International Journal on Semantic Web and Information Systems
Volume 19 • Issue 1

HBert:
A Long Text Processing Method Based on BERT and Hierarchical Attention Mechanisms

Xueqiang Lv, Beijing Information Science and Technology University, China
Zhaonan Liu, Beijing Information Science and Technology University, China
Ying Zhao, Beijing Information Science and Technology University, China
Ge Xu, Minjiang University, China
Xindong You, Beijing Information Science and Technology University, China*

ABSTRACT

With the emergence of a large-scale pre-training model based on the transformer model, the effect of all-natural language processing tasks has been pushed to a new level. However, due to the high complexity of the transformer's self-attention mechanism, these models have poor processing ability for long text. Aiming at solving this problem, a long text processing method named HBert based on Bert and hierarchical attention neural network is proposed. Firstly, the long text is divided into multiple sentences whose vectors are obtained through the word encoder composed of Bert and the word attention layer. And the article vector is obtained through the sentence encoder that is composed of transformer and sentence attention. Then the article vector is used to complete the subsequent tasks. The experimental results show that the proposed HBert method achieves good results in text classification and QA tasks. The F1 value is 95.7% in longer text classification tasks and 75.2% in QA tasks, which are better than the state-of-the-art model longformer.

KEYWORDS

BERT, Hierarchical Attention, Long Text Processing

INTRODUCTION

The Transformer (Vaswani et al., 2017) model achieves excellent results in many natural language processing tasks, including classification, text generation, etc., while advancing the birth of many Transformer-based large-scale pre-trained models such as BERT (Devlin et al., 2018). The Transformer model computes the attention between each token in a sentence by self-attention, obtains the semantic information of each word and the semantic relations between words, and uses positional encoding to obtain the positional information of each word in each sentence. It can capture the whole context of a sequence in these two ways, which also leads to the success of the Transformer model. However, due to Transformer's self-attention mechanism complexity, the time

DOI: 10.4018/IJSWIS.322769 *Corresponding Author

complexity of the Transformer is $O\left(n^2 d\right)$ (where n is the sequence length and d is the dimension of the hidden layer), which results in a limited length of the text that can be processed. Theoretically, the Transformer model can input text of arbitrary length, but due to its large complexity the Transformer cannot handle excessively long text in practical applications. The maximum length that can be processed depends on the actual situation. This also leads to a limited sequence length that Transformer-based models such as BERT can process, which is generally limited by the performance of computer hardware. While the BERT model limits the maximum input sequence length to 512 tokens, this does not mean that the Bert model can handle sentences of 512 words. The BERT model's tokenizer divides an input word into multiple subwords and adds various special tags, such as [CLS] and [SEP], which results in the length of text that BERT can handle being much less than 512 words. Since the number of subwords divided per word is not certain, the BERT does not have a fixed maximum input text length. It is certain that the maximum processable text length is less than 512 words. However, most of the texts, such as press releases, patent texts, etc., are much longer than 512 words. These longer texts are not as easy to process and cannot be processed directly by the BERT model, which limits its use in long texts processing.

Generally, the BERT processes the long text using four types of methods. The first is the truncation method in which the text of a certain length at the beginning or end of the text will be truncated and processed as the original text. This method retains only a small section of text at the beginning or end of the text while the rest of the text is discarded. This method loses a lot of text information. The second is the segmentation method in which the long text is divided into multiple short texts. This method truncates the text into multiple parts according to a fixed length. Each part is encoded using the model to obtain a vector and then the obtained vector is stitched to obtain the text vector. The third method is the compression method in which the long text is divided into multiple short texts and then the meaningless paragraphs are selected and deleted using rules or training with other filtering or scoring models. The effect of the compression method is severely limited to the effect of the filtering method. The fourth method involves changing the structure of the model because the high complexity of the Transformer mainly lies in the self-attention mechanism. The complexity of the self-attention mechanism is reduced by this method through limiting the scope and the way of capturing information, thereby improving the model's ability to process long text.

Aiming at solving some drawbacks of the existing methods mentioned above, the HBert method is proposed in this paper to improve the ability of the BERT model to process long text. In summary, the main contributions of this article are as follows:

1. A long text hierarchical processing method based on BERT, named HBert, is proposed in this paper, which belongs to the segmentation method. The missing inter-segment information of the segment is compensated through the hierarchical mechanism, and then the text vector is obtained for the downstream tasks.
2. The effectiveness and advantages of the proposed method is demonstrated by the classification task and the QA task, which are conducted on three public datasets: IMDb (Maas et al., 2011), Hyperpartisan (Kiesel et al., 2019), and WikiHop (Welbl et al., 2018).
3. The importance and validity of the model with the increased attention to words and sentences between layers are verified by the ablation experiment.

BACKGROUND

To enable Transformer-based models to be applied to long text processing and achieve better results, most of the current research is focused on optimizing the self-attention mechanism, which can reduce complexity to improve the model's ability to process long text. Transformer-XL (Dai et al., 2019) model divides the long text into multiple fixed-length paragraphs. The sequence of

hidden vectors of all layers in the previous paragraph are cached and utilized when processing the current paragraph. All sequences of hidden vectors in the previous paragraph only participate in forwarding calculations but no backpropagation, which can compensate for some of the inter-segment information lost due to paragraph division but is less effective for long-range dependence. XLNet (Yang et al., 2019) uses the dual-stream attention mechanism that contains query Stream and Content Stream while proposing PLM (Permutation Language Model). Transformer-XL is used as a model encoder, which makes it superior to BERT not only in terms of the text length that can be processed but also the effect of the model. However, Transformer-XL is a typical autoregressive pre-trained model, and its effectiveness relies on the autoregressive properties of the text itself. Longformer (Beiltagy et al., 2020) model was proposed to replace the Transformer's self-attention mechanism by the three attention modes with sliding window, telescopic sliding window, and global attention. These three modes of attention are conducted at the same time, which causes the complexity of the attention mechanism to increase linearly with the length of the text. The optimal results are achieved in multiple datasets and are better than RoBERTa (Liu et al., 2019) on all datasets. The BigBird model (Zaheer et al., 2020) adds random attention to the three longformer attention patterns, with no increase in the complexity. A large number of experiments demonstrate that the effect is superior to RoBERTa and longformer. Due to the BERT being limited by its inability to deal with long text effectively and because there is no fine-tuning of long text classification, the DocBERT (Adhikari et al., 2019) model was proposed. The long text classification task was used to fine-tune the model while using the knowledge distillation method to migrate the model to a single-layer bidirectional LSTM model, which can be more efficient in processing long text, but the effectiveness of the model is reduced.

Some studies adapt text length to the BERT model by changing the way of text preprocessing to shorten the text length. Single-layer neural networks and hierarchical LSTM are combined to filter sentences in the long text and applied in long text filtering tasks (Cao et al., 2019). The length of the text was shortened by deleting sentences with low relevance and dependence on the topic. By constructing a text filtering network consisting of a single-layer bidirectional LSTM network, the attention mechanism can select statements of greater importance in long texts and shorten the length of the original text. Then the processed text was inputted into the BERT model (Wang et al., 2020). However, this method relies too much on filtering network performance.

Currently, the idea of layering is widely used in the field of natural language processing. Hierarchical Attention Networks (HAN) was applied to text classification task for the first time (Yang et al., 2016), which consists of two parts: the word encoder and the sentence encoder. The BiGRU was used for both the word encoder and sentence encoder. HAHNN (Abreu et al., 2019) improved the hierarchical attention neural network by extracting features through the convolution layer before inputting the word vector into the word encoder. The two-part model was modified by replacing the word encoder and the sentence encoder (Pappagari et al., 2019). BERT model was used for the word encoder and the BiLSTM model was used for the sentence encoder. The improved model achieves good results on multiple datasets. HAN was also used for long text classification tasks (Che et al., 2019), in which the distributed word vector was used for text title representation and the text vector was integrated for text classification. The layering idea of the hierarchical neural network can divide the long text into multiple parts and each part meets the input requirements of BERT. Then, multiple parts were organically combined to retain both word-level information and inter-paragraph information, which is an effective way to improve BERT's ability to process long text.

As described above, existing methods shorten the text length or adjust the self-attention mechanism to reduce the complexity, which allows models to process long text. However, these methods will lose some text information and limit the effect of the model. The HBert proposed in this article not only retains the text information but also pays attention to the relationship between words and sentences in the text, which can improve the ability to process long text and reduce complexity of the model.

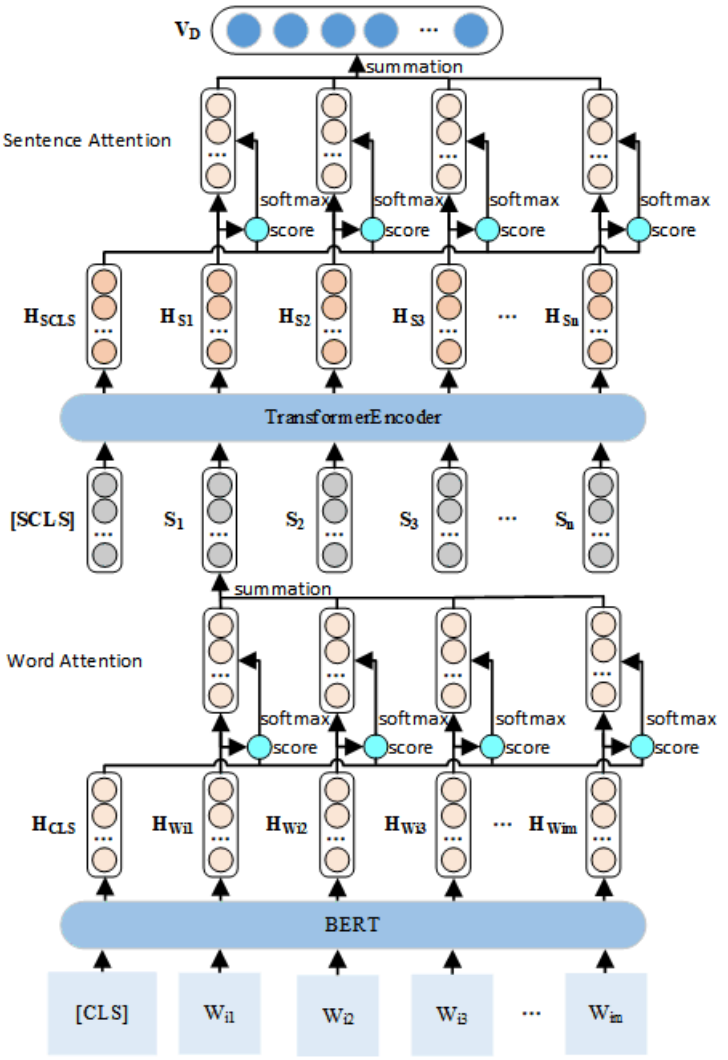
MODEL

Text Vectors

The model hierarchically acquires the text vector to complete the downstream task, as shown by the model structure in Figure 1.

Text is divided by sentence first. There is no guarantee that the position will split the meaning of the text, if the sentence contains the linguistic convention semantic information, structural information, and associated information about part of the context is divided by a fixed length. Semantic information refers to the meaning or content conveyed by words or phrases. It concerns the interpretation of words in isolation or in combination with other words in a sentence. Semantic information includes lexical and grammatical meanings, as well as connotations and associations. Structural information refers to the way that words are organized within a sentence or utterance. It includes information about the relationships between words, such as their syntactic roles (e.g., subject,

Figure 1. The architecture of HBert



object, verb), their grammatical forms (e.g., tense, aspect, mood), and the overall structure of the sentence. Contextual information refers to information that is not explicitly stated in the language itself but is instead tied to a particular element of the context, such as the situation, the speaker, or the listener. It includes information about the speaker's intentions, the listener's background knowledge, and the broader social and cultural context in which the language is being used. Thus, each paragraph of long text can be represented as a collection of multiple sentences, for instance $D = \left(sen_1, \ldots, sen_i, \ldots, sen_n\right)$, where D represents text, sen_i represents the ith sentence of the text sentences, 1, 2... *n* indicates that the text contains *n* sentences. Where the sentence sen_i can be expressed as $sen_i = \left(w_{i1}, \ldots, w_{ii}, \ldots, w_{im}\right)$, w_{ii} represents the ith word of the sen_i, 1, 2... *m* indicates that the sentence consists of *m* words. Two granularities of text information can be obtained by the BERT model. The word attention method is used to compare all word granularity vectors with sentence granularity vectors, and the weighted sum is used to get sentence vectors. The calculation formula is shown in (1)-(2):

$$\left(H_{CLS}, H_{Wi1}, \ldots, H_{Wim}\right) = Bert\left(sen_i, \theta\right) \tag{1}$$

$$S_i = W\_Att\left(H_{CLS}, H_{Wi1}, \ldots, H_{Wim}\right) \tag{2}$$

where the H_{CLS} represents the corresponding vector of [CLS] output by BERT, θ represents the parameter of the BERT model, $H_{Wi1}, \ldots, H_{Win}$ represents the 1st, 2nd... mth words correspond to the vector, S_i represents the sentence vector of the ith sentence, W_Att representing represents the calculation process of word attention.

Using the above method, sentence vector representation corresponding to n sentences can be obtained. First, the 768-dimensional vector [SCLS] is randomly generated according to the standard normal distribution. Then, after splicing all sentence vectors in the [SCLS] vector, the input vector is obtained. Input vectors are inputted into the Transformer encoder to get the last layer output: $\left(H_{SCLS}, H_{S1}, \ldots, H_{Sn}\right)$. The weighted sum of all vectors with sentence attention is used to obtain text vector. The calculation formula is shown in (3)-(4):

$$\left(H_{SCLS}, H_{S1}, \ldots, H_{Sn}\right) = T\left(S_{SCLS}, S_1, \ldots, S_n\right) \tag{3}$$

$$V_D = S\_Att\left(H_{SCLS}, H_{S1}, \ldots, H_{Sn}\right) \tag{4}$$

where the H_{SCLS} represents the [SCLS] vector of Transformer encoder output, $H_{S1}, \ldots, H_{Sn}$ represents the sentence vector output by the Transformer encoder, S_Att represents the sentence attention calculation process, and V_D represents the text vector.

Through this model, semantic information of both word granularity and sentence granularity can be obtained Moreover, longer text can be processed without deleting sentences while the maximum words that can be processed by the basic is 512.

Word Attention and Sentence Attention

To improve the effectiveness of the model, word attention and sentence attention mechanisms are introduced.

The purpose of word attention is to use the '[CLS]' corresponding vector output by BERT to obtain sentence granularity information while making full use of the granularity of each word output by the last layer. The formula for calculating word attention is shown in (5)-(7):

$$w = H_{CLS}(H_{Wi1},\ldots,H_{Wim})^T \tag{5}$$

$$weight = softmax\left(w\right) \tag{6}$$

$$W\_Att = weight\left(H_{Wi1},\ldots,H_{Wim}\right) \tag{7}$$

where w represents the similarity matrix of [CLS] and word vectors. $Weight$ represents the weight of each word vector, and the similarity matrix is obtained by the softmax function. W_Att represents the sentence vector calculated by word attention. The calculation formula is shown in (8):

$$softmax\left(z_i\right) = \frac{\exp\left(z_i\right)}{\sum_{t=1}^{m}\exp\left(z_t\right)} \tag{8}$$

The sentence vector corresponding to each sentence can be obtained through the above method. The sentence vector contains the information not only at the sentence level but also at the word level.

The purpose of sentence attention is to use the '[SCLS]' corresponding vector output by the last layer to obtain text granularity information while making full use of the granularity of each sentence output by the last layer. The formula for calculating word attention is shown in (9)-(11):

$$w\_s = H_{SCLS}(H_{S1},\ldots,H_{Sn})^T \tag{9}$$

$$weight\_s = softmax\left(w\_s\right) \tag{10}$$

$$S\_Att = weight\_s\left(H_{S1},\ldots,H_{Sn}\right) \tag{11}$$

where w_s represents the similarity matrix of [SCLS] and sentence vectors, $weight\_s$ represents the weight vector of each sentence vector, and S_Att represents the text vector calculated by sentence attention.

Sentence attention first multiplies the [SCLS] vector with other vectors to obtain a similarity matrix. The larger the weight is, the higher the correlation between sentence meaning and text

is. The text vector is obtained by using the calculated weights to obtain a weighted average of all sentence vectors.

Complexity

It is well known that the BERT model uses the encoder of the Transformer. The complexity of the BERT model is $O\left(n^2 d\right)$ due to the complexity of the Transformer is $O\left(n^2 d\right)$. This method divides the original text into k-segments, the complexity of processing each segment into BERT is $O\left(\frac{n^2}{k^2} d\right)$ due to the average length of each segment is $\frac{n}{k}$. The first layer of overall complexity is $O\left(k \frac{n^2}{k^2} d\right)$, i.e. $O\left(\frac{n^2}{k} d\right)$ because each segment needs to be processed using the BERT model. The complexity of the second layer is $O\left(k^2 d\right)$ because the Transformer encoder is used and the input length is the number of segments to be divided, that is *k*. In summary, the overall complexity of the method proposed in this article is shown as formula (12):

$$O = O\left(\frac{n^2}{k} d\right) + O\left(k^2 d\right) \tag{12}$$

and because n>>k, the overall time complexity is shown as formula (13):

$$O\left(\frac{n^2}{k} d\right) + O\left(k^2 d\right) \approx O\left(\frac{n^2}{k} d\right) \tag{13}$$

The complexity is reduced compared to the original BERT.

At the same time, the length of the input sequence becomes shorter, and the calculation of the position code is simpler because the long text is divided into multiple segments, which obtains a significant improvement in the overall execution efficiency of the method than the original BERT.

EXPERIMENTS

Data Set

- **IMDb Movie Reviews:** IMDb Movie Reviews is a binary classification dataset which contains 75,000 movie reviews from the Internet Movie Database (IMDb) and includes 50,000 training data and 25,000 test data. These data only considered highly polarized comments. The scores of the comments greater than 7 were selected as positive comments, while those less than 4 were selected as negative comments. Comments with scores of 5-6 were vaguely defined and could not be strictly classified as positive or negative, so they were ignored in the dataset construction.
- **Hyperpartisan News:** Hyperpartisan News is a binary classification dataset with an average text length longer than the IMDb dataset but contains only 645 pieces of manually labeled news data, which includes 238 positive data and 417 negative data, and there are few data volumes. During the experimentation, it was divided into training set, validation set, and test set according to a

9:1:1 ratio. To reduce the impact of the too-small amount of data on the experimental results, the experiment used different random number seeds five times, and the final experimental results were averaged.

- **Reuters-21578:** The Reuters-21578 dataset is a collection of documents with news articles. The original corpus has 10,369 documents and a vocabulary of 29,930 words. We used the ModApte split, removed documents belonging to multiple classes and considered only the eight classes with the highest number of training example.
- **WikiHop:** WikiHop is a QA dataset with the longest average text length. Each piece of data in the dataset consists of a piece of support text, a question, multiple candidate answers (minimum 2, maximum 79), and a real answer. The dataset has a total of 43,738 training data and 5,129 test data. During the experiment, each candidate answer is supplemented by [CLS] and spliced in the support text, all candidate answers [CLS] corresponding vectors are multiplied by text vectors to obtain the prediction results.

The statistics information for the three datasets is shown in Table 1.

Evaluation Indicators

The commonly used accuracy rate is used as an evaluation indicator for the IMDb dataset. The F1 value is used as an evaluation indicator for the Hyperpartisan datasets and WikiHop datasets.

Experimental Parameters

The experimental model in this paper was trained using a piece of Tesla v100 GPU. The results were obtained by performing each experiment three times and averaging the results. The BERT model is implemented by calling 'bert-base-case' in the Transformer package, and the generated vector dimension is 768. The BERT parameters are fine-tuned with the training of downstream tasks. In the Hyperpartisan dataset, several experiments have shown that the result is the local optimal value, so preheating and dynamic learning rates are used in the experiment. The change of learning rate is shown in Figure 2.

The other parameters of the model are shown in Table 2.

EXPERIMENTS

Comparative Experiment Settings

1. **HAN Model (Hierarchical Attention Network):** The baseline model used word2vec for word vectorization and obtains text vector through two hierarchical BiGRU.
2. **RoBERTa Model:** A model that has been improved for BERT by reducing the number of parameters of the model and improving the effectiveness of the pre-trained model.

Table 1 Experimental data set statistics information

Dataset	IMDb	Hyper	Reuters	Wiki
Quantity	75000	645	7674	48867
Average text length	256	560	104	1180
Average number of sentences	16	28	8	59
Average sentence length	16	20	13	20

Figure 2. Dynamic learning rate of the hyperpartisan data set

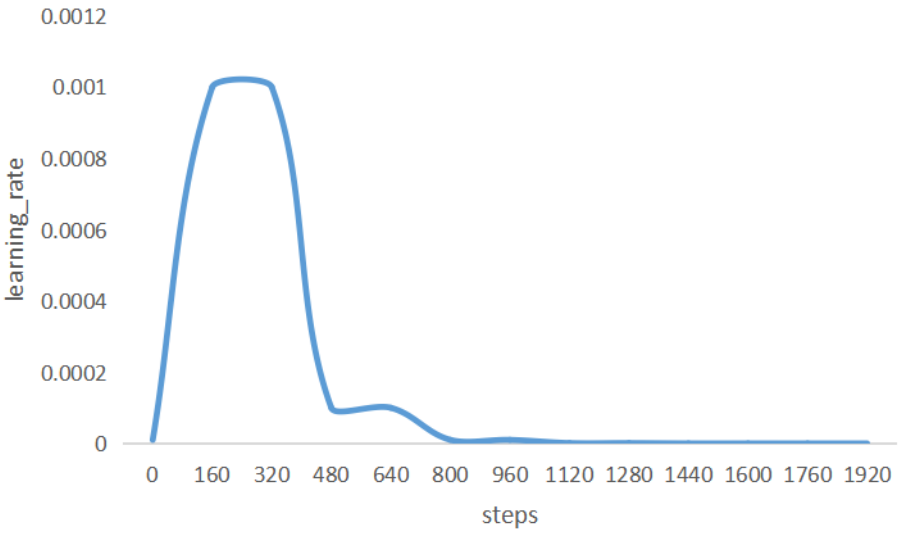


Table 2. Model hyperparameter settings

Parameter	The parameter value			
	IMDb	Hyper	Reusters	WikiHop
Batch_size	8	4	8	4
Dropout	0.5	0.2	0.5	0.5
Optimizer	Adam	Adam	Adam	Adam
Learning rate	0.00001	0.0000001	0.00001	0.00001
Encoder layers	4	2	4	4
Dimensions	768	768	768	768
Early stop round	5	10	5	5

3. **Longformer Model:** A newly proposed model that optimizes the shortcomings of the Transformer's self-attention mechanism and adapts the Transformer to longer text.
4. **HBert:** Long text processing method proposed in this paper, in which the word attention and the sentence attention are used to improve the effect.

Experimental Results

To verify the effectiveness of the HBert proposed in this paper, experiments were performed on IMDb, Hyperpartisan, and WikiHop datasets for classification tasks and QA tasks. The results of the experiments are shown in Table 3.

In the classification task, the results of the proposed HBert method on the IMDb data set are equal to RoBERTa and slightly lower than Longformer. The results on the Hyperpartisan dataset are 0.9% higher than Longformer. In addition, in the QA task, HBert is 0.2% higher than Longformer on the WikiHop dataset. The reason is that the method proposed in this paper can extract the full-text

Table 3. Experimental results of the different models

Model	Classify		QA	
	IMDb	Hyper	Reusters	Wiki
HAN	95.1	92.5	—	—
RoBERTa	95.3	87.4	95.8	72.4
longformer	95.7	94.8	96.9	75.0
HBert	95.3	95.7	96.9	75.2

features of the text better and the impact of the local features of the text is reduced compared to the longformer, which can achieve better results on longer text.

Ablation Experiments

To verify the importance of different parts of the model and the impact on the model, the ablation experiment is carried out, in which the different parts of the model are removed to verify the model effect. The specific experimental settings are described as follows:

1. **BERT+TransformerEncoder:** [CLS] corresponding to each sentence of text I input as a sentence vector into the Transformer encoder to obtain the text vector for classification.
2. **BERT+average+TransformerEncoder:** The BERT's last hidden state without the vector corresponding to [CLS] is averaged as a sentence vector, and it is input to the Transformer encoder to obtain the text vector for classification.
3. **BERT+IDF+TransformerEncoder:** The BERT's last hidden state without the vector corresponding to [CLS] is weighted average as a sentence vector according to the IDF value, and it is input to the Transformer encoder to obtain the text vector for classification.
4. **BERT+WordAttention+TransformerEncoder:** The last hidden layer output of BERT output is input to the word attention to obtain the sentence vector and it is input to the Transformer encoder to obtain the text vector for classification.
5. **BERT+TransformerEncoder+SentenceAttention:** The BERT's last hidden state without the [CLS] vector is averaged as a sentence vector, [SCLS] is added in front of all sentence vectors of the text, and then sentence attention is used to get the text vector for classification.
6. **HBert:** The output of the last hidden layer of BERT is added with word attention to obtaining sentence vector and [SCLS] is added before all sentence vectors of text. The output of the last layer of the Transformer encoder is added with sentence attention to obtain a text vector for classification.

The results of the ablation experiment are shown in Table 4.

The average value of the last hidden state of the BERT model is used as the sentence vector in the BERT+Transformer encoder, which improves the model effect by 0.4% and 1.8% respectively, proving that using [CLS] directly as the sentence vector cannot fully represent the sentence meaning. Word vector weighted average as the sentence vector with IDF as weight, the model effect decreased by 0.6% and 2.1% respectively. The results indicate that IDF is not suitable as a weighted sentence vector. The precision and f1 of the model is decreased by 0.4% and 0.7% on the IMDb dataset, 3.3% and 3.8% on the Hyperpartisan dataset after removing word attention and sentence attention respectively, which indicate that word attention solves the problem of ignoring the different importance of different words when directly finding the average of all word vectors as sentence vectors, and sentence attention makes up for the fact that there is no data in between the segments. Attention mechanism plays an important role in the model in selecting words and sentences that contain semantic information.

Table 4. Results of the ablation experiments

Experimental serial number	Dataset	
	IMDb	Hyperpartisan
1	93.9	87.8
2	94.3	89.6
3	93.7	87.5
4	94.6	91.9
5	94.9	92.4

Memory Usage

It has been found that HBert has lower complexity through mathematical analysis. To verify the lower complexity of the model through experiment, batch_size was set to 8 on the IMDb dataset for the experiment and compared with the Bert model and RoBERTa model. The experimental results are shown in Table 5.

Experimental results show that the average memory usage of the HBert is 2566MiB, far lower than the average memory usage of BERT 6766MiB and RoBERTa 7162MiB. The memory usage of HBert is about one-third of the BERT model and RoBERTa model. The reduction in HBert memory usage allows HBert to process longer text than BERT and RoBERTa can. It shows that HBert achieves better long text processing effect with only one third of BERT's memory.

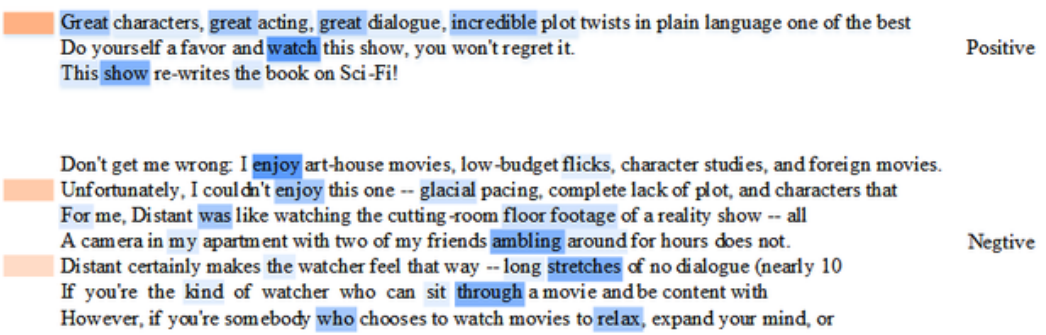
Attention Mechanism

To verify that word attention and sentence attention can help select words and sentences that contain semantic information, the weight of each word and sentence in two paragraphs of text is visualized in Figure 3.

Table 5. Comparison of model memory usage

Model	Average Memory Usage Mib
BERT-base	6766
RoBERTa-base	7162(+5.85%)
HBert	2566(-62.08%)

Figure 3. Visual results of attentional mechanisms



The two pieces of text are one positive and one negative from the IMDb dataset. Each line in the paragraph represents a sentence. Blue represents the weight of each word's attention: the greater the weight, the heavier the color. The orange represents the attention weight of the sentence represented by this line in the text: the greater the weight, the heavier the color.

From the figure it can be observed that the more words and sentences in the positive example that can indicate the critic's favorite for the movie, the greater the weight they are, such as 'great', 'incredible', and the first sentence of the positive example. The words and sentences in the negative cases that could indicate the critic's dislike for the film were given more weight, such as 'glacial', 'long stretches', and the second and fifth sentences of the negative cases, etc. Therefore, word attention and sentence attention can effectively increase the weight of words that better reflect the semantic information and emotion of the text, which is conducive to better extraction of word-level semantic features and sentence-level semantic features.

CONCLUSION

HBert is proposed in this paper in order to solve the problem that BERT can't process long text. Experimental results show that HBert has a good effect on long text classification and QA tasks, and the effect is comparable with other methods on the IMDb dataset with short text length. The precision and f1 are 0.9% and 0.2% higher than the Longformer model on the Hyperpartisan dataset and the WikiHop dataset, respectively. The ablation experiments verify the importance of each component of the model and the effectiveness of the attentional mechanism. Through attention mechanism visualization, it is found that the more words and sentences representing the author's emotions, the more attention weight they are got. At the same time, the memory usage of HBert is 62.08% lower than Bert. We only studied the effect of the method in the text classification task and the QA task in this paper. In the future, we plan to investigate the effect of this method in other natural language processing tasks such as reading comprehension. The execution efficiency and effectiveness of the method on a dataset of a longer text are will also be tested in the future.

ACKNOWLEDGMENT

The authors also acknowledge the National Natural Science Foundation of China under Grants No. 62171043, Natural Science Foundation of Beijing under Grants No.4212020, Defense-related Science and Technology Key Lab Fund project under Grants No.6412006200404, National Language Commission project under Grants No. ZDI145-10, YB145-3, Central Leading Local Project "Fujian Mental Health Human-Computer Interaction Technology Research Center", No. 2020L3024, R&D Program of Beijing Municipal Education Commission under grant No.KM202111232001.

COMPETING INTERESTS

All authors of this article declare there are no competing interests.

FUNDING AGENCY

The research is funded by the National Natural Science Foundation of China under Grants No. 62171043, Natural Science Foundation of Beijing under Grants No.4212020, Defense-related Science and Technology Key Lab Fund project under Grants No.6412006200404, National Language Commission project under Grants No. ZDI145-10, YB145-3,Central Leading Local Project "Fujian Mental Health Human-Computer Interaction Technology Research Center", No. 2020L3024, R&D Program of Beijing Municipal Education Commission under grant No.KM202111232001.

REFERENCES

Abreu, J., Fred, L., Macêdo, D., & Zanchettin, C. (2019). Hierarchical attentional hybrid neural networks for document classification. In *International Conference on Artificial Neural Networks* (pp. 396-402). Springer. doi:10.1007/978-3-030-30493-5_39

Adhikari, A., Ram, A., Tang, R., & Lin, J. (2019). *Docbert: Bert for document classification*. https://www.arxiv-vanity.com/papers/1904.08398/

Beltagy, I., Peters, M. E., & Cohan, A. (2020). *Longformer: The long-document transformer*. https://arxiv.org/abs/2004.05150

Che, L., Yang, X., Wang, L., Liang, T., Han, Z., & Information, S. O. (2019). Text structure oriented hybrid hierarchical attention networks for topic classification. *Journal of Chinese Information Processing*, *33*, 93–102.

Chunping, C., & Ting, W. (2019). Research on LSTM semantic correlation long text filtering based on subject dependence. *Computer Technology and Development*, (11), 1–6.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q., & Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 2978-2988). ACL Press. doi:10.18653/v1/P19-1285

Kenton, J. D. M. W. C., & Toutanova, L. K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT* (pp. 4171-4186). NAACL.

Kiesel, J., Mestre, M., Shukla, R., Vincent, E., Adineh, P., Corney, D., & Potthast, M. (2019). Semeval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 829-839). Academic Press. doi:10.18653/v1/S19-2145

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., & Stoyanov, V. et al. (2019). Roberta: A robustly optimized bert pretraining approach. https://arxiv.org/abs/1907.11692 doi:10.18653/v1/S19-2145

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 142-150). ACL Press.

Pappagari, R., Zelasko, P., Villalba, J., Carmiel, Y., & Dehak, N. (2019). Hierarchical transformers for long document classification. In 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) (pp. 838-844). IEEE. doi:10.1109/ASRU46091.2019.9003958

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 6000-6010). Academic Press.

Wang, K., Zheng, Y., Fang, S., & Liu, S. (2020). Long text aspect-level sentiment analysis based on text filtering and improved BERT. *Jisuanji Yingyong*, *40*(10), 2838.

Welbl, J., Stenetorp, P., & Riedel, S. (2018). Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, *6*, 287–302. doi:10.1162/tacl_a_00021

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: generalized autoregressive pretraining for language understanding. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (pp. 5753-5763). Academic Press.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1480-1489). ACL Press.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., & Ahmed, A. (2020). Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, *33*, 17283–17297.

Xueqiang Lv is a professor in Beijing Information Science & Technology University. Before joining in Beijing Information & Technology University, he is a post-doctoral of Peking University from 2003 to 2005. Before as a post-doctoral, he is a PhD candidate in Northeastern University from 1998 to 2003. He received his PhD degree in 2003. Until now, he has been in charge of the National Nature Science Foundation of China three times. He has authored about 60 papers in the international conference or journals, most of them are indexed by EI or SCI database. His current research areas include Cloud Computing, Distributed Computing, Natural Language Processing, Image Processing, Information retrieval, Machine Learning, Deep Learning, etc.

Zhaonan Liu received the B.S. degree in Nanjing University of Science and Technology (NJUST), Nanjing, China in 2019. He is currently working toward the M.S. degree in Beijing Information Science and Technology University (BISTU), Beijing, China. His main research interests include Natural Language Processing, Infomation retrieval, Machine Learning and Deep Learning.

Ying Zhao received the B.S. degree in information and computing Science from Northeastern University at Qinhuangdao (NEUQ), Qinhuangdao, China in 2020. She is currently working toward the M.S. degree in computer technology at Beijing Information Science and Technology University (BISTU), Beijing, China. Her main research interests include natural language processing and patent knowledge mining.

Ge Xu is currently a professor and a vice dean in the College of Computer and Control Engineering at Minjiang University. He received the B.S. degree in Hohai University in 1998 and M.S. degrees in Fuzhou university in 2002. He received the Ph.D. degree in School of Electronics Engineering and Computer Science from Peking University in 2012. During 2011-2012, he was a visiting student in Hong Kong Polytechnic University. His research interests include natural language processing, sentiment analysis, dialog system, machine learning, semantic similarity computation, as well as the construction and applications of various language resources.

Xindong You is corresponding author. She is an associated professor in Beijing Information Science & Technology University. Before joining in Beijing Information Science & Technology University, she is a post-doctoral of Beijing Institute of Graphic Communication union with Tsinghua University from 2016 to 2018. Before as a post-doctoral, she is an associate professor in Hangzhou Dianzi University from 2007 to 2016. Before joining Hangzhou Dianzi University, she was a PhD candidate in Northeastern University from 2002 to 2007. She received her PhD degree in 2007. Her current research areas include Natural Language Processing, Image Processing, distributed computing, Cloud Storage, Energy Management, Data Replica Management, etc.