IAS-BERT: An Information Gain Association Vector Semi-supervised BERT Model for Sentiment Analysis

Linkun Zhang(✉), Yuxia Lei, and Zhengyan Wang

Qufu Normal University, Rizhao 276800, Shandong, China
ilinkzhang@gmail.com

Abstract. With the popularity of large-scale corpora, statistics-based models have become mainstream model in Natural Language Processing (NLP). The Bidirectional Encoder Representations from Transformers (BERT), as one of those models, has achieved excellent results in various tasks of NLP since its emergence. But it still has shortcomings, such as poor capability of extracting local features and exploding of training gradients. After analyzing the shortcomings of BERT, this paper proposed an Information-gain Association Vector Semi-supervised Bidirectional Encoder Representations from Transformers (IAS-BERT) model, which improves the capability of capturing local features. Considering the influence of feature's polarity to overall sentiment and association between two word-embeddings, we use information gain on the training corpus. And then, the information gain results are used as an annotation of training corpus to generate a new word embedding. At the same time, we use forward-matching to optimize the computational overhead of IAS-BERT. We experiment the model on dataset of sentiment analysis, and it have achieved good results.

Keywords: Information gain · Semi-supervised · Local feature

1 Introduction

Sentiment analysis is one of the important components in Natural Language Processing (NLP), and is the main branch of text classification. It helps us to understand human emotional behavior, and be used in many places, such as movie evaluation [1], opinion mining [2], behavior prediction [3], social network [4], etc. There are many sentiment analysis models with excellent effects, like Neural Network Language (NNL) [5], Support Vector Machines (SVM) [6], Word2vec [7], Embedding from Language Models (ELMo) [8], Generative Pre-Training (GPT) [9] and the Bidirectional Encoder Representation from Transformers (BERT) [10]. The task of sentiment analysis can be understood as reflecting specific corpus to different sentiment categories [11].

In 2003, Yoshua Bengio proposed Neural Network Language (NNL) [5]. And then, NLP methods based on neural networks was impacted by what based on Support Vector

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2021
Published by Springer Nature Switzerland AG 2021. All Rights Reserved
L. Qi et al. (Eds.): CloudComp 2020, LNICST 363, pp. 31–42, 2021.
https://doi.org/10.1007/978-3-030-69992-5_3

Machines (SVM) [6]. SVM achieves good results with less samples. However, its robustness of missing data is poor, and interpretation of high-dimensional kernel functions is not strong.

In 2013, the Google team led by Tomas Mikolov proposed Word2vec method [7]. They proposed two important models: Continuous Bag-Of-Words Model (CBOW) and Continuous Skip-gram Model (Skip-gram) [12]. Word2vec transforms word embeddings from high-dimensional and sparse representation to low-dimensional and dense representation. At the same time, it considers the context information, and makes semantic information more accurate [13].

In 2018, Matthew E. Peter proposed Embedding from Language Models (ELMo) [8]. ELMo solves ambiguity problem by saving multiple word embeddings of a word [14]. OpenAI proposed the Generative Pre-Training (GPT) [9]. GPT has made great progress in feature extraction [15]. In October, Google proposed the Bidirectional Encoder Representation from Transformers (BERT) [10]. BERT is a general model for natural language processing. Compared with the previous model, this model has achieved good results on most tasks.

Although the existed models, especially BERT, are very helpful for sentiment analysis, their structure without RNN and CNN makes their capability of capturing local features reduced [16, 17]. Due to the structure of Transformer, BERT performs poorly of capturing sequential sequences [18]. In later study, constructing auxiliary sentences [19] and window slicing [20] were also proposed, but their improvement is not obvious.

The contributions of this paper are as follows:

1. In order to improve model's capability of capturing local features and get better training gradient, we propose a new model called IAS-BERT. In this model, we simulate RNN structure to improve the model's understanding capability of sentence deeper implication. In terms of processing sequence information, RNN-like structure performs better than single sequence rearrangement structure of Transformer.
2. Considering the effect of feature polarity frequency in corpus during model learning, we balance feature polarity for word embedding. By annotating the corpus with information gain and association vector results, it learns the knowledge about different features frequency on model training. It improves accuracy by balancing features polarity frequency.
3. IAS-BERT is scientifically tested on English public datasets CoLA, SST-2 and Chinese public datasets waimai_10k, weibo_senti_100k to prove the valid of it.

In what follows, the framework of IAS-BERT will be introduced in Sect. 2. The results of IAS-BERT on experimental dataset will be introduced in Sect. 3. Finally, the paper is concluded in Sect. 4.

2 Model and Method

2.1 Framework of IAS-BERT

IAS-BERT is improved on $\text{BERT}_{\text{BASE}}$. The description of $\text{BERT}_{\text{BASE}}$ is as follows [10]:

$$P(\omega_t = ''BERT''|\omega_1, \omega_2, ..., \omega_{(t-1)}; \theta) \tag{1}$$

$$L = \sum_{\omega \in C} \log^{P(\omega|context(\omega))} \tag{2}$$

This model is a stack of multilayer bidirectional Transformer encoders and decoders. In IAS-BERT, its presentation layers, hidden layers, self-attention heads and feedforward size are 12, 768, 12 and 3072 respectively.

The input is word embeddings. In general, word embedding is constituted by token embedding, segment embedding and position embedding. However, the input of IAS-BERT consists by token embedding, segment embedding, position embedding and annotation embedding. We use information gain as corpus annotation to generate new word embeddings. It will be outputted after multi-layer encoding and decoding (see Fig. 1).

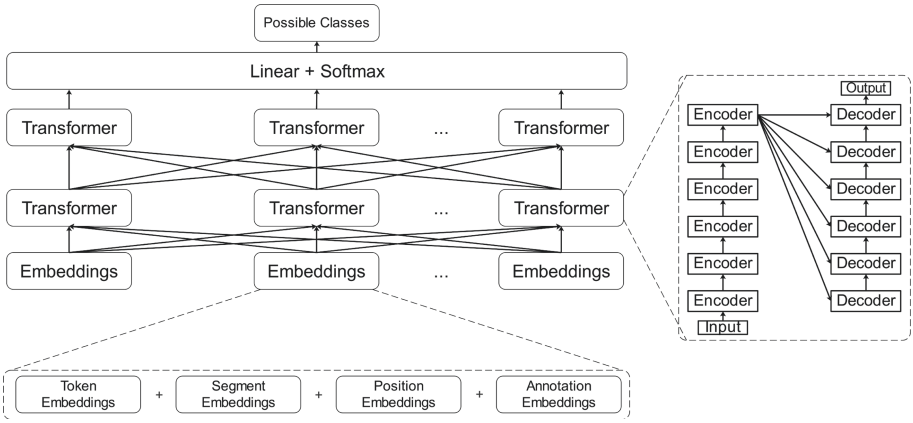

Fig. 1. Framework of IAS-BERT

2.2 Encoder and Decoder of IAS-BERT

In the Transformer, encoder is composed with two layers: Self-attention and Feed Forward. Decoder is composed with three layers: Self-attention, Encode-Decode Attention, and Feed Forward [15].

IAS-BERT's Encoder and Decoder are add annotation classifier layer before Self-attention layer to balance the weight of features polarity. Self-attention layer focuses on predicted words of current word embedding. Encode-decode attention layer focuses on unpredicted words of current word embedding. IAS-BERT can eliminate the impact on polarity weights between high-frequency words and low-frequency words by this design, and it would enhance the understanding of training corpus (see Fig. 2).

In LSTM-models, some researchers have proposed methods for sparse self-attention to solve sentence implication, and those methods have achieved better results [21]. Therefore, we add annotation embeddings by information gain and association vector to improve the understanding of local features on sentences. It will balance the weight of features polarity by annotation classifier. After this process, the optimized weight and

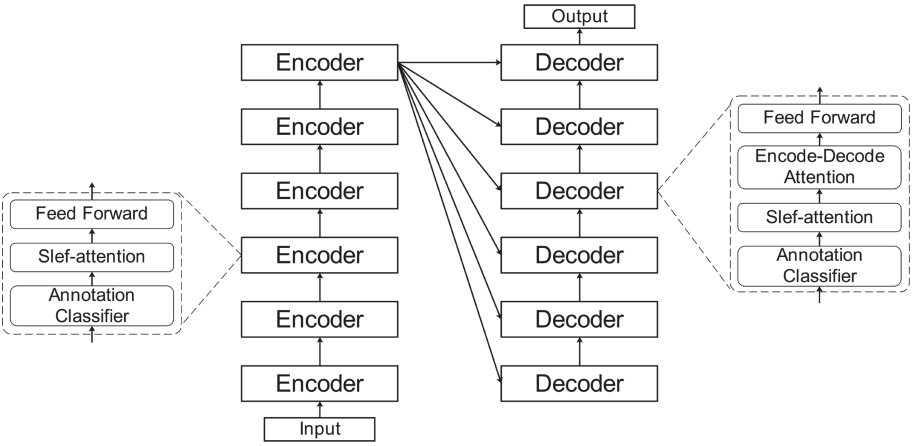

Fig. 2. Encode and Decode of IAS-BERT

association vector based on residual calculation will be added to the residual weight. It solves the problem of unilaterally reordering sequence and ignoring the weight of internal association of feature words during training.

2.3 Annotation Classifier of IAS-BERT

Different polarity has different effects on features [22]. The $\text{BERT}_{\text{BASE}}$ pay more attention to high-frequency features rather than low-frequency. Information gain can well solve the problem of low-frequency features' weight. The calculation of information gain is as follows:

$$IG(E) = H(C) - H(C|E) \tag{3}$$

where $H(C)$ represents information entropy and $H(C|E)$ represents conditional entropy.

$$H(C) = -\sum_{i=1}^{n} p(C_i) \log_2^{p(C_i)} \tag{4}$$

$$H(C|E) = \sum_{i=1}^{n} p(e_i) H(C|E = e_i) \tag{5}$$

In sentiment analysis, $T = t$ usually indicates existing feature word, and $T = \bar{t}$ usually indicates opposite situation. The conditional entropy is calculated by this definition.

$$H(C|E) = p(t)H(C|t) + p(\bar{t})H(C|\bar{t}) = -p(t)\sum_{i=1}^{n} p(C_i|t) \log^{p(C_i|t)} - p(\bar{t})\sum_{i=1}^{n} p(C_i|\bar{t}) \log^{p(C_i|\bar{t})} \tag{6}$$

Id	Negative	Positive	Sentiment
1	Y	Y	Negative
2	N	N	Negative
3	Y	N	Positive
4	N	Y	Negative
5	Y	Y	Negative
6	Y	N	Negative
7	Y	N	Positive
8	N	N	Negative
9	N	N	Positive
10	N	N	Negative
11	N	Y	Negative
12	Y	N	Positive
13	N	N	Positive
14	Y	Y	Negative
15	Y	Y	Negative

⇒

	Negative		Positive		Total
	Y	N	Y	N	
Positive	3	2	0	5	5
Negative	5	5	6	4	10
Sum	8	7	6	9	15

Fig. 3. Annotation training corpus (The Negative column represents the feature of negative polarity in the corpus, and the Positive column represents the feature of positive polarity in the corpus. The Y means appear, and the N means not appear. The Sentiment column represents the sentiment tendency of the corpus.)

IAS-BERT gets the information gain of training corpus by following steps. Firstly, we annotation the training corpus (see Fig. 3).

After calculation, results are as follows:

$$H(C) \approx 0.9182$$

$$H_{Negative=Y}(C|E) \approx 0.9543$$

$$H_{Negative=N}(C|E) \approx 0.8631$$

$$H_{Positive=Y}(C|E) = 0$$

$$H_{Positive=N}(C|E) \approx 0.2983$$

$$IG_{Negative}(E) \approx 0.0065$$

$$IG_{Positive}(E) \approx 0.7392$$

According to results, in example corpus, the information gain of Positive is bigger than that of Negative. In other words, Positive is more important than Negative. However, Negative's word frequency is higher than Positive. In summary, the information gain of IAS-BERT can solve the problem about imbalance corpus polarity caused by word frequency.

Association vectors focus on the internal features of successively entered words. In this way, the relevance between two word-embeddings is improved, and the model learns associative memory.

$$A(E_1, E_2) = \sum_{i=1}^{n} \delta(E_{1i}, E_1)(E_{2i}, E_2) = \frac{\sum_{i=1}^{n} E_{1i} \times E_{2i}}{\sqrt{\sum_{i=1}^{n} E_{1i}^2 \sum_{i=1}^{n} E_{2i}^2}} \tag{7}$$

where E_{1i} represents i th of the first input, and E_{2i} represents i th of the follow word. The inputted word embedding is represented as $E = E_1 E_2 ... E_i ... E_n$.

The weight function is set in annotation classifier layer as follows:

$$w_j = \frac{IG(D, a_j)}{\sum_{j=1}^{K} IG(D, a_j)} + \frac{\sum_{i=1}^{n} E_{1i} \times E_{2i}}{\sqrt{\sum_{i=1}^{n} E_{1i}^2 \sum_{i=1}^{n} E_{2i}^2}} \tag{8}$$

where, a_j represents the information gain of j th feature in the corpus D. In addition, the ReLU activation function is usually added to two linear transformations in feedforward network layer [15].

$$FFN(Z) = max(0, ZW_1 + b_1)W_2 + b_2 \tag{9}$$

After normalized by w_j and $FFN(Z)$, it will pass to the next encoder (see Fig. 4).

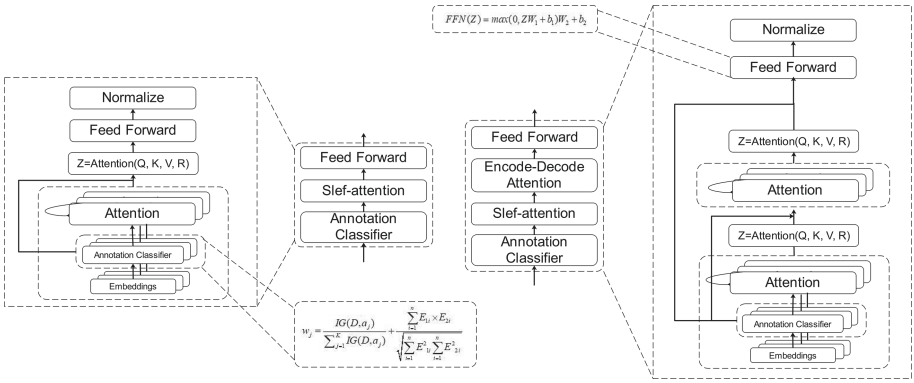

Fig. 4. Annotation classifier of IAS-BERT

2.4 Computational Overhead of IAS-BERT

IAS-BERT uses forward-matching regular expression form, and combines with following loss function:

$$L = \sum_{r \in R} l(\{e_i, P(e_i|E, PV; \hat{\theta}) | i \in S_r\}) \tag{10}$$

$$L = \sum_{r \in R} \sum_{i \in S_r} e_i \log^{P(e_i|E,PV;\hat{\theta})} \tag{11}$$

In this way, the gradients in training is optimized. Using a forward-matching regular expression can reduce the computational overhead for deep networks. The forward-matching regular expression is as follows:

$$\frac{\partial \varepsilon}{\partial E_l} = \frac{\partial \varepsilon}{\partial E_L} \times (1 + \sum_{k=1}^{L-1} \frac{\partial \xi(LN(E_k), \theta_k)}{\partial E_L}) \tag{12}$$

Under the optimization of forward-matching regular expression, the state of $L + 1$ layer as follows: $x_{l+1} = Y(y_0, y_1, ..., y_l) = \sum_{k=0}^{l} W_k^{l+1} LN(e_k)$.

Some studies have pointed out that pre-processing corpus can achieve better results [23, 24]. IAS-BERT mainly improves capturing local features of $\text{BERT}_{\text{BASE}}$ by information gain and association vector.

3 Experiment

3.1 Datasets

In this paper, we use CoLA, SST-2, waimai_10k and weibo_senti_100k to verify the validity of IAS-BERT [25]. Where CoLA is a classification corpus, and its classification labels are unbalanced. The purpose of adding this database is to verify the scalability of IAS-BERT in other problems in NLP. Table 1 summarizes the details of each dataset.

Table 1. Information of datasets

Dataset	Metric	Train			Dev	Test
		Total	Positive	Negative		
CoLA	MCC	8551	6023	2528	527	516
SST-2	ACC	6119	3195	2924	3984	2391
waimai_10k	ACC	10798	3599	7199	3258	4102
weibo_senti_100k	ACC	110289	54410	55879	5648	30635

Note: Table 1 includes metrics for model, training/dev/test sizes (in number of sentences), and positive/negative samples.

These datasets contain sentiment and labels. We randomly shuffle the datasets firstly to make experiment repeatable. And then, we cut out 60% from entire datasets as the training, 20% as the dev, and the remaining 20% as the test. Finally, due to the requirements of model, we add the start tag [CLS] at the beginning of each sentence and the end tag [SEP] at the end of each sentence.

3.2 Parameter Setting

In training, we set learning function as follows:

$$l = d_{model}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5}) \tag{13}$$

After setting learning function, we use the optimal one. Considering the impact of training epoch times, we experiment 20 epochs (see Fig. 5).

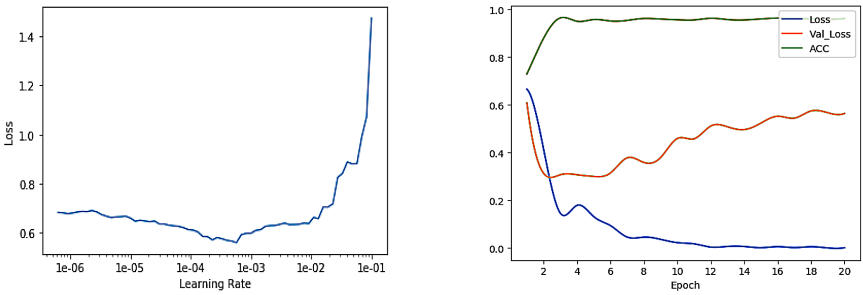

Fig. 5. Parameter of IAS-BERT

From Fig. 5, we get the maximum learning rate before the loss increasing sharply. The maximum learning rate is $e^{-6} + e^{-5}$. Thus, we set the learning rate as $2(e^{-6} + e^{-5})$. When the epoch is less than 3, Val_loss and Loss decrease together. It shows that model has achieved good results. When the epoch is more than 3, Loss is still decrease, but Val_loss has begun to increase. It shows that model works well on the training, but not well on the dev and test. This situation indicates that training has overfitted. In general, when setting epochs is 3, the training achieves best results.

3.3 Iteration and Loss

In this section, we use BiLSTM and $\text{BERT}_{\text{BASE}}$ comparing with our new model (see Fig. 6).

From Fig. 6, at the beginning of iteration, IAS-BERT has the best gradient among three models, and it is the fastest optimized model. As the iteration goes on, the loss of three models being consistency gradually. At the end of the iteration, IAS-BERT finds the optimum loss firstly. At the same time, the loss of IAS-BERT and $\text{BERT}_{\text{BASE}}$ still fluctuate within a larger range, while BiLSTM fluctuates within a smaller range. It is caused by different framework between Transformer and LSTM.

3.4 Experimental Results

Evaluation Metrics

In order to rigorous analyze IAS-BERT, we adopted multiple measures, including precision (P), recall (R), F1-score (F1), Matthews Correlation Coefficient (MCC), accuracy (ACC) and area under curve (AUC).

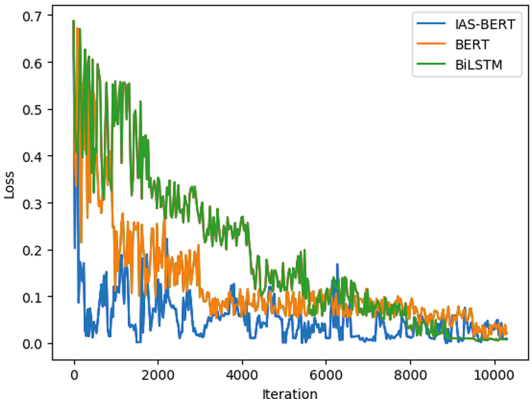

Fig. 6. Iteration VS loss for IAS-BERT, $\text{BERT}_{\text{BASE}}$ and BiLSTM

Accuracy, recall, and F1-score are a set of metrics widely used in model evaluation. *TP* indicates that the sample is positive, and the model predicts it as a positive sample. *TN* indicates that the sample is negative, and the model predicts it as a negative sample. *FP* indicates that the sample is negative, and the model predicts it as a positive sample. *FN* indicates that the sample is positive, and the model predicts it as a negative sample.

The precision is the ratio of positive samples classified correctly to samples classified as positive.

$$P = \frac{TP}{TP + FP} \tag{14}$$

The recall is the ratio of positive samples correctly classified to actual positive samples.

$$R = \frac{TP}{TP + FN} \tag{15}$$

F1-score is the weighted average of precision and recall.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{16}$$

MCC is the Matthews Correlation Coefficient. This indicator comprehensively considers *TP*, *TN*, *FP* and *FN*. It has a good effect on different quantity between positive and negative samples.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{17}$$

ACC is the ratio of correctly classified test samples to the total number of test samples. It is used to measure the capability of a model to correctly predict classification of new data.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{18}$$

AUC is an area under the ROC curve, which is used to judge the validity of models.

$$AUC = \frac{\sum ins_i \in positiveclassRank_{ins_i} - \frac{M \times (M+1)}{2}}{M \times N} \tag{19}$$

where, $\sum ins_i \in positiveclass$ and $Rank_{ins_i}$ represents randomly selected samples in the area.

Results

Table 2 records the results of IAS-BERT on datasets.

Table 2. Results of IAS-BERT on datasets

Dataset	ACC	F1	AUC	MCC	Precision	Recall
CoLA	74.2	82.6	65.5	42.9	77.0	89.0
SST-2	96.5	96.6	96.5	93.0	96.7	96.6
waimai_10k	95.2	92.7	94.3	89.1	93.7	91.8
weibo_senti_100k	98.1	98.1	98.1	96.2	99.6	96.6

From Table 2, both English datasets and Chinese datasets are achieved good results. And with the increasing scale of data in training, the score of IAS-BERRT is improved.

Table 3. Score comparison of models on CoLA, SST-2, waimai_10k and weibo_senti_100k

Model	CoLA	SST-2	waimai_10k	weibo_senti_100k
	MCC	ACC	ACC	ACC
CBOW	–	80.0	–	–
LSTM	–	84.9	93.1	95.2
BiLSTM	11.6	82.2	94.3	96.0
DCNN	–	86.8	–	–
DSCNN	–	89.1	–	–
Pre-OpenAI SOTA	35.0	93.2	–	–
BiLSTM + ELMo	32.1	89.3	–	–
BiLSTM + ELMo + Attn	36.0	90.4	–	–
OpenAI GPT	45.4	91.3	–	–
$BERT_{BASE}$	52.1	93.5	90.1	97.9
$BERT_{LARGE}$	60.5	94.9	–	–
Mobile BERT	51.1	92.6	–	–
IAS-BERT	42.9	96.5	95.2	98.1

We compare the results with CBOW, LSTM, BiLSTM, DCNN, DSCNN, Pre-OpenAI SOTA, BiLSTM + ELMo, BiLSTM + ELMo + Attn, OpenAI GPT, $\text{BERT}_{\text{BASE}}$, $\text{BERT}_{\text{LARGE}}$ and Mobile BERT [7, 10, 12, 16, 26]. Table 3 records the comparison results.

According to Table 3, we can draw the following conclusions:

Although IAS-BERT achieved higher-than-benchmark scores in CoLA, it was still lower than OpenAI GPT, $\text{BERT}_{\text{BASE}}$, $\text{BERT}_{\text{LARGE}}$ and Mobile BERT. It shows that the adaptability of IAS-BERT is poor when the proportion of positive samples and negative samples in the corpus huge different. However, in SST-2, waimai_10k and weibo_senti_100k, it has achieved higher scores than existed models. It shows that our model is valid on emotion equipartition characteristics datasets.

4 Conclusions

In this paper, we proposed a model called IAS-BERT for sentiment analysis in NLP. Comparing with existed models, it improves the capability of extracting local features by association vector. And it balances the polarity of training corpus features by information gain annotation. Our new model achieves good results in both English datasets and Chinese datasets. However, due to the framework of Transformer, IAS-BERT lacks the capability of capturing sequence order and processing long sequence.

For future work, we will focus on improving the adaptability of long sequence.

Funding. This work is partly supported by the Undergraduate Education Reform Project in Shandong Province (no. Z2018S022).

References

1. Palkar, R.K., Gala, K.D., Shah, M.M., Shah, J.N.: Comparative evaluation of supervised learning algorithms for sentiment analysis of movie reviews. Int. J. Comput. Appl. **142**(1), 20–26 (2016)
2. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: LREc, vol. 10, no. 2010, pp. 1320–1326, May 2010
3. Sisk, J.: U.S. Patent Application No. 13/308,496 (2013)
4. Deitrick, W., Hu, W.: Mutually enhancing community detection and sentiment analysis on twitter networks (2013)
5. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**(Feb), 1137–1155 (2003)
6. Chang, Y.W., Hsieh, C.J., Chang, K.W., Ringgaard, M., Lin, C.J.: Training and testing low-degree polynomial data mappings via linear SVM. J. Mach. Learn. Res. **11**(4), 1471–1490 (2010)
7. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In International Conference on Machine Learning, pp. 1188–1196, January 2014
8. Peters, M.E., et al.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
9. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

11. Soh, C., Yu, S., Narayanan, A., Duraisamy, S., Chen, L.: Employee profiling via aspect-based sentiment and network for insider threats detection. Expert Syst. Appl. **135**, 351–361 (2019)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
13. Rong, X.: word2vec parameter learning explained. arXiv preprint arXiv:1411.2738 (2014)
14. Cheng, J., Dong, L., Lapata, M.: Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733 (2016)
15. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
16. Santos, I., Nedjah, N., de Macedo Mourelle, L.: Sentiment analysis using convolutional neural network with fastText embeddings. In: 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI), pp. 1–5. IEEE, November, 2017
17. Lee, J., et al.: BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
18. Alsentzer, E., et al.: Publicly available clinical BERT embeddings. arXiv preprint arXiv:1904.03323 (2019)
19. Sun, C., Huang, L., Qiu, X.: Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. arXiv preprint arXiv:1903.09588 (2019)
20. Lei, Y., Wu, Z.: Time series classification based on statistical features. EURASIP J. Wireless Commun. Netw. **2020**(1), 1–13 (2020). https://doi.org/10.1186/s13638-020-1661-4
21. Deng, D., Jing, L., Yu, J., Sun, S.: Sparse self-attention LSTM for sentiment lexicon construction. IEEE/ACM Trans. Audio Speech Lang. Process. **27**(11), 1777–1790 (2019)
22. Tang, J., et al.: Progressive self-supervised attention learning for aspect-level sentiment analysis. arXiv preprint arXiv:1906.01213 (2019)
23. Chen, T., Xu, R., He, Y., Wang, X.: Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. Expert Syst. Appl. **72**, 221–230 (2017)
24. Tien, N.H., Le, N.M., Tomohiro, Y., Tatsuya, I.: Sentence modeling via multiple word embeddings and multi-level comparison for semantic textual similarity. Inf. Process. Manage. **56**(6), 102090 (2019)
25. Warstadt, A., Singh, A., Bowman, S.R.: Neural network acceptability judgments. Trans. Assoc. Comput. Linguist. **7**, 625–641 (2019)
26. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)