Sentiment Analysis Using an CNN-BiLSTM Deep Model Based on Attention Classification

WANG, Yue

(出版者 / Publisher)
法政大学大学院理工学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 理工学研究科編
(巻 / Volume)
63
(開始ページ / Start Page)
1
(終了ページ / End Page)
7
(発行年 / Year)
2022-03-24
(URL)
https://doi.org/10.15002/00025357

Sentiment Analysis Using an CNN-BiLSTM Deep Model Based on Attention Classification

Name: *WANG Yue*
Major: Applied Informatics, Graduate School of Science and Engineering,
Hosei University,
Supervisor: *Lei Li*

***Abstract*— Sentiment analysis has been a hot research topic in NLP and data mining fields in the last decade. To solve the feature dimension is too high and the pool layer information is lost, which leads to the loss of the details of the emotion vocabulary. I proposed a Word2vec-CNN-BiLSTM hybrid model means the Word Vector Model, Bidirectional Long-term and Short-term Memory networks and convolutional neural network are combined in Quora dataset. The experiment shows that the accuracy achieved 91.48% performs better than each single model in short text. However, even with the hybrid approach that leverages the powers of these two deep-learning models, the number of features remains huge and hindering the training process. Secondly, I propose an attention-based CNN-BiLSTM hybrid model in IMDB movie reviews dataset. When the data size was 13 k, the proposed model had the highest accuracy at 0.908, and the F1 score also showed the highest performance at 0.883. When data was 20 k, the F1 score showed the best performance at 0.906, and accuracy was the highest at 0.929. The experimental results show that the BiLSTM-CNN model based on attention mechanism can effectively improve the performance of sentiment classification when processing long-text tasks.**

Keywords: Sentiment Analysis, CNN, BiLSTM, Attention Mechanism, Text Classification

I. INTRODUCTION

In recent years, the analysis of text affective tendency has attracted the attention of many scholars and has become a hot topic. With the rise of neural network, its related algorithms show a higher classification effect in text classification.

Sentiment analysis [1] is a set of linguistic operations belonging to the automatic processing of natural language. It's objective to identify the sentiment expressed in the text and to predict its polarity (positive or negative) towards a given subject.

At present, Deep Learning (DL) has amazing effects in the field of machine vision [2] and natural language processing. This paper also focuses on sentiment analysis of comment data based on deep learning framework. Mikolov et al. [3] proposed the Word2Vec: The first technological breakthrough to establish the neural network word embedding representation. KIM Y [4] used CNN for text feature extraction for text classification for the first time in 2014. Shen et al. [5] proposed a special design that combines the CNN and BiLSTM models for optimal performance. They found that this combination gave an accuracy of 89.7%.

Attention is arguably one of the most powerful recent concepts in deep learning, bringing out many breakthroughs in the field and improving the performance of neural networks [6]. It can effectively give a model the ability to "attend to" a certain part of the input sequence, which would arguably be a part with higher importance.[7] He et al. [8] used two transfer methods besides the attention-based LSTM for document-level sentiment analysis.

The goal of this paper is to improve the performance of the text sentiment analysis system, mainly studying the method of applying deep learning technology to text sentiment analysis and studying the improvement of the existing general deep learning model to be suitable for specific text sentiment analysis tasks.

The overview of the paper is as follows: Section 2 proposes a literature review of neural models for sentiment analysis and text classification; Section 3 presents the theoretical background of the proposed neural models; Section 4 describes the proposed Word2vec-CNN-BiLSTM Short Text Classification model experiments and results; Section 5 describes CNN-BiLSTM Based on Attention Mechanism Long text Classification model experiments and results ; Section 6 concludes the paper and offers some directions for future research.

II. METHODS

A. CNN

Convolutional Neural Networks (CNN) is a feed-forward neural network with convolution structure [9].

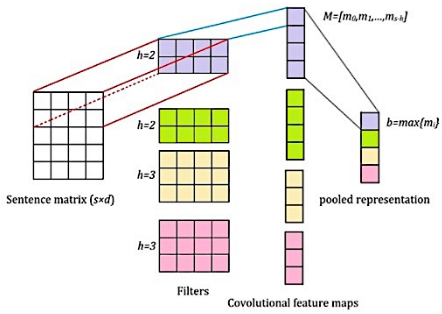

Fig. 1. CNN for feature extraction.

The convolution layer extracts local text features through multiple filters in Fig.1. Equation (1) defines a vector of sequential text obtained by concatenating embedding vectors of the component words:

$$X_{1:T} = [x_1, x_2, x_3, \cdots x_T] \ , \tag{1}$$

$$h_{d,t} = \tanh(W_d x_{t:t+d-1} + b_d) \ , \tag{2}$$

where $x_{t:t+d-1}$ are the embedding vectors of the words in the window, W_d is the learnable weights matrix, and b_d is the bias. Since each filter must be applied to various regions of the text, the feature map of the filter with convolution size d is:

$$h_d = [h_{d1}, h_{d2}, h_{d3}, \cdots x_{T-d+1}]. \tag{3}$$

For each convolution kernel, we apply max pooling to the feature maps with convolution size d to obtain the final feature map of the window, we concatenate pd for each filter size d = 1, 2, 3 and extract the unigram, bigram, and trigram hidden features

$$p_d = Max^t(h_{d1} h_{d2} h_{d3} \cdots x_{T-d+1}), \tag{4}$$

$$h_d = [p_1, p_2, p_3]. \tag{5}$$

To apply a CNN on a sentence S with s words, first, an embedding vector of size e is created. Then, a filter F of size $e \times h$ is repeatedly applied to the sub-matrices of the input feature matrix. This, produces a feature map $M = [m_0, m_1, \ldots, m_{s-h}]$ as follows:

$$m_i = F \times S_{i:i+h-1} \tag{5}$$

where, $i = 0,1,\ldots,s-h$ and $S_{i:j}$ is a sub-matrix of S from row i to j. Max-pooling is a common pooling strategy which select the most important feature b of the feature map as follows:

$$b = \max_{0 \le i \le s-h} \{m_i\} \tag{6}$$

B. BiLSTM

Bi-directional Long Short-Term Memory (BiLSTM) is a combination of directional Long short-term Memory

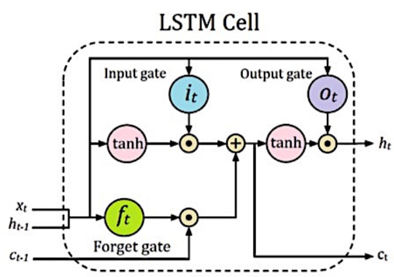

Fig. 2. RNN units LSTM.

for the forward LSTM and the backward LSTM (Fig.2).

BiLSTM is well suited for sequential labeling tasks that relate to the up and down. Therefore, it is often used to model context information in NLP [10,11].

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \ , \tag{7}$$

$$f_t = \sigma\big(W_f x_t + U_f h_{t-1} + b_f\big) \ , \tag{8}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \ , \tag{9}$$

$$\tilde{c}_t = Tanh(W_c x_t + U_c h_{t-1}) \ , \tag{10}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \ , \tag{11}$$

$$h_t = o_t \odot Tanh(c_t) \ . \tag{12}$$

i_t is obtained through linear transformation of input x_t and output h_{t-1} of the hidden layer in the previous step, and then through activation function σ. The result of the input gate i_t is a vector, where each element is a real number between 0 and 1 to control the amount of information that flows through the valve in each dimension. The two matrices W_i, U_i and vector b_i are the parameters of the input gate, which need to be learned in the training process.

The forget gate f_t and output gate o_t are calculated similarly to the input gate, with their own parameters W, U and b. Different from the traditional RNN, the transfer from the state c_{t-1} of the previous memory unit to the current state c_t does not necessarily depend entirely on the state calculated by the activation function, and is controlled by the input gate and the forget gate together.

C. Attention mechanism

The Attention mechanism is essentially a process of addressing in a bidirectional network. as shown in the Fig 3.

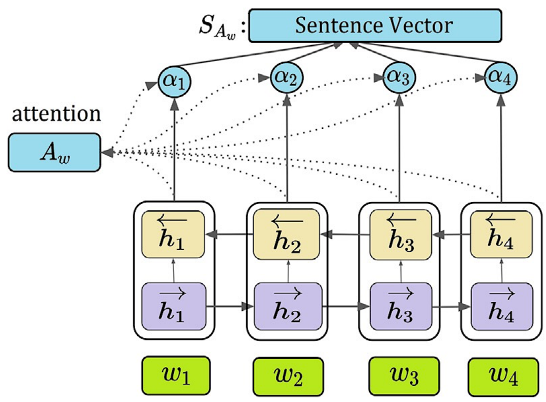

Fig. 3. The attention mechanism in a bidirectional network.

The attention layer can access all previous states and weigh them according to some learned measure of relevant to the current token, providing sharper information about far-away relevant tokens [12].

A common way of assigning different weights to different words in a sentence is to use a weighted combination of all hidden states, S_{AW} as follows:

$$\boldsymbol{\alpha}_t = \frac{\exp\left(\mathbf{v}^{\mathrm{T}} \cdot \tilde{\mathbf{h}}\right)}{\sum_t \exp\left(\mathbf{v} \cdot \tilde{\mathbf{h}}\right)} \tag{13}$$

$$\mathbf{S}_{\mathbf{Aw}} = \sum_t \boldsymbol{\alpha}_t \mathbf{h}_t \tag{14}$$

where h˜ and h are defined as shown in Eqs.(13) and (14) and v is a trainable parameter.

D. Algorithm

$$x_i = \tan u\left(W_p P_i + b_p\right) \quad (15)$$

$$\alpha_i = \operatorname{softmax}(W_\alpha\, x_i) \quad (16)$$

$$S_{cnn} = \sum_{i=1}^{n} \alpha_i P_i \quad (17)$$

$$y_i = \tan u\left(W_q Q_i + b_q\right) \quad (18)$$

$$\beta_i = \operatorname{softmax}\left(W_\beta y_i\right) \quad (19)$$

$$S_{bilstm} = \sum_{i=1}^{n} \beta_i Q_i \quad (20)$$

$$[\, S_{cnn}\,, S_{bi-lstm}\,] \quad (21)$$

In formula (15) to (20), $x_i\,/y_i$ used tanh activation function.The attention weight of each component and weight represents the importance of the word.

SoftMax function obtains the attention weight of each component x_i, W_{cnn}= (α_1,α_2, ... α_n), The weight represents the importance of the word.

Formula S_{cnn}weighted sum of the output vectors of CNN structure and obtained the semantic vector of sentence level extracted by CNN to represent S_{cnn}. Similarly, the output vector of BiLSTM is expressed as vector [q_1, q_2... q_n],

Finally, S_{bilstm} is weighted and summed to the vectors [q_1, q_2... q_n] output from the BiLSTM structure to obtain the semantic vector of sentence level extracted by LSTM to represent $S_{bi-lstm}$.

Hence, the semantic vector representation extracted from CNN and BiLSTM is spliced, that is Formula (21) $[\, S_{cnn}\,, S_{bi-lstm}\,]$ which is used as the input of the later matching layer, to extract more abundant features by combining the respective advantages.

To generate the input review matrix, a pre-trained embedding matrix $W_g \in \mathbb{R}^{n\times e}$ with n and e being the total number of words and embedding dimension, is used to embed a comment vector $c \in \mathbb{R}^m$ with m being the padding length or maximum number of words $w_t, t \in [1, m]$ considered in the reviews as follows.

$$c_t = W_g w_t, t \in [1, m]. \quad (22)$$

Then, a parallel layer of Bi-LSTM is applied on the output of embedding layer to process the sequences of arbitrary length and extract long dependencies in both forward and backward directions. I employed LSTM to make the proposed model capable of remembering both short and longer sequences.

$$\overrightarrow{h}_{tLSTM} = \overrightarrow{LSTM}(c_t), t \in [1, m]. \quad (23)$$

$$\overleftarrow{h}_{tLSTM} = \overleftarrow{LSTM}(c_t), t \in [m, 1]. \quad (24)$$

For each word, w_t can now obtain an annotation by concatenating forward and backward contexts as follows:

$$h_{tLSTM} = [\overrightarrow{h}_{tLSTM}, \overleftarrow{h}_{tLSTM}] \quad (25)$$

The attention mechanism is applied on h_{tLSTM} to make the model capable of paying more or less attention to different words in the reviews. To achieve this, I modified the feature vector by extracting informative words as follows:

$$u_{t_{LSTM}} = \tanh\left(\boldsymbol{W}_{w_{LSTM}} \boldsymbol{h}_{t_{LSTM}} + \boldsymbol{b}_{w_{LSTM}}\right) \quad (26)$$

$$\alpha_{t_{LSTM}} = \frac{\exp\,(\mathbf{u}_{t_{LSTM}}^T \mathbf{u}_{w_{LSTM}})}{\sum_t \exp\,(\mathbf{u}_{t_{LSTM}}^T \mathbf{u}_{w_{LSTM}})} \quad (27)$$

$$s_{LSTM} = \sum_t \alpha_{t_{LSTM}} h_{t_{LSTM}} \quad (28)$$

where u_t is a hidden representation of h_t and u_w is a context vector which is randomly initialized and jointly learned in the training phase. The importance of a word u_t is calculated using the similarity of u_t with u_w and is normalized as shown in Eqs. (27). These importance weights α_t are finally aggregated into s by applying a weighted sum on them. S is the comment vector and summarizes all the information of words in the comment.

III. RESULTS

A. Datasets

In the current study, the proposed models are evaluated on long review and short text datasets for sentiment analysis using the following datasets.

- Quora Question Pairs [13,14]: The data is from Quora website, which contains **2006** problem pairs, 1256 neutral, 482 positive and 273 negative ones. Training set is 75% and test set is 25%.
- IMDB reviews dataset [15,16]: **50,000** reviews from IMDB movie, allowing no more than 30 reviews per movie. The constructed dataset contains an even number of positive and negative reviews:
  A negative review score ≤ 4 (10),
  A positive review score ≥ 7 (10).
  Neutral reviews are not included in the dataset.

B. Parameter settings

1. Word2vec-CNN-BiLSTM model

I propose the following architecture composed of three parts, which are described in more detail below (Figure 3).

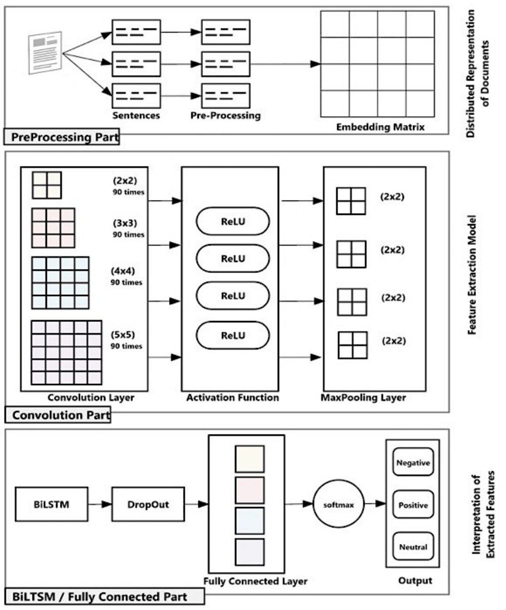

Fig. 3. CNN-BiLSTM general architecture

Table 1. Word2vec-CNN-BiLSTM configurations

Word Embedding	Epochs	Batch size	Optimizer	Accuracy
Word2vec	8	32	Adam	81.18%
			SGD	67.72%
		64	Adam	82.49%
			SGD	69.53%
	10	32	Adam	84.55%
			SGD	71.21%
		64	Adam	87.03%
			SGD	73.40%
	20	32	Adam	88.96%
			SGD	75.67%
		64	Adam	**91.48%**
			SGD	78.96%

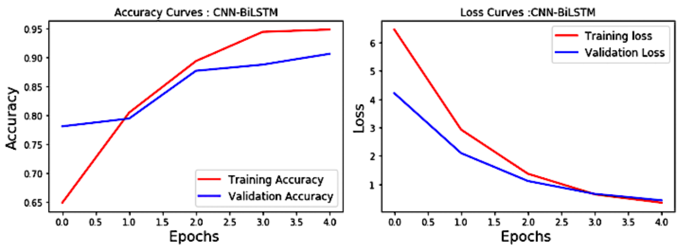

Fig. 4. CNN-BiLSTM accuracy and loss

The Word2vec-CNN-BiLSTM model achieved an accuracy of 91.48% (Table 1 and Fig. 4).

2.CNN-BiLSTM based on attention mechanism model

To address the limitations of the existing deep architectures for sentiment analysis, the paper proposes a new hybrid deep model, for polarity detection of long movie reviews.[17] In this model, word embedding, BiLSTM, attention mechanism, and CNN are used to better capture both long-term dependencies and local features.

This is the proposed architecture , The input is sequentially passed through the BiLSTM layer, the attention layer, the convolution and pooling layer. By considering the temporal information flow in two directions, the past and future contexts are extracted, and different words are emphasized. The convolution and pooling mechanisms are used to reduce the feature dimension and extract local features.

The training batch size for is set as 64 and the dropout rate is 0.2. The Adam stochastic optimizer with the learning and decay rate of 10^{-3} and 10^{-10} are used to train the network using the back-propagation algorithm. Binary cross-entropy [18] is used as the loss function and accuracy metric is calculated to detect the convergence.

Table 2. Attention-CNN-BiLSTM configurations

Word Embedding	Epochs	Batch size	Optimizer	Accuracy
Doc2vec	10	32	Adam	**93.06%**
		64	Adam	88.35%
	30	32	Adam	85.22%
		64	Adam	86.69%
	5	32	Adam	83.48%
		64	Adam	81.84%

Thus, from Table 2. the accuracy was computed according to three different iterations (i.e., 5, 10 and 30), between two values of the batch size (i.e., 64 and 128), and the optimizer Adam. The Doc2vec-CNN-BiLSTM model achieved an accuracy of 93.06%.

Evaluation Metrics [19]:

$$\text{Precision} = \frac{TP}{(TP + FP)}, \tag{29}$$

$$\text{Recall} = \frac{TP}{(TP + FN)}, \tag{30}$$

$$\text{F1 Score} = \frac{2 \times (Pr \times Re)}{(Pr + Re)} \tag{31}$$

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{32}$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively.

C. *Experimental analysis*

1.Word2vec-CNN-BiLSTM model

The proposed model was the highest at 91.48% in Table 3, followed by CNN-BiLSTM, BiLSTM, CNN and LSTM.

Table 3. Models' comparison

Word Embedding	Model	Accuracy
Word2vec	LSTM	79.83%
	CNN	81.25%
	BiLSTM	85.69%
	CNN-LSTM	87.44%
	CNN-BiLSTM	**91.48%**

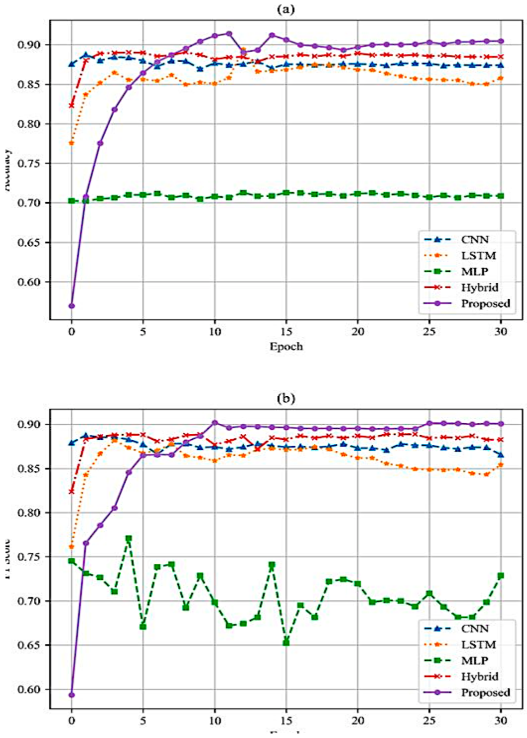

Fig 5. Model Performance as epochs increases: (a)Accuracy (2)F1 Score.

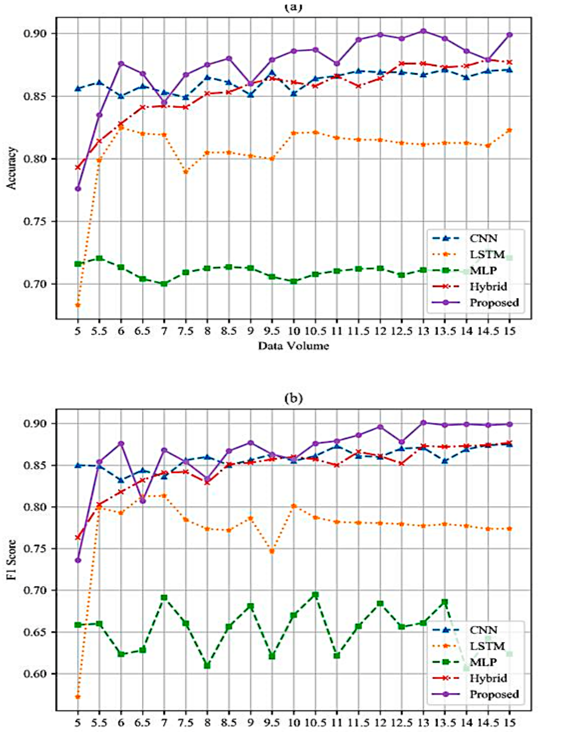

Fig 6. Model Performance as data volume increases: (a)Accuracy (2)F1 Score.

2.CNN-BiLSTM based on attention mechanism

Fig. 5 shows the performance of CNN, LSTM, MLP, hybrid combined with CNN-BiLSTM and the proposed model that change as the number of epochs increases when the data size is fixed at 20 k.

Fig. 6(a) shows the performance of the model changing as the number of epochs is fixed at 10 and the data size is increased from 5 k to 15 k. Fig. 6 (b) shows the accuracy and F1 score, respectively.

IV. DISCUSSION

A. Data volume and Epoch analysis

Table 4 shows the highest accuracy, F1 score, epoch value and training data for each model observed with the data size fixed at 20 k and increasing the number of epochs. First, the accuracy of the proposed model was the highest at 0.929, and the average accuracy was also the highest at 0.914. Next, the accuracy was high in the order of hybrid, LSTM, CNN and MLP[20].

The epoch value for obtaining the optimal accuracy of each model was 10 for CNN, 14 for LSTM, 5 for MLP and 15 epochs for CNN-LSTM, which took the longest learning time. The F1 score also showed the best performance of the proposed model at 0.906, followed by hybrid, CNN, LSTM. The proposed model for F1 score also showed the best performance at 0.906.

Table 4. Experimental results according to optimal epochs.

Model	CNN	LSTM	MLP	CNN-LSTM (hybrid)	Att-CNN-BiLSTM (proposed)
Accuracy(max)	0.861	0.882	0.702	0.885	**0.929**
Accuracy(avg)	0.859	0.849	0.698	0.873	**0.914**
Data volume	20k	20k	20k	20k	20k
Epoch	2	13	13	8	12
F1 score	0.863	0.876	0.768	0.871	**0.906**
Recall	0.885	0.937	0.849	0.889	0.903
Precision	0.877	0.813	0.701	0.870	0.897
Data volume	20k	20k	20k	20k	20k
Epoch	2	4	5	23	10

Table 5. Experimental results according to optimal data volume.

Model	CNN	LSTM	MLP	CNN-LSTM (hybrid)	Att-CNN-BiLSTM (proposed)
Accuracy(max)	0.855	0.808	0.703	0.863	**0.908**
Accuracy(avg)	0.849	0.766	0.695	0.847	0.812
Data volume	9.5k	6.5k	14k	12.5k	13k
Epoch	10	10	10	10	10
F1 score	0.857	0.801	0.689	0.861	**0.883**
Recall	0.871	0.764	0.885	0.879	0.887
Precision	0.851	0.812	0.681	0.866	0.883
Data volume	11k	7k	7.5k	12.5k	13k
Epoch	10	10	10	10	10

Next, Table 5 shows the results of the experiment conducted while fixing the number of epochs and increasing the data size from 5 k to 15 k. When the data size was 13 k, the proposed model had the highest accuracy at 0.908, followed by the hybrid model at 0.879, CNN model showed the same accuracy as 0.863, LSTM model 0.808, and MLP model 0.703. In terms of average accuracy, the proposed model was the highest at 0.812, followed by CNN, hybrid, LSTM, and MLP. The F1 score also showed the highest performance at 0.883 when the data size was 13 k.

B. Analysis between positive and negative

Another point in the results is that the improvements for the negative class are higher in comparison to those for the positive class (Fig.7). This may be the result of the fact that local relations such as negations and comparisons are more prevalent in the negative reviews in comparison to the positive reviews [21] and capturing such semantically negative relations is easier than positive relations.

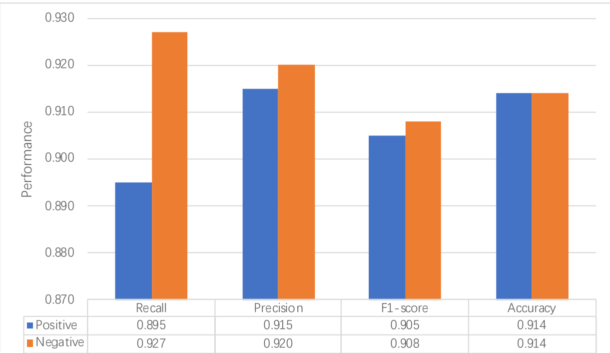

Fig. 7. Attention-CNN-BiLSTM proposed model results both Pos. and Neg. from IMDB dataset.

The model trained using the Internet Movie Database (IMDB) movie review data to evaluate the performance of the proposed model, and the test results showed that the proposed attention-Bi-LSTM-CNN model produces more accurate classification results, as well as higher recall and F1 scores, than individual multi-layer perceptron (MLP), CNN or LSTM models as well as the hybrid models.

V. CONCLUSION

Firstly, a combination of CNN and BiLSTM for short-text sentiment analysis with Word2vec Embedding was proposed. The combined CNN-BiLSTM model gives good results, since it benefits from the CNN's ability to extract features and the BiLSTM's characteristic to learn short-term bidirectional dependencies of the text.

However, even with the hybrid approach that leverages the powers of these two deep-learning models, the number of features to remember for classification remains huge and hindering the training process.

Hence, I proposed a BiLSTM-CNN attention hybrid model for text classification., which has demonstrated improved performance compared to the existing models, and the accuracy increases as the size of the data increases and the number of training increases. This approach addresses the data-loss and long-term dependency problems which affects the existing models especially when the data size becomes high.

One disadvantage of the current model is that it requires more training data and training time than the existing baselines. Even with this limitation, it can be effective in classification that requires a lot of training data.

In the future I will make the research of pre-training model, combining with the current advanced GPT and BERT, and working with the team to research the big model with larger parameters. And graph neural network (GNN) in the social network, knowledge graph, recommendation system and even life science and other fields.

ACKNOWLEDGEMENT

The research of Sentiment Analysis using Word2vec-CNN-BiLSTM Classification is supported by The Conference: 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS) in December 2020.

DOI:10.1109/SNAMS52053.2020.9336549

REFERENCES

[1] Ceraj, T.; Kliman, I.; Kutnjak, M. *Redefining Cancer Treatment: Comparison of Word2vec Embeddings Using Deep BiLSTM Classification Model;* Text Analysis and Retrieval 2019 Course Project Reports; Faculty of Electrical Engineering and Computing, University of Zagreb: Zagreb, Croatia, July 2019.

[2] Anvarjon, T.; Kwon, S. Deep-net: A lightweight CNN-based speech emotion recognition system using deep frequency features. Sensors 2020, 20, 5212.

[3] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems, Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.

[4] KIM Y. Mehta, N. Majumder, A. Gelbukh, E. Cambria, Recent trends in deep learning-based personality detection, Artif. Intell. Rev. 53 (2020) 2313–2339.

[5] Shen, Q.; Wang, Z.; Sun, Y. Sentiment analysis of movie reviews based on cnn-blstm. In International Conference on Intelligence Science; Springer: Berlin, Germany, 2017, pp. 164–171.

[6] Shuang, K.; Ren, X.; Yang, Q.; Li, R.; Loo, J. AELA-DLSTMs: Attention-Enabled and Location-Aware Double LSTMs for aspect-level sentiment classification. Neurocomputing 2019, 334, 25–34.

[7] Chen, H.; Sun, M.; Tu, C.; Lin, Y.; Liu, Z. Neural sentiment classification with user and product attention. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 21 September 2016; pp. 1650–1659.

[8] R. He, W.S. Lee, H.T. Ng, D. Dahlmeier, Effective attention model- ing for aspect-level sentiment classification, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1121–1131.

[9] Wang, X.; Jiang, W.; Luo, Z. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics, Osaka, Japan, 11-16 December 2016; pp. 2428–2437.

[10] Wei, J.; Liao, J.; Yang, Z.; Wang, S.; Zhao, Q. BiLSTM with multi-polarity orthogonal attention for implicit sentiment analysis. Neurocomputing 2020, 383, 165–173.

[11] Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing 2019, 337, 325–338.

[12] Giannopoulos, P.; Perikos, I.; Hatzilygeroudis, I. Deep learning approaches for facial emotion recognition: A case study on FER-2013. In Advances in Hybridization of Intelligent Methods; Springer: Cham, Switzerland, 2018; pp. 1–16

[13] SHANKAR I, NIKHILD. First Quora dataset release: question pairs [EB/OL]. [2019-03-01] https://data.Quora.com/First-Quora-Dataset-Release-Question-Pairs.

[14] Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Kudlur, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.

[15] Yenter, A.; Verma, A. Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis. In Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York, NY, USA, 19–21 October 2017; pp. 540–546.

[16] Jasmir, J.; Nurmaini, S.; Malik, R.F.; Abidin, D.Z. Text Classification of Cancer Clinical Trials Documents Using Deep Neural Network and Fine-Grained Document Clustering. In Proceedings of the Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019), Palembang, Indonesia, 16 November 2019; Atlantis Press: Paris, France, 2020; pp. 396–404.

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

[18] Paszkiel, S. Using neural networks for classification of the changes in the EEG signal based on facial expressions. In Analysis and Classification of EEG Signals for Brain–Computer Interfaces; Springer: Cham, Switzerland, 2020; pp. 41–69.

[19] Yadav, A.; Vishwakarma, D.K. Sentiment analysis using deep learning architectures: A review. *Artif. Intell. Rev.* 2020, 53, 4335–4385.

[20] Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint 2014, arXiv:1406.1078.

[21] Bengio, Y. Learning deep architectures for AI. Found. Trends Mach. Learn. 2009, 2, 1–127.

[22] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation. 2014, 2018, URL: https://nlp.stanford.edu/projects/glove/ [accessed 2018-01-11] [WebCite Cache ID 6wOYSJxnU].

[23] Luong, M.T.; Pham, H.; Manning, C.D. Effective approaches to attention-based neural machine translation. arXiv preprint 2015, arXiv:1508.04025.