Intelligent Judicial Research Based on BERT Sentence Embedding and Multi-Level Attention CNNs

Bin Yang[a], Dakui Li[b,*], Nanhai Yang[c]

School of Software Technology, Dalian University of Technology, Dalian, 116620, China

[a]binary_yang@mail.dlut.edu.cn, [b]ldk@dlut.edu.cn, [c]nanhai@dlut.edu.cn

*Corresponding author

Keywords: multi-label text classifications,Intelligent Justice, BERT Sentence Embedding, Attention, CNN

Abstract: The multi-label text classifications of accusations and relevant law articles are important tasks in the construction of intelligent justice. In this paper, we apply multi-level attention mechanisms to the multi-core CNN, and combine the BERT sentence embedding to propose the BERT-ACNN for the tasks. The architecture can selectively extract features and incorporate features extracted by the BERT pre-training language model. Experiments show that our model can achieve better results on the CAIL2018-Small dataset than Average Pooling models, RNNs and CNN. Finally, we improve the performance of BERT-ACNN by oversampling and increasing the number of convolution layers.

1. Introduction

Nowadays, the construction of intelligent justice is in full swing. At the same time, courts accumulated a large number of judgment documents in the long-term handling of cases, providing big data support for intelligent judicial research. In the construction of intelligent justice, accusations prediction and relevant law articles prediction are very practical tasks, and are of great significance for the establishment of auxiliary sentencing systems. In this paper, the accusations and the relevant law articles prediction are regarded as multi-label text classification tasks. There are 202 accusations and 183 relevant law articles labels in our experimental dataset, and the distribution of data is extremely unbalanced. Numerous labels and serious data imbalance issue make multi-label classification tasks very difficult. In the field of natural language processing (NLP), some researchers applied attention mechanism to RNN and achieved good results. However, RNN is very time consuming because it is difficult to train in parallel. In this paper, we apply multi-level attention mechanisms to the multi-core CNN, and combine the BERT sentence embedding to propose the BERT-ACNN for the multi-label text classifications of accusations and relevant law articles. To begin with, we apply attention mechanism to CNN to obtain three ACNNs, which can selectively extract features. Compared to CNN, three ACNNs perform better on the two multi-label classification tasks. Besides, we employ three average pooling models, including the Word2vec Word Embedding Average Pooling model, the BERT Word Embedding Average Pooling model, and the BERT-word2vec Word Embedding Average Pooling model. We concatenate the BERT sentence embedding and the word2vec sentence embedding to get a new sentence embedding and utilize it for classification. Compared with the two average pooling models of single word embedding, the hybrid BERT-word2vec Word Embedding Average Pooling model performs better on the two tasks. In addition, we propose three BERT-ACNNs by combining BERT sentence embedding with three ACNNs. Compared with the BERT Word Embedding Average Pooling model and the ACNN, the corresponding BERT-ACNN performs better on the two tasks. Finally, we utilize some methods to improve the performance of BERT-ACNN. On the one hand, we oversample the train data to ease the problem of data distribution imbalance. On the other hand, we utilize two layers of convolution to extract features.

2. Related Works

Early intelligent judicial research relied mainly on mathematical statistics and traditional machine learning methods. But these methods do not have strong generalization capabilities and are costly. With the development of deep learning, many researchers began to apply deep learning methods in intelligent judicial research. Luo et al. proposed an attention-based model for charge prediction with weighted relevant articles as legal basis [1]. Shubhashri et al. proposed a legal chat robot LAWBO using techniques such as dynamic memory network (DMN) and Glove word vector [2]. In the field of text classification, Kim first applied convolutional neural networks to sentence classification tasks and achieved good results [3]. Hochreiter et al. proposed Long Short-Term Memory (LSTM) to avoid the problem of long-term dependencies in traditional RNN [4]. Cho et al. proposed the Gated Recurrent Unit (GRU), a simplified version of the LSTM, maintains the effect of LSTM and reduces training time [5]. In practice, bidirectional RNNs are more commonly used than unidirectional RNNs and can acquire features in both directions. Recently, attention mechanism has begun to shine in the field of deep learning. Bahdanau first utilized attention mechanism in the field of NLP and applied it to machine translation tasks [6]. Then attention-based RNNs became popular. Luong et al. proposed a global attention mechanism and a local attention mechanism in RNN [7]. Yang et al. utilized BiGRU as encoder and applied attention mechanism on both the word and sentence-level [8]. In the field of NLP, CNN can also be combined with attention mechanism. Yin et al. applied attention mechanism to Basic Bi-CNN to propose three ABCNNs for sentence pair modeling tasks [9]. This can be seen as the earliest work of applying attention mechanism to CNN in the field of NLP. Wang et al. applied attention mechanism to CNN for relation classification tasks [10]. In October 2018, Google launched BERT pre-training language model, which achieved state-of-the-art results in 11 NLP tasks [11]. However, the BERT model requires a large amount of computing resources. Xiao developed the bert-as-service tool, which can apply the BERT pre-training language model as a sentence encoder and mapped a variable-length sentence to a fixed-length embedding for downstream tasks, greatly reducing computing resources and time [12]. In our paper, we propose the BERT-ACNN by combining BERT sentence embedding with attention-based CNN for multi-label text classifications of accusations and relevant law articles and achieve good results.

3. Methodology

3.1 Acnns

The structures of our ACNNs are shown in Figure 1. The ACNN of this paper consists of three types, including ACNN1, ACNN2, and ACNN3. The ACNN1 adds an attention layer after the word embedding layer. The ACNN2 adds an attention layer after the convolutional layer. The ACNN3 adds an attention layer separately after the word embedding layer and the convolution layer.

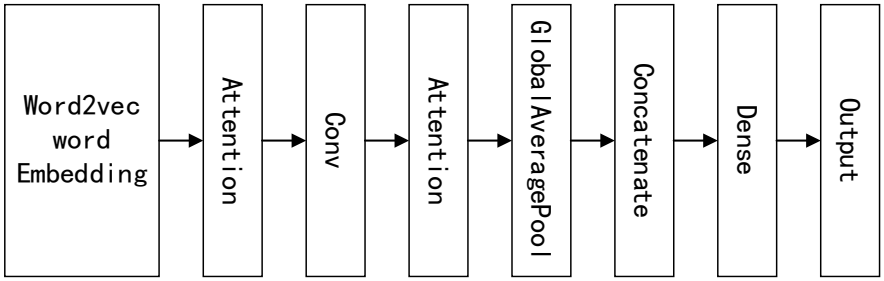

Figure 1. ACNNs

We set the filter widths to (1, 2, 3, 4, 5), the number of filters to 128, and the number of neurons in the penultimate fully connected layer to 1024. And we employ the GlobalMaxPooling method for pooling.

For the three ACNNs, the attention layer is added after different sizes of filters. The attention layer can selectively extract features. The implementation of the attention layer in this paper mainly includes two steps: weight calculation and weighting operation.

3.1.1 Weight Calculation

The input features first pass through a fully connected layer with one neuron. Then the output of the fully connected layer calculates the attention weight vector by the softmax activation function.

3.1.2 Weighted Operation

After repeating attention weight vector to match the dimensions of input, we multiply the attention weight matrix of the new dimension by the input to get the weighted input.

3.2 Word Embedding Average Pooling

3.2.1 Word2vec Word Embedding Average Pooling

We apply the average pooling operation to the word2vec pre-training word embedding with 512 dimensions to get the word2vec sentence embedding for classification.

3.2.2 BERT Word Embedding Average Pooling

First, we utilize the bert-as-service to extract the BERT word embedding from the Chinese version of the BERT-Base pre-training language model. Then we apply the average pooling operation to the BERT word embedding to get the BERT sentence embedding. Since the resulting sentence embedding is of shape (1,768), we need to add a Flatten layer after the sentence embedding for classification.

3.2.3 BERT-word2vec Word Embedding Average Pooling

We concatenate the word2vec sentence embedding and the BERT sentence embedding to generate a new sentence embedding of 1280 dimensions. Then we utilize the new sentence embedding for classification.

3.3 BERT-ACNNs

The structures of our BERT-ACNNs are shown in Figure 2. In this paper, we concatenate the BERT sentence embedding and the output of the three ACNNs to propose the following three BERT-ACNNs.

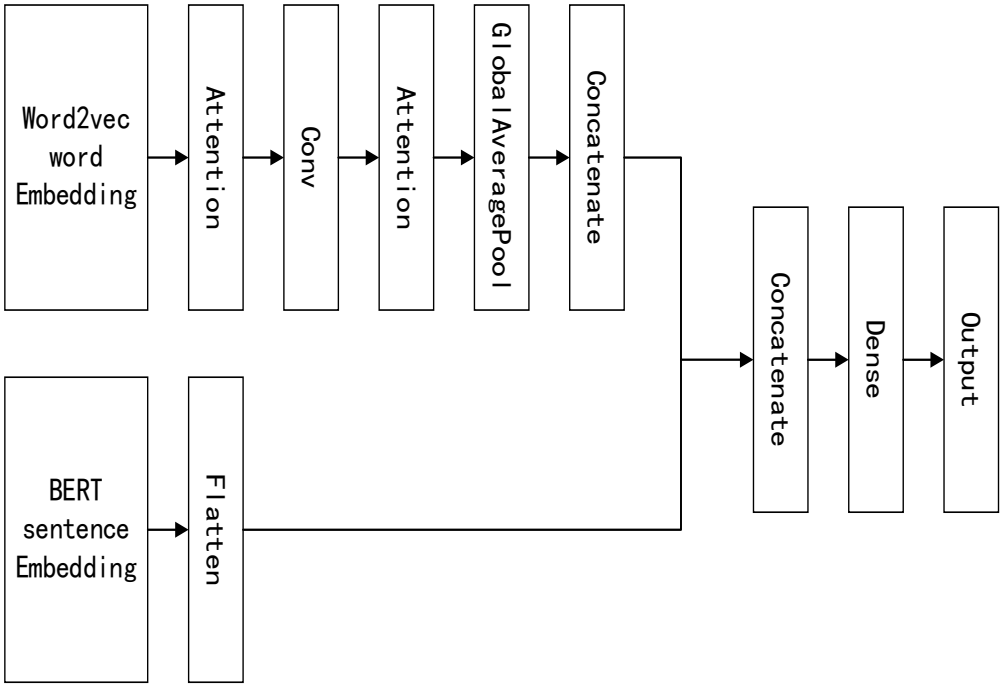

Figure 2. BERT-ACNNs

4. Experiment

4.1 Datasets and Evaluation Metrics

We utilize the CAIL2018-Small dataset in our experiment [13]. All data in the dataset are all collected from China Judgments Online. The dataset includes 154,592 train data, 17,131 valid data, and 32,508 test data. And the dataset includes a total of 202 accusations and 183 relevant law articles. Each piece of data in the dataset includes fact, criminal, accusations, relevant law articles, and so on. Each case may involve multiple accusations and multiple relevant law articles. There is only one criminal in each case of the dataset.

In this paper, we utilize the micro-average F1 value (F1micro) and the macro-average F1 value (F1macro) as the evaluation metrics. We utilize 100 times the average of F1micro and F1macro as the score.

We calculate the score for each task as follows:

$$\text{score} = \frac{F1_{micro} + F1_{macro}}{2} * 100$$

4.2 Data Preprocessing

In this paper, we predict accusations and relevant law articles based on case facts. Our preprocessing operations include extracting the fact from the original text, adding a custom dictionary, utilizing the jieba tool for word segmentation, deleting common stop words, constructing a fact dictionary, serializing the text, and processing the sequence to the same length.

4.3 Baselines

We choose five models as baselines for comparison, including CNN, LSTM, GRU, BiLSTM, and BiGRU. There are 512 neurons in the recurrent layer of LSTM and GRU.

4.4 Implementation Details

For the input of the model, we set the length of the sequence to 400, and we utilize the CBOW model to train word embedding with 512 dimensions [14]. The parameters of the word2vec word embedding are not fine-tuned in training. For the output of the model, we utilize the sigmod activation function for classification.

For training, we utilize Adam as the optimizer [15]. We choose binary crossentropy as the loss function. We set the initial learning rate to 0.001. When the loss in the valid dataset does not decrease within 5 epochs, we halve the current learning rate. We utilize the score in the valid dataset as the stop training indicator and utilize the early stopping method to determine when to stop training. We set the batch size to 512 for three Average Pooling models and set the batch size to 128 for other models. For all models, we add the dropout layer with 0.5 dropout rate before the output layer. For all CNNs, we also add the dropout layer with 0.5 dropout rate before the convolutional layer of and add the dropout layer with 0.2 dropout rate before the penultimate fully connected layer. For all RNNs, we also add the dropout layer with 0.5 dropout rate before the recurrent layer. For all models, we add the batchnorm layer before the penultimate fully connected layer [16]. For all CNNs, we also add the batchnorm layer after the convolutional layer.

Since the single-label data in the CAIL2018-Small dataset accounts for a large proportion, we first classify the predicted probabilities array of a piece of data by multi-label classification method to obtain the prediction label. If the prediction label is an all-zero array, we regard the category with the highest probability in the predicted probabilities array as the prediction label.

4.5 Results and Analysis

For the multi-label text classifications of accusations and relevant law articles, the F1micro, F1macro and score of all models on the test dataset are shown in Table I.

From TABLE I, we can first find that the BERT-ACNN2 has the best scores on the two tasks. And we can see that BERT-ACNN2 has better scores than BERT-ACNN1 on the two tasks. We think this is because the convolutional features are more representative of semantic information. In addition, the scores of BERT-ACNN3 on the two tasks are not the highest among the three BERT-ACNNs. We think this is because BERT-ACNN3 throws away too many features.

And we can see that the three ACNNs have better scores on the two tasks than the CNN. This shows that applying attention mechanism to the CNN can selectively extract features and avoid noise. It can be also seen that the BERT-word2vec Word Embedding Average Pooling has better scores on the two tasks among the three Average Pooling models. This shows that model performance can improve by combining different word embedding features. We also find that the three BERT-ACNNs have better scores on the two tasks than the BERT Word Embedding Average Pooling model and the corresponding ACNN. This shows that three ACNNs can improve performance by combining the BERT sentence embedding. In addition, we find that the CNN has better scores on the two tasks than the four RNNs. This shows that the judgments of the accusations and relevant law articles in crime cases rely more on keywords, while the RNNs focusing on the contextual information have no advantage here.

Table 1. Model Comparsion

Method		Accusations			Relevant Law Articles		
		F1micro	*F1macro*	*score*	*F1micro*	*F1macro*	*score*
Baselines	LSTM	0.8606	0.7261	79.34	0.8247	0.6551	73.99
	GRU	0.8640	0.7493	80.67	0.8355	0.6853	76.04
	BiLSTM	0.8663	0.7595	81.29	0.8352	0.6866	76.09
	BiGRU	0.8688	0.7532	81.10	0.8365	0.6896	76.31
	CNN	0.8677	0.7600	81.39	0.8400	0.7139	77.70
ACNN1		0.8726	0.7744	82.35	0.8409	0.7185	77.97
ACNN2		0.8734	0.7801	82.68	0.8454	0.7333	78.94
ACNN3		0.8721	0.7752	82.37	0.8424	0.7126	77.75
word2vec Word Embedding Average Pooling		0.8271	0.6941	76.06	0.8076	0.6593	73.35
BERT Word Embedding Average Pooling		0.8267	0.7029	76.48	0.8064	0.6617	73.41
BERT-word2vec Word Embedding Average Pooling		0.8481	0.7318	79.00	0.8288	0.6880	75.84
BERT-ACNN1		0.8755	0.7766	82.61	0.8490	0.7409	79.50
BERT-ACNN2		0.8784	0.7851	83.1	0.8503	0.7403	79.5

Method	Accusations			Relevant Law Articles		
	F1micro	*F1macro*	*score*	*F1micro*	*F1macro*	*score*
			8			3
BERT-ACNN3	0.8743	0.7781	82.62	0.8476	0.7371	79.24

4.6 Model Performance Improvement Research

For the BERT-ACNN2 with the highest scores on the two tasks in Table I, we improved its performance by oversampling and increasing the number of convolution layers.

4.6.1 Oversampling

In the train dataset, the data distribution is extremely unbalanced. Since the single-label cases account for a large proportion in the train dataset, we can get a better effect at a lower cost by only oversampling the single-label cases. We utilize the oversampling method of directly copying small categories of data to alleviate the problem of data imbalance. For the cases with single accusation label in the train dataset, we expand no more than 400 times, and the expanded data volume does not exceed 800. After oversampling, the amount of data in the train dataset reaches 226,733. For the cases with single relevant law article label in the train dataset, we expand no more than 800 times, and the expanded data volume does not exceed 800. After oversampling, the amount of data in the train dataset reaches 225,078.

4.6.2 Increase the number of convolution layers

We increase the number of convolution layers of the BERT-ACNN2 model to 2.

Table II shows the effect of oversampling and increasing the number of convolution layers on BERT-ACNN2, where one-conv represents a layer of convolution and two-conv represents two layers of convolution.

Table 2. Model Performance Improvement

Method	Accusations			Relevant Law Articles		
	F1micro	*F1macro*	*score*	*F1micro*	*F1macro*	*score*
no oversampling one-conv	0.8784	0.7851	83.18	0.8503	0.7403	79.53
oversampling one-conv	0.8782	0.7994	83.88	0.8491	0.7568	80.30
oversampling two-conv	0.8854	0.8107	84.81	0.8510	0.7617	80.64

From Table II, we can see that by oversampling and increasing the number of convolution layers, BERT-ACNN2 can improve the scores of the multi-label text classifications of accusations and relevant law articles.

5. Conclusion

First of all, we apply attention mechanism to the multi-core CNN model to obtain three ACNNs that can selectively extract features, which outperform CNN in the multi-label text classifications of accusations and relevant law articles. Moreover, we employ three average pooling models in the two tasks, including the Word2vec Word Embedding Average Pooling model, the BERT Word Embedding Average Pooling model, and the BERT-word2vec Word Embedding Average Pooling model. We find that the BERT-word2vec Word Embedding Average Pooling model performs better by combining different word embedding. Furthermore, we combine the BERT sentence embedding with the three ACNNs to propose three BERT-ACNNs with better performance. At last, we improve the performance of BERT-ACNN by oversampling and increasing the number of convolution layers. In the future, we will solve the problem of data distribution imbalance by adjusting the loss function

and other methods, and carry out more in-depth research on the application of attention mechanism in the field of NLP.

References

[1] Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. Learning to predict charges for criminal cases with legal basis [C]. Copenhagen: In Proceedings of EMNLP, 2017.

[2] G Shubhashri, N Unnamalai, and G Kamalika. LAWBO: A Smart Lawyer Chatbot [C]. Goa, India: In Proceedings of The ACM India Joint International Conference on Data Science & Management of Data, 2018.

[3] Yoon Kim. Convolutional neural networks for sentence classification [C]. Doha: In Proceedings of EMNLP, 2014.

[4] Hochreiter, Sepp and Schmidhuber, Jürgen. Long Short-Term Memory [J]. Neural Computation, vol. 9, no. 8, pp. 1735 - 1780, 1997.

[5] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, and Fethi Bougares, Holger Schwenk, et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation [C]. Doha: In Proceedings of EMNLP, 2014.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate [C]. San Diego: In Proceedings of ICLR, 2015.

[7] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation [C]. Lisbon, Portugal: In Proceedings of EMNLP, 2015.

[8] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola1, and Eduard Hovy. Hierarchical attention networks for document classification[C]. San Diego: In Proceedings of NAACL, 2016.

[9] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs [J]. Transactions of the Association for Computational Linguistics, vol. 4, pp. 259 - 272, 2016.

[10] Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. Relation Classification via Multi-Level Attention CNNs [C]. Berlin, Germany: In Proceedings of ACL, 2016.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [C]. Minneapolis, United States: In Proceedings of NAACL, 2019. in press.

[12] Xiao Han. bert-as-service [OL]. [2018,12,16]. http://github.com/hanxiao/bert-as-service.

[13] Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, et al. CAIL2018: A Large-Scale Legal Dataset for Judgment Prediction[OL]. arXiv preprint arXiv:1807.02478, 2018. unpublished.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality [C]. Harrahs and Harveys, Lake Tahoe: In Proceedings of NIPS, 2013.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization [C]. San Diego: In Proceedings of ICLR, 2015.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]. Lille, France: In Proceedings of ICML, 2015.