Legal text classification model based on text statistical features and deep semantic features

Jiaming Gao[a], Hui Ning[a], Zhongyuan Han*[b], LeiLei Kong[b] and Haoliang Qi[b]

[a]*Harbin Engineering University Harbin, China*
[b]*Foshan University, Foshan, China*

Abstract

In this paper, we propose a legal text classification model based on text statistical features and deep semantic features, which is mainly used to solve the legal text classification task of FIRE2020(Forum for Information Retrieval Evaluation)@AILA. We use the TF-IDF feature of text as the statistical feature of the text, the [CLS] token information output by the BERT model as the deep semantic feature of the text, and the combination of the two as the joint feature of the text. Joint features are used to train classifiers such as Logistic Regression and SVM. Compared with the deep learning method and TF-IDF based method, the method using joint features has a greater performance improvement. The F1 score of this method on the test set reaches 0.457, which is the second in all the submitted teams.

Keywords

joint text features, BERT, deep semantic features, tf-idf, logistic regression, Adaboost, SVM

1. Introduction

Legal case documents have a set of common writing structures, such as: ”Facts of the case”, ”Issues being discussed”, ”Arguments given by the parties”. Specific parts play a specific role in legal case documents. Distinguishing these components of a document can not only help us improve the readability of the document, but also help us to complete other natural language processing tasks, such as semantic similarity calculation, text summarization, etc. This text classification task[1] is to identify the various components of legal documents, including seven categories: Facts, Ruling by Lower Court, Argument, Statute, Precedent, Ratio of the decision, Ruling by Present Court. The evaluation released 60 legal case documents. The training set includes 50 legal documents, including 9380 training data. The test set includes 10 legal documents, including 1905 test data.

2. Our Approach

The overall process of our method is shown in Figure 1. As shown in Figure 1, our task is regarded as a multi-classification task. The legal text is converted into statistical features and

Forum for Information Retrieval Evaluation 2020, December 16-20, 2020, Hyderabad, India
EMAIL: gaojiaming24@163.com (J. Gao); ninghui@hrbeu.edu.cn (H. Ning); Hanzhongyuan@gmail.coml (Z. Han*); Kongleilei1979@gmail.com (L. Kong); haoliang.qi@gmail.com (H. Qi)
ORCID: 0000-0001-8960-9872 (Z. Han*)

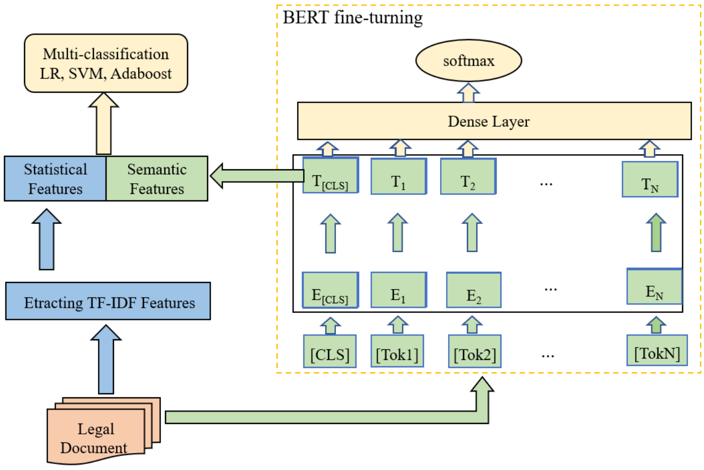

Figure 1: Model structure

semantic features to train a multi-classifier. Among them, the tf-idf vector of the text is used as the statistical feature, and the CLS vector at the output of the BERT[2] is used as the semantic feature of the text. The multi-classification model uses logistic regression[3], support vector machine[4], and Adaboost[5] algorithm to classify and predict legal text. To allow BERT to better represent the semantic features of legal texts, we connect the CLS output of BERT to the neural network of 7 classifications, so that the fine-tuning of BERT can be achieved through legal training data. After the fine-tuning stage of the model, we need to take the legal text as input and get the output of the model's [CLS] token as the deep semantic features of the text.

3. Experimental

3.1. Dataset

The data set provided in this evaluation is divided into training data and test data. The training data set contains 50 files, 9380 pieces of training data, and the test data contain 10 files and 1905 pieces of test data. The training data is classified into seven categories. The proportion of each label is shown in Table 1.

1. **Facts:** sentences that denote the chronology of events that led to filing the case
2. **Ruling by Lower Court:** Ruling by Lower Court: since we will be providing Indian Supreme Court cases, these cases were given a preliminary ruling by the lower courts. These sentences correspond to the ruling/decision given by these lower courts.
3. **Argument:** sentences that denote the arguments of the contending parties
4. **Statute:** relevant statute cited
5. **Precedent:** relevant precedent cited
6. **The ratio of the decision:** sentences that denote the rationale/reasoning given by the Supreme Court for the final judgment

Table 1
proportion of each label

Label category	Train dataset		Test dataset	
Ratio of the decision	3624	38.64%	587	30.81%
Facts	2219	23.66%	403	21.15%
Precedent	1468	15.65%	319	16.75%
Argument	845	9.0%	256	13.44%
Statute	646	6.89 %	167	8.77%
Ruling by Lower Court	316	3.37 %	94	4.93%
Ruling by Present Court	262	2.79 %	79	4.15%
Total	9380	100 %	1905	100%

7. **Ruling by Present Court:** sentences that denote the final decision given by the Supreme Court for that case document

To select the optimal experimental parameters, we divide the training data into a 7:3 ratio: training set (6566 items) and development set (2814 items), and select the final model hyperparameters based on the experimental results on the development set.

3.2. Experimental Setting

3.2.1. Experimental Procedures

Our method is divided into three steps: use the training data to fine-tune the BERT based on the classification task. Use the BERT source code to extract the output results of the last layer of the model. Only the feature vector in the [CLS] token in the model is used as the deep semantic feature of the sentence, so the deep semantic feature of each sentence is a 768-dimensional vector. Next, use the scikit-learn[1] tool to extract the TF-IDF features of the text as the text statistical features. Each input sentence can get the 5814-dimensional tf-idf features. Combine deep semantic features and text statistical features as the final 6582-dimensional joint text feature. Finally, the joint text features are input into a logistic regression, SVM, AdaBoost classification models for final training and prediction.

3.2.2. Parameter Selection

The parameters of the final model are selected according to the performance of each classifier on the development set. The fine-tuning parameters of the BERT model are selected according to the classification accuracy of the BERT on the development set. Since the maximum sentence length of 128 can cover 99.5% of the sentence length, because seq_length is directly set to 128, no parameter adjustment is required. The parameter settings and classification effects of the BERT model are shown in the following Table 2 (the main evaluation metrics is the Accuracy value).

[1] https://scikit-learn.org/

Table 2
BERT fine-tuning experiment results on development set

Number	Seq_length	Batch_size	Learning_rate	Epoch	Loss	Accuracy
1	128	16	2e-5	2	1.0723	0.6325
2	128	32	2e-5	2	1.0572	0.6364
3	128	64	2e-5	2	1.0952	0.6222
4	128	32	1e-5	1	1.1971	0.5856
5	128	32	1e-5	2	1.0884	0.6282
6	128	32	1e-5	4	1.0822	0.6240
7	128	32	1e-5	6	1.1119	0.6218
8	128	32	1e-5	8	1.1913	0.6208
9	128	32	1e-5	10	1.2503	0.6190
10	128	32	2e-5	2	1.0504	0.6396
11	128	32	2e-5	4	1.1200	0.6289
12	128	32	2e-5	6	1.3411	0.6279
13	128	32	2e-5	8	1.5322	0.6265

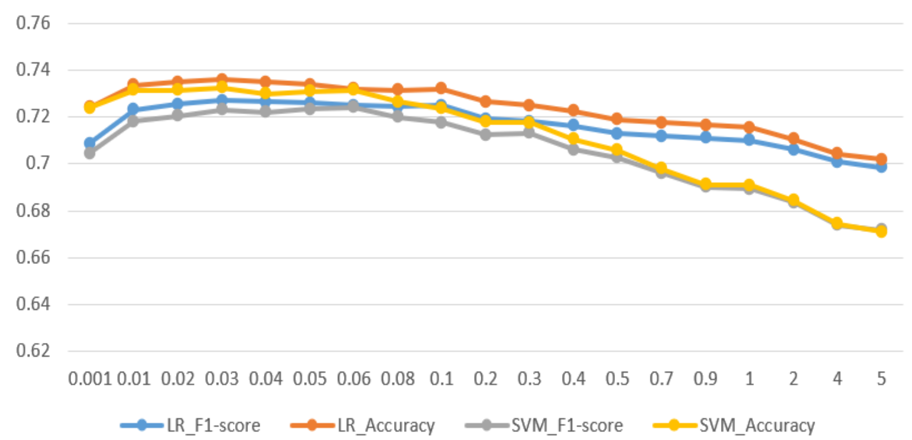

Figure 2: Parameter adjustment experiment of logistic regression and SVM model on development set using joint features

Finally, select the 10th group of experimental parameters as the final BERT fine-tuning parameters, seq_length=128, Batch_size=32, Learning_rate=2e-5, and epoch=2. The main adjustment parameter of logistic regression and SVM is the C value, which is selected according to the F1-score of the two classifiers on the development set. Since both logistic regression and SVM need to set parameter C, because the two tuning experiments are combined and compared, the experimental feature is joint features, and the evaluation indicators are Accuracy and F-score.As shown in Figure 2.The parameter adjusted by Adaboost is the number of weak classifiers in the model n_estimators.As shown in Figure 3.

The final experimental parameter settings are shown in Table 3, logistic regression, SVM classifier only adjusts the value of parameter C, the parameter adjusted by Adaboost is n_estimators

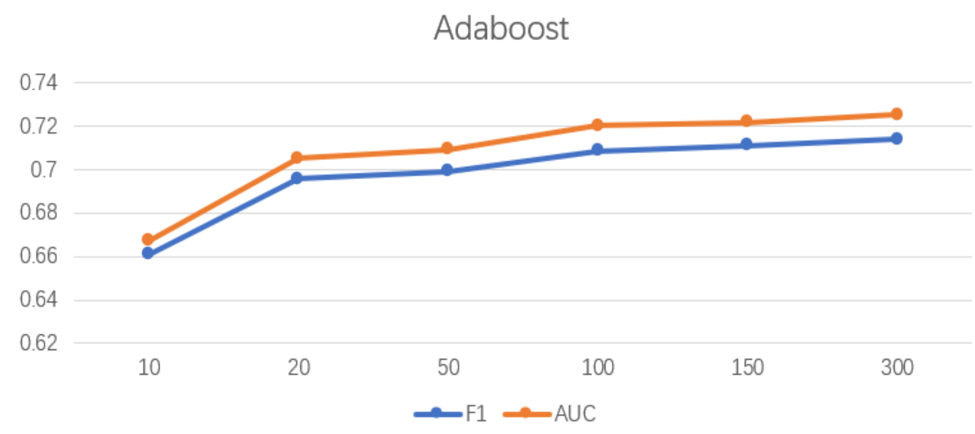

Figure 3: Parameter adjustment experiment of Adaboost model on development set using joint features

Table 3
Model parameter settings

Model	Parameter	F-score	Accuracy
Logistic Regression	C=0.03	0.7269	0.7359
SVM	C=0.03	0.7231	0.7324
Adaboost	n_estimators=300	0.7138	0.7249
BERT	seq_length=128,batch_size=32,lr=2e-5,epoch=2	-	0.6396

Table 4
Final evaluation result on test sets using joint features

Run	Precision	Recall	F-score	Accuracy	Rank
heu_gjm_1(LR)	0.541	0.472	0.457	0.603	2
heu_gjm_2(SVM)	0.526	0.468	0.451	0.598	4
heu_gjm_3(Adaboost)	0.529	0.456	0.444	0.59	5

representing the number of weak classifiers, and other parameters are set to default settings.

3.3. Experimental Result

Our final experimental results on the test set using joint features are shown in Table 4 (sorted by F-score).In the experimental comparison between the Tf-Idf feature and joint feature, the evaluation index is the F-score value. The experimental result is the F-score on the development set.The experimental results are shown in Figure 4.

Comparison of the experimental effects of various classifiers and various feature combinations under different parameters (take the best experimental results) and the improvement is calculated according to the corresponding model's tf-idf feature F-score experimental result.As shown in Table 5.

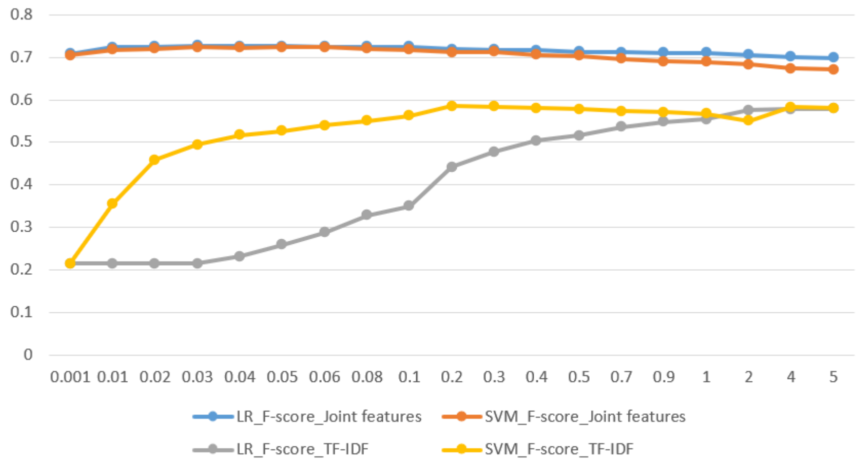

Figure 4: Joint features and TF-IDF features experiment on development set

Table 5
Comparison of experimental results on dev dataset

Model	Parameters	Accuracy	F-score	Increase
LR+Joint features	C=0.03	0.7359	0.7269	26.39%
SVM+Joint features	C=0.03	0.7324	0.7231	24.38%
Adaboost+Joint features	C=300	0.7249	0.7137	77.44%
BERT	batch_size=32,lr=2e-5,epoch=2	0.6396	-	-
LR+tf-idf	C=2	0.5980	0.5751	-
SVM+tf-idf	C=0.2	0.6066	0.5856	-
Adaboost+tf-idf	n_estimators=300	0.4584	0.4022	-

4. Conclusion

In this evaluation, we used joint features combined with logistic regression support vector machines and Adaboost to solve the multi-classification problem in legal texts. We use the training data to fine-tune the BERT model. After fine-tuning, we extract the [CLS] information output by the model as the deep semantic information of the sentence, combining the tf-idf feature of the text as the joint feature of the text. Use logistic regression, support vector machine, and Adaboost classifier for classification and prediction. On the development set, the classification accuracy of the BERT model is 0.63, the optimal accuracy result of the logistic regression classifier using only tf-idf features is 0.59, and the optimal classification accuracy of the joint features is 0.73, which is improved compared to the BERT model 15.9%, compared with the tf-idf feature increased by 23.7%, the performance improvement is obvious.

Acknowledgments

This work is supported by the National Social Science Fund of China (No.18BYY125).

References

[1] P. Bhattacharya, P. Mehta, K. Ghosh, S. Ghosh, A. Pal, A. Bhattacharya, P. Majumder, Overview of the FIRE 2020 AILA track: Artificial Intelligence for Legal Assistance, in: Proceedings of FIRE 2020 - Forum for Information Retrieval Evaluation, 2020.

[2] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding (2018).

[3] D. R. Cox, The regression analysis of binary sequences, Journal of the Royal Statal Society. Series B: Methodological 20 (1958) 215–242.

[4] Schölkopf, Bernhard, Learning with kernels : support vector machines, regularization, optimization, and beyond, MIT Press, 2003.

[5] R. E. Schapire, A brief introduction to boosting, in: Sixteenth International Joint Conference on Artificial Intelligence, 1999.