Language Model-based Approaches for Legal Assistance

Zhiran Li[b], Leilei Kong[a,*]

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

Abstract

This paper mainly introduces our approaches for the task provided by the FIRE2020(forum for information retrieval evaluation). Task 1 has 2 sub-tasks. The first part of it is to match similar legal cases, and the second part is to match legal cases with relevant statues. We tried the language model on the first task. Language models with different hyper-parameters are used to rank the queries. And the second task is handled as a multi-classification task. We applied a pre-training language model(BERT model developed by Google) on it, with different training parameters.

Keywords

Language Model, BERT, Multi-classification, Legal Assistance [1]

1. Introduction

With the accumulation of legal cases and statues, an efficient way of retrieval this massive amount of data has become more and more mandatory. In the recent years, it comes to the people's eyes that artificial intelligence approaches have a great potential on these tasks. In this regard, FIRE2020 proposed the task of legal information retrieval and named it as AILA.

The AILA task1 has 2 subtasks. Subtask A requires us to match legal cases with prior similar ones, and subtask B requires us to match legal queries with relevant statutes. The AILA provides 2 parts of data for task1. The first part of it contains 3000 judgment cases delivered by the Supreme Court which will be used in subtask A, the second part of the data is composed of 197 statues from India Law for subtask B.

The task2 requires us to classify a series of sentences by the following classes:

Fact: sentences that points out when, where and what happened.

Ruling by Lower Court:the rules given by the lower courts before it was sent to the Supreme Court of India.

Argument: Points given by the contending parties.

Statute:relevant statute cited

Precedent:relevant precedent cited

Ratio of the decision and Ruling by Present Court:sentences that denote the rationale/reasoning given by the Supreme Court for the final judgment.

50 documents are given as the train set.

2. Methods

[1] Forum for Information Retrieval Evaluation 2020, December 16–20, 2020, Hyderabad, India
EMAIL:leezhiran@yeah.net (B. 1);
kongleilei@fosu.edu.cn (A. 1)(*corresponding author)
ORCID: 0000-0002-4636-3507 (A. 1)

1)Task 1A(Case Retrieval)

Fig 1 shows the procedure of which the data and queries are processed.

First the data will be pre-processed. In this step, a stop word list will be applied to the raw text to get rid of *punctuation* marks and the words that has so high frequency that it carries less information

Then, we use Lucene to create an index of the given text. We use language model with Jelinek Mercer smoothing to determine the similarity within the index. The λ we use is 0.7. The next step is a query with two steps. First, we use the original queries and store it's results. Then, we apply an IDF screening on the queries. The terms that ranked top 50% are used to form a new query. At Last we accumulate the score of these two queries as the final results.

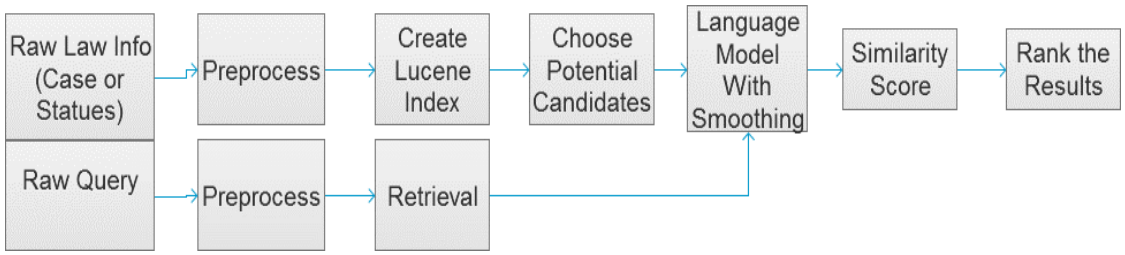

Figure 1. The process procedure of Task1

2) Task 1B(Statute Retrieval)

The preprocessing and creating index process of the task 1b is the same as task 1a.However, in this case, matching the keywords is much more difficult than the task1a since the document is much smaller. At the last ranking part, instead of using the IDF screening, we process the query by converting it into a bag-of-word vector, and calculate the Jaccard similarity of the vectors. Then, rank the results according to the similarity.

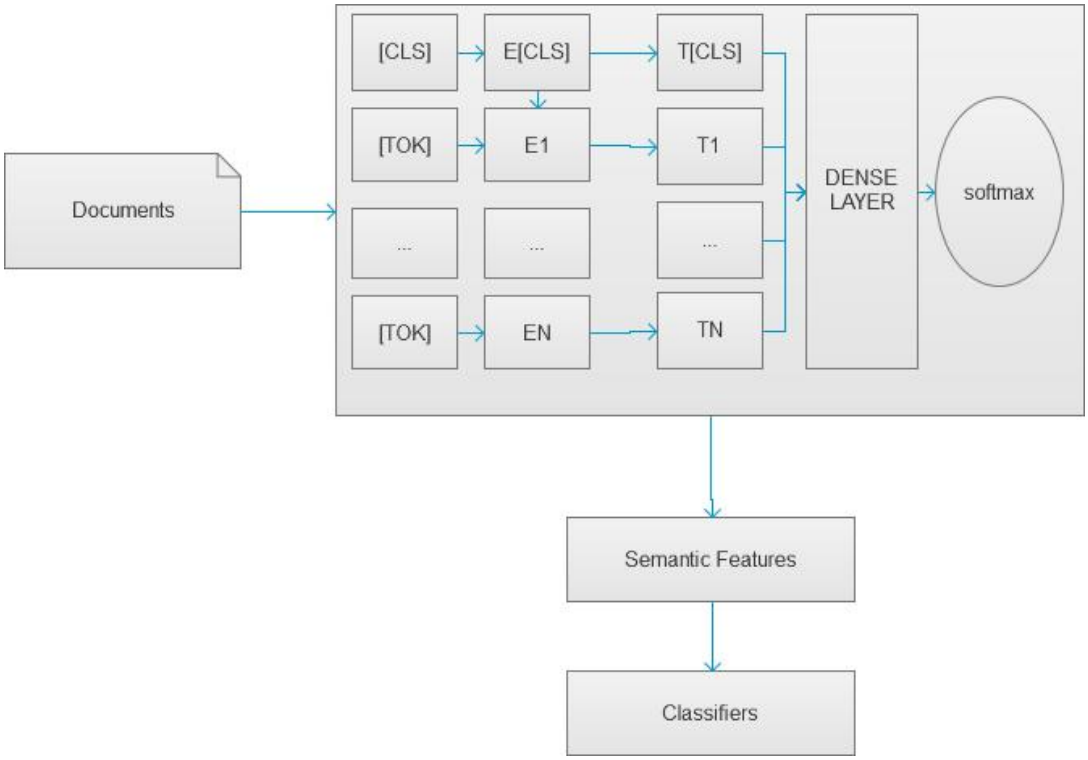

Figure 2. The procedure of task 2

3) Task 2(Rhetorical Role Labeling)

Fig2 shows the process of task2.

The task2 is considered as a multi-classification task. As far as we know the BERT(Bidirectional Encoder Representations from Transformers) model performed the best in this sort of tasks. We chose the BERT-base model pretrained by Google as the base model and finetuned it with the train data.

Different learning rate, epoch , and random seed are tested.When it come to the train data, we randomly chose 80% of the given documents and use them as the train data,10% of the documents are used as validation data and 10% of the document are used as test data. The BERT model results a better result than the other models we tried, such as logistic regression model with the feature of TF-IDF provided by the sklearn python library.

3. Experimental Setups

The settings of task1 is describe as below in Table 1：

Table 1

Task_Category	Parameters
Task1A	Λ =0.7, μ =2000
Task1B	Λ =0.7

Parameter Explanation: Λ is the parameter of the Luence built-in language model with Jelinek Mercer Similarity.It is said that Λ =0.1 is optimal for titles, and Λ =0.7 is optimal for long query. The μ stands for the default parameter in the Lucene language model with Dirichlet smoothing.

The settings of task2 is shown as below in Table2:

Table 2

Runs	Epoch	Learning Rate	Batch_size	Random seed	Bert Base Model
1	2	1e-5	128	1	Bert_Base
2	3	1e-5	128	1	Bert_Base
3	10	1e-5	128	1	Bert_Base
4	5	1e-5	128	1	Bert_Base
5	3	1e-5	128	Random integers	Bert_Base
6	3	1e-6	128	1	Bert_Base
7	3	1e-5	64	1	Bert_Base
8	3	1e-5	128	1	Bert_Base
9	3	1e-5	16	1	Bert_Large
10	3	1e-5	128	1	Bert_small

In the experiment process, different epoch, learning rate and random seed is applied. We discovered that the batch size only affect the memory usage and train time cost.What's more, the prime epoch setting we found is 3.

Beyond that ,we observe a significant over fitting.In other word, when epoch is over 3, the model is recording the data set.Through the loss is dropping continuously, the result on the test set is not improving.Therefore , we use the epoch value of 3.Learning rate will result in longer training time and no significant improvement on result.

For the choice of the BERT pretrained model, a bigger scale pretrained model results in a better result. However, memory usage and processing power consumption. In theory, we will have better result if we run our fine-tune on the bigger scale of BERT model. However, due to limitation of computation power and memory capacity, a rather small BERT model is chosen(In our experiment, the BERT-Base).

The random seed also have an impact on the final result, and rely on the specific test data set. The impact of the random seeds is caused by the dividing process applied to the document. Document

divided in different ways will affect the final fine-tuned model. We submitted two submissions with different random seeds, and it seems to make a dramatic difference.

4. Experimental Results

The results of the submissions Task1 are as following Table3.

Table 3

Results of the AILA Task 1 - Precedent Retrieval

Run_ID	MAP	BEREF	recip_rank	P@10	TASK_CATEGORY
fs_hu_task1a	0.1351	0.0885	0.2041	0.1	Task1A
fs_hu_task1b	0.235	0.198	0.3581	0.08	Task1B

The evaluation results of task 1a submit shows that our method of task 1a is rather efficient, compared to the methods we used before without the last step of processs with IDF feature.

Table 4

Results of the AILA Task 2 - Rhetorical Role Labeling

Run_ID	Macro Precision	Macro Recall	Macro F-Score	Accuracy
fs_hu_1	0.493	0.454	0.428	0.562
fs_hu_2	0.262	0.343	0.266	0.457

Fine tuning on a bigger BERT base model may help improve the performance. Due to limitation of computing power, we are not able to fine-tune the model on a bigger BERT base model. We do observe improvement when we increase the scale of the pre-trained BERT model.

The differences between the two submit is different random seed by which the documents are divided into test sets, validation sets and test sets.

5. Conclusions

This paper describes the methods we used in FIRE2020 and shows the potential improvement approaches. In conclusion IDF feature with language model will result better in long text tasks such as task 1a. In the contrast, tasks with smaller documents will gain more benefits from TF feature.

In the multi-classfication tasks, such as task2, Google Bert out performed many prior models, such as out-of-box logistic regression classifiers. We're considering if we will achieve better results if we had other classifier to cooperate with the BERT model, instead of one.

6. Acknowledgements

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

7. References

[1] Robertson, S., Steve Walker, and H. BM. "GM Okapi at trec-3." Proceedings of the Third Text REtrieval Conference (TREC 1994). 1994.

[2] Zhai, Chengxiang, and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. ACM SIGIR Forum. New York, NY, USA, 2017.

[3] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[4] Bhattacharya, Paheli and Mehta, Parth and Ghosh, Kripabandhu and Ghosh, Saptarshi and Pal, Arindam and Bhattacharya, Arnab and Majumder, Prasenjit, Overview of the FIRE 2020 AILA track: Artificial Intelligence for Legal Assistance. Proceedings of FIRE 2020 - Forum for Information Retrieval Evaluation. Hyderabad, India, December, 2020.