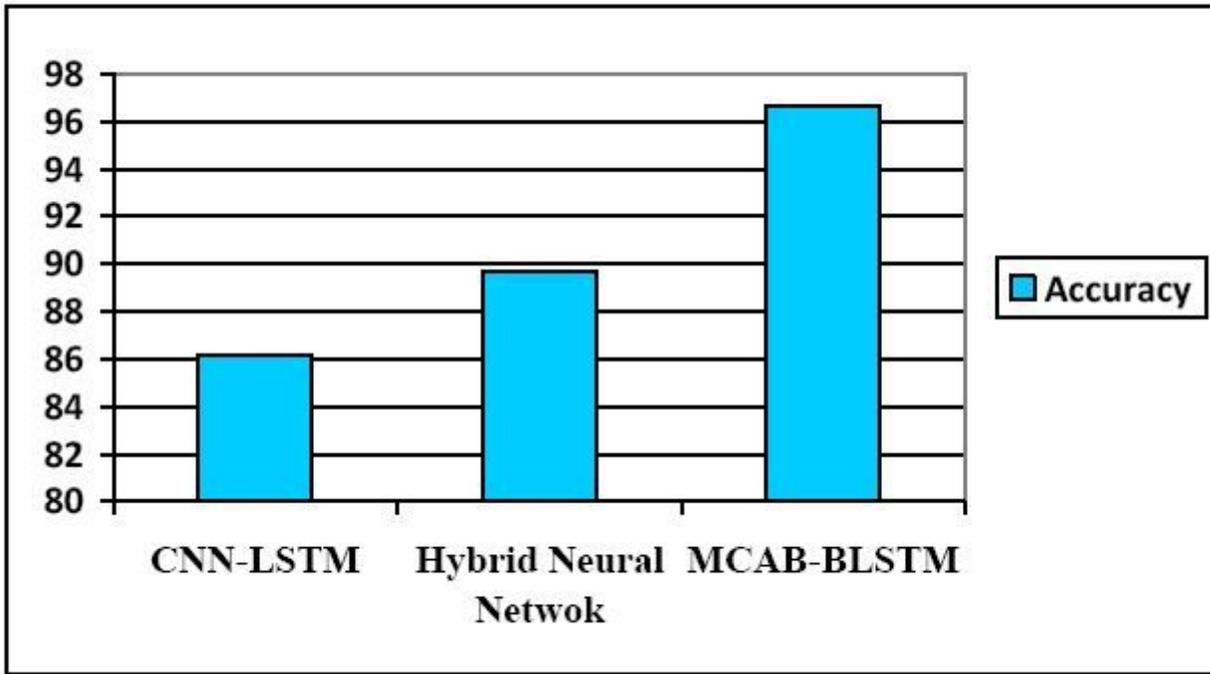


Figure 2

Accuracy Comparison of the Proposed Model with Baseline Models.