Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/68/20241456

Enhancing the interpretability of sentiment analysis with signatures

Hanchu Liu
School of Mathematics, Shandong University, Jinan, China

202100210051@mail.sdu.edu.cn

Abstract. Sentiment analysis is a significant research area within the field of natural language processing, with applications in various social contexts. Recent advances in related research have led to notable improvements in accuracy, yet existing text sentiment analysis models continue to face challenges in terms of interpretability. This paper proposes a text sentiment classification method integrating BERT and BiLstm, comprising three stages: training, signature generation, and testing. Our primary contribution lies in the signature generation and testing stages. In the training stage, BERT is employed to generate word embeddings, while BiLstm is utilized for learning to construct the text sentiment classification model. In the signature generation stage, the training and validation sets are passed through the aforementioned classification model, the feature vectors are extracted following their passage through the fully-connected layer, and the feature vectors are utilized to generate the signature generation stage. The signature is then stored in the database. In the Testing Stage, the aforementioned operation is repeated on the test set in order to obtain the signatures. The signatures are then matched in the constructed signature database in order to obtain the final classification results. The matching degree is subsequently used as a threshold in order to be employed for subsequent evaluation of the results. The method enhances the interpretability of the classification task by matching through signatures. Two datasets, SST-2 and IMDB, were subjected to experimentation. Four quantitative metrics were employed for the evaluation of the results: accuracy, recall, precision, and F1-score. With a threshold of 0.99, the model's optimal accuracy, recall, precision, and F1-score results reached 0.896, 1, 0.896, 0.945 and 0.847, 0.999, 0.848, 0.917, respectively.

Keywords: Sentiment Analysis, BERT, BiLSTM

1. Introduction

Sentiment analysis is a significant research avenue in natural language processing(NLP), with the main purpose to automatically identify the sentiment tendencies in texts. The swift growth of the Internet have facilitated the uninhibited expression of opinions, resulting in a significant increase in the number of online comments. The extraction of useful sentiment information from these vast quantities of textual data has significant practical value. In the commercial sector, sentiment analysis can be employed to comprehend consumers' attitudes and feedback,facilitating prompt adjustments to products. In the media industry, sentiment analysis facilitates the monitoring of the emotional tendencies associated with public topics,beneficial for media reports in identifying key issues that are of interest to the public.Sentiment analysis can assist government departments in understanding social opinions, providing crucial reference information for decision-making.It is evident that sentiment analysis has both theoretical and practical

DOI: 10.54254/2755-2721/68/20241456

value in various fields.Therefore, the research and improvement of sentiment analysis methods are of great significance.

At present, sentiment analysis technology is undergoing constant innovation and improvement, and has made significant progress. However, there are still a series of important problems requiring further investigation. The first problem is the ambiguity of the text. The polysemy of the vocabulary, the complexity of the structure and expressions such as irony can impede the ability to accurately assess the emotional tone. The second is the challenge of fine-grained sentiment analysis, such as aspect-level sentiment analysis, the classification efficacy of the current models remains suboptimal. The third is that of model interpretability. The existing sentiment analysis methodologies employ deep learning approach to extract features and predict. However, the series of operations in the decision-making process remain opaque, rendering it challenging to elucidate the underlying principles of the method in concrete terms. Therefore, the interpretability of these methodologies is often poor.

Aiming to address the issue of poor interpretability of the model, we have proposed a novel sentiment analysis method that offers a certain degree of interpretability. Our main contributions are as follows:

- A method is proposed for classifying and evaluating text through signatures. It uses trained models to generate feature vectors from training and validation sets and then uses the feature vectors as signatures, saved in the database along with raw text and classification labels, which makes the method innovative.
- The interpretability of the sentiment analysis method has been enhanced. The test set will also generate the corresponding signatures and identify the closest match in the database based on similarity. The signature with the highest similarity in the database will be selected as the matched signature, which will then be used to determinate of the sentiment of the tested data. This process enables the method to achieve a certain degree of interpretability.
- The code achieved satisfactory results. Two datasets, SST-2 and IMDB, were subjected to experimentation and four quantitative metrics, accuracy, recall, precision, and F1-score, were employed for the evaluation of the classification results.With a threshold of 99.9%,the best result of model achieved the accuracy, recall, precision, and F1-score results of 0.896, 1, 0.896, 0.945, and 0.847, 0.999, 0.848, 0.917, respectively. The data and code used will be open-sourced on Github and can be accessed via the following link: `https://github.com/YYL991/BERT_BiLSTM`.

The remainder is divided into four sections. The first introduces the research background. The second section introduces the newly proposed sentiment analysis method. The third section presents the results of experiments. The final section summarises the proposed method and proposes a direction for future research.

2. Related Work

There are many research efforts in sentiment analysis, which are divided into three main categories: Lexicon Based Approach,Traditional Machine Learning Approach and Deep Learning Approach.

2.1. Lexicon Based Approach

Lexicon Based Approach is pre-constructing Lexicon with pre-set scores to indicate its polarity for each token.After the text is tokenized, the sentiment tendency is calculated using specific rules. The advantage is that it's easy to implement and use. However, it also has some disadvantages. Firstly, the quality of the analysis depends to a large extent on the quality of the sentiment lexicon, which may not cover all the necessary aspects. Secondly, the method is domain dependent and may not perform well when the same word exhibits different sentiment in different domains [1].

2.2. Traditional Machine Learning Approach

Traditional Machine Learning Approach is improved over Lexicon Based Approach by inferring decision rules for sentiment analysis based on manually annotated data and labels, extracting features from

DOI: 10.54254/2755-2721/68/20241456

the text, and transferring them to a relevant model.Pang first combined Traditional Machine Learning Approach with Sentiment Analysis and applied it to the analysis of film review data. Gamon et al. applied Support Vector Machines to 40,884 customer feedbacks,using various combinations of feature sets, and finally achieved an accuracy of 85.47% [2]; Kang et al. used an improved version of the Naïve Bayes classifier on a restaurant review dataset, which improved the negative classification accuracy [3] ; YanYan et al. used decision trees based on a graph-based propagation approach to integrate features inside and outside sentences, outperforming other representational previous approaches [4].

Compared to Lexicon Based Approach,Traditional Machine Learning Approach has shown a significant improvement in performance and efficiency, but the ability to understand context and generalisation is still lacking.

2.3. Deep Learning Approach

In recent years, more and more scholars have introduced deep learning methods into sentiment analysis.Liu et al. applied the traditional RNN , which has the significant advantage of dealing with inter-dependency relationships among elements in a sequence and capture contextual information, but it risks at suffering from gradient explosion, and it's not able to remember the long-term relationships in the sequence [5]; Abid et al. by applying Bi-LSTM to solve the contextual long-term dependencies [6]; On the basis of making full use of the contextual information, the attention mechanism is also introduced to the sentiment analysis task,to focus on the salient parts of the text; Basiri et al. proposed an attention-based bidirectional CNN-RNN deep model for sentiment [7].Nowadays, more research has been done by combining word embedding models and neural networks for sentiment analysis. Word embedding,mapping words to real number vectors that reflect the similarity between lexical relationships, helps to improve the accuracy of the model.Stein et al. trained the models using the CNN algorithm and static word embeddings generation methods-GloVe, word2vec, and fastText , which showed a significantly higher then previous models [8].In 2018, Google AI Language Researchers open-sourced a new model for NLP called BERT,which is an extension of the Transformers.Due to BERT's stronger textual representation capability, it has now become a pre-trained language model for most natural language processing tasks.

With continuous new breakthroughs in deep learning, sentiment analysis have also developed rapidly, but the main improvement goal of these methods is to improve the accuracy and other indicators.The various sentiment analysis methods still have the problem of poor interpretability, therefore, we propose a new sentiment analysis method with some interpretability.

3. Methodology

Our approach is illustrated in Fig 1 and is implemented through three stages: training stage, signature generation stage and testing stage.

DOI: 10.54254/2755-2721/68/20241456

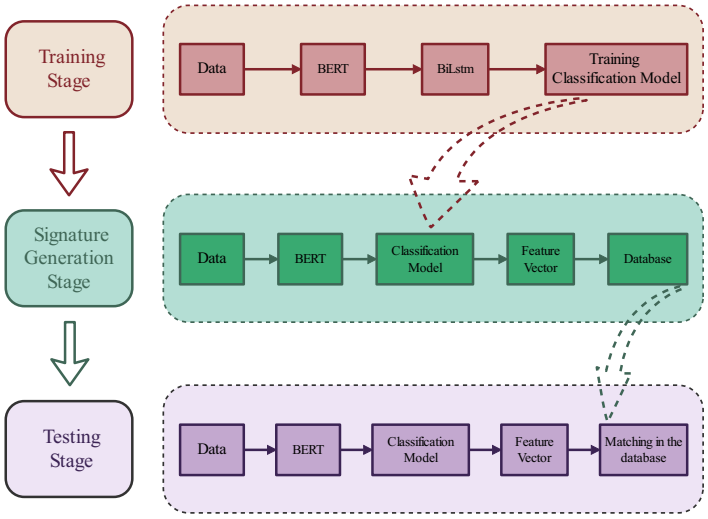


Figure 1: Our Method

In the training stage, BERT is first used to generate word embeddings. Subsequently, these word vectors are fed into a Bidirectional Long Short-Term Memory network (BiLSTM) to train the classification model. In the training stage, the data from the training and validation sets are used to generate word embeddings using BERT, fed into the classification model trained in the previous stage, extracted the feature vectors as signatures after the full connected layer, and construct database.In the testing stage, the same feature vector extraction operation as previous stage is performed on the test set to obtain the signatures ,which are looked up in the constructed database. The similarity between the signatures of test set and the ones in the database is computed, and matching is performed based on the similarity, to determine the sentiment of the test data.

The specific implementation of each stage will be developed below.

3.1. Training Phase

The specific model in the training stage is shown in Fig **??**.Firstly, the BERT pre-training model is utilized to obtain word embeddings containing contextual semantic information.Then BiLSTM network is used to extract contextually relevant features.Finally the results are obtained by the processing of the fully-connected layer and the nonlinear activation function Softmax function.

3.1.1. BERT Layer Unlike static pre-trained word embedding models such as Word2Vec or Glove, BERT generates different word vectors for the same word according to different contextual environments, extracting more feature information, so we chose to use BERT to generate word embeddings.

The input of the BERT model is comprised of three types of embeddings: token embeddings, segment embeddings, and position embeddings. Token embeddings is the basic component of the model's input, mapping each token into vector space to generate the vector representation of the word,to obtain the semantic information of each token. Segment Embeddings add special tokens CLS and SEP to the beginning and the end of each sentence, which are used to assist the model in distinguishing different input sequences. Position Embeddings are determined according to the position of words in the sentence, thereby enabling BERT to learn the order and positional information of words. Consequently, in the output of BERT, the same word in different sentences and different positions is represented by a distinct vector, enabling the Word Embeddings to deeply mine text feature information and to generate a comprehensive representation that encompasses word meanings, inter-sentence relationships and word order information.

The most critical component is the bi-directional Transformer coding layer, which employs an encoder for feature extraction. The encoder consists of multiple repetitive units, and each of which

DOI: 10.54254/2755-2721/68/20241456

comprises two main parts: Multi-Head Attention and Feed Forward Neural Network, with residual connection and normalisation designed between each two parts. Among them, Multi-Head Attention is the core component. It splits the input sequence into multiple "heads", each of which focuses on a different part of the sequence. Then the attention weights are culculated independently for each head and combined. Multi-Head Attention is capable of identifying the interrelationships between words within a sentence, therefore enabling it to gain a comprehensive understanding of the sequence. The sequences processed by the attention mechanism are passed through a Feed Forward Neural Network, which is essentially made up of two simple linear mappings and an activation function. Following each section, the values before and after the computation are summed to obtain the residual connection, and the hidden layer of the neural network is normalised in order to achieve a standard normal distribution, which serves to accelerate the training process and facilitate convergence. Finally, the layer outputs a bidirectional representation vector of the word, which is then used as an input to the subsequent network model.

3.1.2. BiLSTM layer LSTM can solve the problem of long distance dependence of RNN, and BiLSTM on this basis, can fully extract the following information, better mining contextual features, so we use to BilSTM to learn the text.

Each LSTM cell introduces cell states, forget gate,input gate and output gate to effectively deal with long sequence dependencies.

The cell state(C_t) is the memory part of the LSTM, which carries information across time steps and is updated at each moment based on the previous cell state, the input gate and the forget gate.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

The forget gate is employed to regulate which information will be retained and which will be forgotten. A portion of the the previous moment output is discarded, and the inputs of this moment and the outputs of the previous moment are integrated through a sigmoid function to obtain the output vector for the current time step f_t.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

The input gate is employed to regulate the input information of current moment . The tanh function is employed to extract the valid information, after which the sigmoid function is used to filter it. This process ultimately yields the vector i_t.

$$\begin{cases} i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_f) \end{cases}$$

The output gate is used to control the final output. The input information from the present moment and the cell state are combined in order to determine the final output.

$$\begin{cases} o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t = o_t * \tanh(C_t) \end{cases}$$

BiLSTM consists of a forward LSTM and a reverse LSTM superimposed on each other, and the output h_t of each moment is jointly determined by the output $\overrightarrow{h}_t$ of the forward LSTM, the weight matrix of the forward output w_t ,the output $\overleftarrow{h}_t$ of the reverse lstm,the weight matrix of the forward output v_t and the bias b_t at the moment t,which is calculated as follows:

$$\begin{cases} \overrightarrow{h}_t = LSTM(x_t, \overrightarrow{h}_{t-1}) \\ \overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t-1}) \\ h_t = w \cdot \overleftarrow{h}_t + v \cdot \overleftarrow{h}_t + b_t \end{cases}$$

DOI: 10.54254/2755-2721/68/20241456

3.2. Signature Generation Stage

Then,generate word embeddings from the texts of the training and validation sets by BERT, input them into the model, take out the vectors as signatures after passing the full connectivity layer but before passing the Softmax . And combined the original information of each text, the corresponding classification label , and the signature as a new data entry, and store all the entries into the constructed database so as to find the matches quickly in the next stage.

3.3. Testing Stage

The data of test set is also passed through the model to get the signature, calculate the similarity between each signature of the test set and each signature in the database, take the one with the largest similarity as the matching one, and take the classification labels of the matching signatures as the label results of the test text. In this process, the labels of the test text are obtained by matching the signatures with the maximum similarity, which works by letting the test text choose the same label as the sentence with the most similar semantic features, which improves the interpretability of the method.

We choose cosine similarity to calculate the similarity, and its calculation formula is as follows:

$$cosine\_similarity(A, B) = \frac{(AB)}{(||A|| * ||B||)}$$

4. Experiment

4.1. Experimental environment

The CPU used for the experiment is Intel(R) Xeon(R) CPU @ 2.00GHz, GPU is NVIDIA Tesla T4, RAM is 12GB, the programming language used is Python 3.10.The deep learning framework is Pytorch.

4.2. Experimental datasets

The datasets used in this paper are based on two binary classification datasets for sentiment analysis on movie reviews released by Stanford University, the SST-2 dataset (The Stanford Sentiment Treebank) [9] and the IMDB dataset [10].In order to facilitate the evaluation of the model results, we select the validation set with labels (67,350 entries) of SST-2 and the IMDB (25000 entries), and divide the new training set, the validation set and the test set in the ratio of 6:2:2.

4.3. Experimental evaluation indicators

The evaluation metrics used in this paper to evaluate the experimental results are F1 value, Accuracy, Precision and Recall.

Since the test set labels are obtained based on the matching signature with highest similarity , the concept of similarity threshold is introduced to evaluate the results. When its similarity is greater than the threshold value and the label is same as the actual label of the text, it is called True Positive (TP); when its similarity is greater than the threshold value, but the label is different, it is called False Positive (FP);when the similarity of this data is less than the threshold value, it is called False Negative (FN). The specific evaluation index calculation formula is as follows:

$$\begin{cases} \text{Accuracy} = \dfrac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \\ \text{Precision} = \dfrac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{Recall} = \dfrac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{F1} = \dfrac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \end{cases}$$

DOI: 10.54254/2755-2721/68/20241456

4.4. Parameter setting and ablation experiments

In the BERT layer, the pre-trained English model "BERT-BASE,uncased" released by Google is used, which uses a 12-layer Transformer, with a hidden layer size of 768, a Multiple-headAttention parameter of 12, and a total model parameter of 110MB. Negative Log Likelihood Loss Function(NLLLoss) is used as the loss function, AdamW is used as the optimiser, and Dropout regularisation is used in LSTM to prevent overfitting.

We carried out ablation experiments with 12 parameter combinations, in which the parameters of No.1 were set as follows: batch size is 256, the number of lstm layers is 2,the number of features in the hidden state is 768, lstm dropout is 0.5,the number of fully connected layer is 1,and the learning rate is 0.001.And the rest of the 11 sets of parameters were modified on the basis of No.1,listed in the table 1:

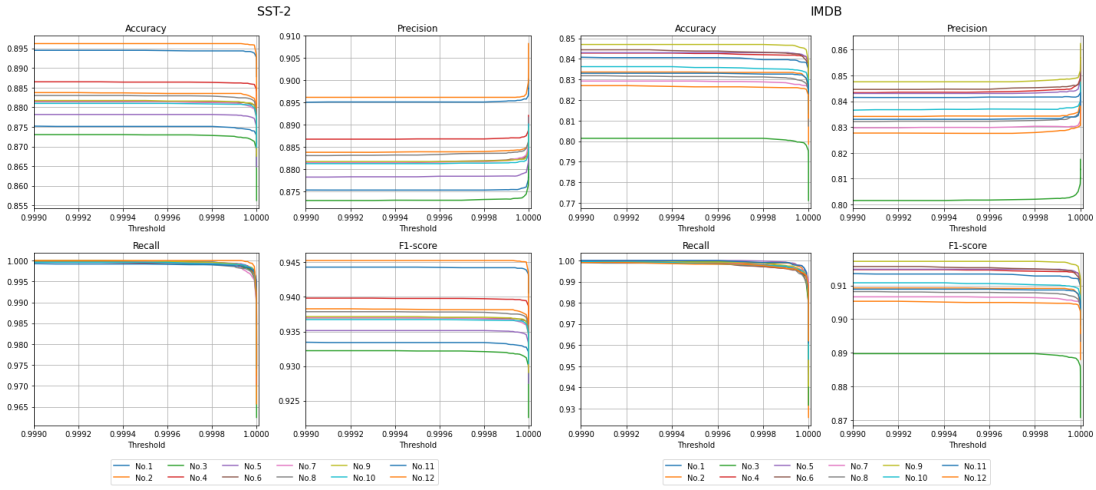
Table 1: Parameter setting

Parameter combination number	Changes relative to No. 1
No.1	——
No.2	learning rate = 0.0005
No.3	learning rate = 0.0001
No.4	batch size = 128
No.5	batch size = 64
No.6	number of lstm layers = 3
No.7	number of lstm layers = 4
No.8	dropout = 0.3
No.9	number of features in the hidden state = 512
No.10	number of features in the hidden state = 1024
No.11	number of fully connected layer = 2, number of features in the fully connected layer = 512
No.12	number of fully connected layer = 2, number of features in the fully connected layer = 256

4.5. Analysis of experimental results

Under different thresholds, the evaluation of classification results on the two datasets of SST-2 and IMDB are shown in Fig 3a and Fig 3b, respectively, and it can be seen that all the parameter combinations on the two datasets satisfy the law that Precision gradually rises as the thresholds increase, and Accuracy, Recall, and F1 all decrease.The change speed of each index is getting faster as the thresholds increase. From the comprehensive view, the parameter set of best classification effect on STT-2 dataset is No.12, and on IMDB dataset is No.9.Both two the worst is No.3, it may be that the learning rate is too small, so it falls into the local extreme point and does not really find the optimal solution.All of the rest of the parameter selection is appropriate and have achieved good results.

DOI: 10.54254/2755-2721/68/20241456



(a) Classification Results of SST-2 (b) Classification Results of IMDB

Figure 3

4.6. Two Cases of Signature Matching Mechanism

In order to better understand the signature matching process, we selects a piece of data from the SST-2 dataset and a piece from the IMDB to show the matching process and results.

Under the parameter combination No.12, which has the best classification effect with the SST-2 dataset, a piece of data in the test set is extracted, "the moral shrapnel and mental shellshock will linger long after this film has ended." The label is positive and the feature vector is [-0.00652470812201499,0.108263097703456].

Traversing the signatures one by one in the database, the matched text ,with the feature vector [-0.00777799263596534,0.122177526354789], is that "you may feel compelled to watch the film twice or pick up a book on the subject ." And the label is also positive ,the similarity of two is 0.999994284.

Under the parameter combination No.9 , a piece of data from the test set of IMDB dataset is extracted:

"Contains spoilers The movie plot can be summarized in a few sentences: Three guys go hunting in the forest. Two of them along other people get shot in the head without explanation. The last guy can stand in the clear, shout and do anything without getting shot ..... To pay money to rent this as a DVD is totally inappropriate. The one thing that is a little funny is the extra with the director telling, how they local police didn't realize that they were shooting and treated them like a random guy walking around with a gun. If they'd have filmed that, I'd be sure it would be more fun to watch then the movie." The label is negative and the feature vector is [0.0330197028815746,0.0274281315505504].

Traversing the signatures in the database, the matched text is:

"Have you ever had a cool image in your mind that you thought it would be nice to be in a movie: Like seeing a detective peeking through the cracks of a broken fence of some abandoned house? Or seeing a woman walking down a street looking cold and intense and awfully alert? Yeah. Imagine stretching that image to a whole movie ..... In the end, I felt like: OK, I know what you are trying to say here but is that the point you are trying to make by spending two hours building up all these tension? It is rather irrelevant with who the characters are and what kind of life they have.And we are given very little about who the characters are.All we have is this circumstance that just took placed. Disappointing but I guess the director did not have much material to work with and it shows."And the label is also negative ,the feature vector is [0.0317571572959423,0.02719122543931], and its similarity is 0.999888433.

It can be observed that through signature matching, the text to be tested is able to match the most semantically similar data in the database. Therefore, it is reasonable to use the matched signature to inform the determination of the labels of the text to be tested. In other words, the test text uses the same labels of the matched signature. This process enhances the transparency of the decision-making process

DOI: 10.54254/2755-2721/68/20241456

of the sentiment analysis model and facilitates the interpretability of the sentiment analysis method.

5. Conclusion and Future Work

In this paper, a method based on pre-trained model BERT,LSTM neural network and signature of text is proposed, experiments are conducted on two datasets, SST-2 and IMDB, and the model is evaluated using four metrics under different thresholds, which shows an improvement in interpretability compared to other existing methods. However, there are still some problems with the model, which is that there is a certain decrease in model accuracy while improving the interpretability. Therefore, the focus of future work is to ensure that the accuracy of the model remains unchanged or improves while improving the interpretability.

References

[1] A. Moreo, M. Romero, J. L. Castro, and J. M. Zurita. Lexicon-based Comments-oriented News Sentiment Analyzer system. *Expert Systems with Applications*, 39(10):9166–9180, August 2012.

[2] Michael Gamon. Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 841–847, Geneva, Switzerland, August 2004. COLING.

[3] Hanhoon Kang, Seong Joon Yoo, and Dongil Han. Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications*, 39(5):6000–6010, April 2012.

[4] Yan-Yan Zhao, Bing Qin, and Ting Liu. Integrating Intra- and Inter-document Evidences for Improving Sentence Sentiment Classification. *Acta Automatica Sinica*, 36(10):1417–1425, October 2010.

[5] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent Neural Network for Text Classification with Multi-Task Learning.

[6] Fazeel Abid, Muhammad Alam, Muhammad Yasir, and Chen Li. Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter. *Future Generation Computer Systems*, 95:292–308, June 2019.

[7] Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, Erik Cambria, and U. Rajendra Acharya. ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis. *Future Generation Computer Systems*, 115:279–294, February 2021.

[8] Roger Alan Stein, Patricia A. Jaques, and João Francisco Valiati. An analysis of hierarchical text classification using word embeddings. *Information Sciences*, 471:216–232, January 2019.

[9] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[10] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.