Multi-label Classification of Chinese Judicial Documents based on BERT

Mian Dai
Department of Computer Science
National Chengchi University,Taiwan
Taipei,Taiwan
mian.dai.work@gmail.com

Chao-Lin Liu
Department of Computer Science
National Chengchi University,Taiwan
Taipei,Taiwan
chaolin@cs.nccu.edu.tw

***Abstract*—Judicial decisions are an important part of modern democratic societies. In this paper, I present results of multi-label classification of Chinese judicial documents. The experiments employ the same corpus that was used in Chinese AI & Law Challenge(CAIL) 2018.**

Keywords—Natural language processing, Multi-label classification, BERT

I. INTRODUCTION

In this paper, I describe my work on a 2018 CAIL classification task(http://cail.cipsc.org.cn:2018/), which is about making multi-label accusation predictions of judicial documents. A legal case may have multiple labels because a sequence of criminal activities may violate the law in various ways. It is typical to use a label to indicate one specific type of violation for the legal cases, so we have a multi-label classification task.

Back in 2018, the competition organizer released a baseline adopting term frequency–inverse document frequency(TFIDF) method to vectorize the legel cases and Linear Support Vector Classification(LinearSVC) to make classifications. Besides, in the training set, only the first accusation corresponding to each legal case was chosen for single label prediction in the test set.

My exploration is based on the Bidirectional Encoder Representations from Transformers(BERT)[1]. Just after fine-tuning, BERT obtains a score of 0.773 in the test set after averaging micro-f1 and macro-f1 score.

To gain a more comprehensive feature vector of the judicial documents, I refer to some ideas from team AlphaCourt in CAIL 2019[2] about concatenating feature vectors including BERT output [CLS] vector, vectors of BERT hidden layers and vectors obtained by TFIDF method .

As for vectors obtained by TFIDF method, first, I apply the TFIDF method to vectorize the law cases after word segmentation. The second experiment is to obtain the vector by retaining the word segments which related to legal field after employing word segmentation on the legal documents and adopting TFIDF method.

As each BERT layer's capability of encoding linguistic features varies[3], I experiment with different combinations of the vectors of some specific hidden layers and the BERT output [CLS] vector for further classification.

II. RELATED WORK

The classification problem has been extensively studied in the field of natural language processing(NLP). The traditional method is to vectorize the text first and then to make classifications with a classifier. There are many ways of text feature extraction, such as TFIDF, N-gram counts and sentence length counts. The wildly-used classifiers includes Naive Bayes Classifier, Support Vector Machine Classifier and Decision tree.

With the popularity of deep learning, some neural network models that achieve good results have begun to attract attention, such as Convolutional Neural Network (CNN). In addition, the position of a word in the text is very important, because the meaning of a word is closely related to its context. Thus, Recurrent Neural Network(RNN) addresses this linguistic feature and is able to distinguish different meanings that the word may have in different contexts. And the Long Short-Term Memory(LSTM) can solve the gradient problem which may occur during training phase of previous RNN.

In recent years, models based on the attention mechanism outperforms[4]. Attention mechanism is about to find relationships between word embeddings in parallel with relative position information. After some fine-tuning for specific tasks, including question answering, sentence classification and sentence tagging, BERT model can obtain great results.

III. EXPERIEMNTS

A. Dataset

The dataset of judicial documents which the competition provides is built based on criminal judicial documents published by the Supreme People's Court of China. Each piece of data is composed of the fact of the legal case, relevant provisions, the accusations of the defendant and the length of the sentence. There are two datasets called CAIL2018-Small and CAIL2018-Large respectively.

The dataset used in this paper is CAIL2018-Small. It involves a total of 202 accusations. The average text length of the case fact is 478.88 characters and the average accusations involved is 1.21 per case. I take 85% of the 150,000 training data as the training set, the rest as the validation set and use the given 30,000 test data. The case facts are used as the feature and accusations as the label.

B. Approaches

1) Model

I use PyTorch's implementation of BERT-base-chinese and set the max input length as 512, batch size as 10, training epochs as 4 and learning rate as 2e-5.

For this specific task, I adopt BertForSequenceClassification class and slightly modify it to get 202 logits (as the total number of accusations is 202), and then use sigmoid function to get the possibility of each accusation while evaluating and testing.

In the phase of evaluating model with validation set, each probability ranging from 0.1 to 0.8 is selected as the threshold for classifying as a crime, and the corresponding average score of micro-f1 and macro-f1 is calculated. I pick the possibility which reaches the highest score as the threshold for the testing stage.

2) Embeddings For Tthe Classifier Input

- Method 1:BertModel [CLS] output embedding
- Method 2:Employ word segmentation on the judicial documents with the tool of THULAC[5], use TFIDF method while setting ngrams as 1 and 2, max features as 5000 and then use Principal Component Analysis(PCA) to get a 50-dimensional vector. Concatenate the vector with BertModel [CLS] output embedding.
- Method 3:Obtain a law-related word list from Tsinghua Open Chinese Lexicon(THUOCL)[6]. Employ word segmentation with the tool of THULAC on both law-related word list and judicial documents. Then filter the words of the judicial documents while only retaining the words which are also in the law-related word list on which is after employed word segmentation. Then use TFIDF method while setting ngrams as 1 and 2, max features as 5000 and use PCA to get a 50-dimensional feature vector. Concatenate the vector with BertModel [CLS] output embedding.
- Method 4:Concatenate the vector of the last hidden layer with index 0 and BertModel [CLS] output embedding.
- Method 5:Concatenate the vectors of the last 4 hidden layers with index 0.

C. Result

	Micro f1	Macro f1	Average f1 (mirco-f1+macro-f1)/2
Method 1	***0.845***	***0.700***	***0.773***
Method 2	***0.865***	***0.705***	***0.785***
Method 3	***0.865***	***0.721***	***0.793***
Method 4	***0.881***	***0.768***	***0.825***
Method 5	***0.876***	***0.782***	***0.829***

Compared to the original classification using BERT [CLS] output, adding more feature information can help improve the f1 score. As the results show, the vectors of the BERT hidden layers provide more effective information than the vectors obtained by the TFIDF method. Combining BERT [CLS] output with the first vector of the last hidden layer or using only the vectors of the last 4 hidden layers achieves about 5% higher average f1 score than the original method 1.

However, the limitation of using BERT is that BERT can only be trained with sequence lengths up to 512. The approach in this paper is to use the first 512 words. For longer articles, further improvements are needed in order to vectorize the full text, such as cutting text into multiple 512 words to get multiple vectors and averaging the vectors.

IV. Conclusion and Future work

In this paper, I make modifications to BertForSequenceClassification class to make multi-label classifications and adopt different embeddings to improve the model performance. BERT shows its powerful ability on specific tasks by fine-tuning. According to the experiment results, a more comprehensive feature vector for classification may contribute to better model performance.

For future work, I would try to vectorize the full text which is longer than 512 by BERT and train other neural network models with the knowledge learned by BERT as input.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre- training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[2] AlphaCourt (2019) CAIL2019 [Source code]. https://github.com/GuidoPaul/CAIL2019

[3] G. Jawahar, B. Sagot, D. Seddah, "What does BERT learn about the structure of language?" in Proceedings of the 57th Conference of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3651–3657.

[4] A.Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need, " in Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[5] M. Sun, X. Chen, K. Zhang, Z. Guo, Z. Liu. THULAC: An Efficient Lexical Analyzer for Chinese. 2016.

[6] S. Han, Y. Zhang, Y. Ma, C. Tu, Z. Guo, Z. Liu, M. Sun. THUOCL: Tsinghua Open Chinese Lexicon. 2016.