HLJIT2017@IRLed-FIRE2017: Information Retrieval From Legal Documents

Liuyang Tian
School of Computer Science and Technology, Harbin Engineering University, Harbin, China
tianliuyang2016@outlook.com

Hui Ning
School of Computer Science and Technology, Harbin Engineering University, Harbin, China
ninghui@hrbeu.edu.cn

Leilei Kong*
School of Computer Science and Technology, Heilongjiang Institute of Technology, Harbin, China
kongleilei1979@gmail.com

Zhongyuan Han
School of Computer Science and Technology, Heilongjiang Institute of Technology, Harbin, China
Hanzhongyuan@gmail.com

Ruiming Xiao
School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, China
xiaoruiming11@outlook.com

Haoliang Qi
[1]School of Computer Science and Technology, Heilongjiang Institute of Technology, Harbin, China
haoliang.qi@gmail.com
[2]State Key Laboratory of Digital Publishing Technology

ABSTRACT

This paper details the approach of implementing the Catchphrase Extraction and Precedence Retrieval tasks to be presented at Information Retrieval from Legal Documents by Forum of Information Retrieval Evaluation in 2017(Fire2017 IRLeD). For the task of Catchphrase Extraction, the classification-based and Rank-based methods were exploited, and various types of features were attempted. With respect to the task of Precedence Retrieval, the language model, the BM25 and the vector space model were employed. Comparisons to other submissions for the same tasks, show the presented methods to be one of the top performers.

KEYWORDS

Information Retrieval from Legal Documents, Catchphrase Extraction, Precedence Retrieval

1 Introduction

With the recent developments in information technology, the number of digitally available legal documents has rapidly increased. In general, the legal text is long and complex in structure, which makes their thorough reading time-consuming and strenuous[1]. The task of *Information Retrieval from Legal Documents*[1] is devoted to this problem. The task is divided into two parts by Forum of Information Retrieval Evaluation (FIRE): *Catchphrase Extraction* and *Precedence Retrieval*.

The task of Catchphrase Extraction focuses on extracting the catchphrases (short relevant phrases) from legal documents. We formalized the task of Catchphrase Extraction as a multi-classification firstly and used the bagging classification methods to identify the catchphrases. Then, we tried a learning to rank method to rank the words in document to select the catchphrases.

The task of *Precedence Retrieval* can be viewed as an information retrieval problem. Its purpose is to retrieve the relevant prior cases for a given current case. We used three classical models of information retrieval, the language model, the BM25, and the vector space model, to retrieve the relevant documents for a given current case document.

The rest of this paper is organized as follows. In Section 2, we introduced the methods and related features used in Catchphrase Extraction. In Section 3, we described the various search model methods used in Precedence Retrieval. In Section 4, we reported the experimental setting and results. In the last section, we concluded our study.

2 Method of Catchphrase Extraction

Let d_i be a legal document, and a training corpus can be defined as:

$$D = \{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i), ..., (x_n, y_n)\} \quad (1)$$

where $x_i = (x_i^{(1)}, x_i^{(2)} ..., x_i^{(n)})^T, i = 1,2,...,n$. And y_i is the label to denote whether the word w_i is the catchphrase of d_i or not. Then, our goal is to learn a model to decide whether a word is the catchphrase when given a new d_i. The classification-based methods and Ranking-based methods are exploited to learn the model respectively.

2.1 Classification Method: Bagging

Bagging is based on bootstrap sampling. First, we use the M-round bootstrap sampling method to obtain *M* samples containing *N* training samples. Then, based on these sampling sets, a base learner is trained. Finally, the M-based learners are combined. The problem of multi-classification is resolved by a simple voting method[2].

*Corresponding Author
[1] https://sites.google.com/view/fire2017irled/track-description

Decision tree and random forest[3] are adopted as our base classifiers. Denoted as Bagging(DTC) and Bagging(RFC) respectively.

2.2 Ranking Method: RankSVM

RankSVM is a pair-wise learning to rank method that uses the SVM model to solve the ranking problem on document pairs. To rank the candidate catchphrase, we trained a ranking function on a training corpus using the Ranking SVM[2].

2.3 Features

We construct the features from five aspects: statistical features, position features, syntactic features, mutual information and prior probabilities.

Statistical Features We use the term frequency(TF), inverse document frequency(IDF), TF*IDF, and BM25 score of a word in document d_i as the statistical features.

Position Features

1) The first time of the word appears.

$$\text{FirstOccur\_score(w)} = \frac{\sum_{i=1}^{num} FirstOccur\ (w,d)}{num} \tag{2}$$

$$FirstOccur\ \ (w,d) = \frac{precede\ \ (w,d)}{len\,(d)} \tag{3}$$

where num is the number of legal documents, *precede(w,d)* represents the number of words in front of the first occurrence of the word *w* in the legal document *d*, *len(d)* is the number of words of *d*.

2) sentence-initial or sentence-end position.

$$\text{InFirstLast\_}\ score(w) = \frac{\sum_{i=1}^{num} InFirstLast(w,d)}{num} \tag{4}$$

$$FirstOccur(w,d) = \frac{precede(w,d)}{len(d)} \tag{5}$$

POS features We also choose the part-of-speech of a word as the features. For each sentence, we get the POS of each word using the Stanford POS Tagger[6]. We choose a subset of POS (i.e. NNS, NNPS, NNP, NN, VBZ, VBP, VBN, VBG, VBD, VB, TO, JJ, RB) as our features.

num is the number of legal documents, *countFL(w,d)* represents the number of occurrences of the word *w* in the first of sentence or the end of the sentence in the legal document *d*.

Mutual Information High quality keywords should be semantically related. If the relevance between a word w and a keywords *k* is low, then w may not be suitable as a keyword[7]. The degree of correlation between words is measured by the average mutual information.

$$MI(w_1, w_2, ...w_k) = \frac{1}{n} \sum_{i,j=1,2,...k;i\neq j} MI(w_i, w_j) \tag{6}$$

$$I(w_1, w_2) = \sum\sum p(w_1, w_2) \log(\frac{p(w_1, w_2)}{p(w_1)p(w_2)}) \tag{7}$$

where N is the number of word pairs *(w_i,w_j)*, And *$MI(w_i, w_j)$* represents the mutual information of w_i and w_j.

Prior Probabilities Using the *gold standard*, we build a priori keyword set *P*. *P* contains the keywords with their idf is greater than two.

$$\Pr ior\_score(w,P) = \begin{cases} 1, w \in P \\ 0, w \notin P \end{cases} \tag{8}$$

3 Methods of Precedence Retrieval

The task of Precedence Retrieval is to retrieve the prior cases for a given a current case. We view the current case as the query, while the prior cases as the documents, and use three classical information retrieval models to resolve the problem of precedence retrieval.

3.1 Language Model method

For the query model and the document model, we use the language model based on Dirichlet Prior Smoothing[8]. The relevance between query and document is computed as follows.

$$score(q,d) = \log p(q \mid d)$$

$$= \sum_{\substack{q_i \in d \\ w_i \in q}} c(w,q) \log[1 + \frac{c(w_i,d)}{\mu p(w_i \mid C)}] + n \log \alpha_d \tag{9}$$

3.2 Probability Model

The second search model we chose is BM25 model[9]. The relevance score is computed as follows.

$$BM25 = \sum_{w_i \in q}^{n} \log(\frac{tf(w_i,d) \cdot idf(w_i) \cdot (k_1+1)}{tf(w_i,d) + k_1 \cdot (1 - b + b \cdot \frac{len(d)}{avdl})}) \tag{10}$$

where *q* is the query set, *d* is the candidate document, *avdl* is the average length of the document, *k1* and *b* are the adjustment parameters.

3.3 Vector Space Model

We also used lucene[10] which implemented the vector space model to estimate the relevance of query and document. The Formula is shown as follows:

$$score(q,d) = coord(q,d) \times queryNorm(q) \times$$

$$\sum_{t\ in\ q} (tf(t\ in\ d) \times idf(t)^2 \times t.getBoost() \times norm(t,d)) \tag{11}$$

Here, *t* represents the term containing the domain information; *coord (q, d)* means that when a document contains more search terms, the higher score of the document. *tf(t in d)* represents the word frequency that appears in document *d*; *idf(t)* word reverse document frequency; *norm (t, d)* represents the normalization factor;

4 Experimental Results

4.1 Results of Catchphrase Extraction

4.1.1 Dataset

In this task, a set of legal documents (Indian Supreme Court decisions) are provided. For a few of these documents

[2] http://www.cs.cornell.edu/People/tj/svm_light/svm_rank.html

(training set), the catchphrases (*gold standard*) are provided. Catchphrases are short phrases from within the text of the document. These catchphrases have been obtained from a well-known legal search system Manupatra (www.manupatra.co.in), which employs legal experts to annotate case documents with catchphrases. The rest of the documents will be used as the test set. The dataset of Catchphrase Extraction contains 100 training examples and 300 testing examples.

4.1.2 Experimental Settings

Firstly, we get a candidate catchphrase set. The statistics on the training corpus of catchphrase extraction show that, the nouns accounted for 68.85%, the verbs accounted for 9.13%, adjectives accounted for 12.49%, prepositions accounted for 5.90%, adverbs accounted for 1.43%, other types accounted for 2.20%. According to the above distribution, we choose noun, verbs, adjectives and adverbs.

On the candidate set, We have different combinations of features in the classification and ranking methods. The bagging model uses TF*IDF, BM25, Position Features, POS, Mutual Information and Prior Probabilities as feature set. The RankSVM uses TF*IDF, POS and Prior Probabilities as feature set.

A detailed description of the method parameter settings is shown in the following Table 1:

Table 1: Parameter setting of the model in Task1

Method	Parameter
Bagging(DTC)	min_samples_split=2, n_estimators=66, max_samples=0.5, max_features=0.5
Bagging(RFC)	n_estimators=78, min_samples_split=2, max_samples=0.5,max_features=0.5
RankSVM	c=16.0

4.1.3 Results

We submitted the results of the three groups, bagging (DTC)(denoted as HLJIT2017_IRLeD_Task1_1), bagging (RFC)(denoted as HLJIT2017_IRLeD_Tas-k1_2) and RankSVM(denoted as HLJIT2017_IRLeD_Task-1_3). The experimental results are shown in Table 2.

Table 2: FIRE 2017 IRLeD Run Evaluation for Catchphrase Extraction

Method	bagging (DTC)	bagging (RFC)	RankSVM
Mean R Precision	0.0297	0.0335	0.0864
Mean Precision at10	0.0576	0.06	0.1220
Mean Recall at100	0.0328	0.0440	0.1514
Overall Recall	0.0328	0.0440	0.1519
MAP	0.1401	0.1241	0.1649

Note that our three submitted results are closed on the evaluation metrics MAP but different on the other evaluation metrics. We surmise that it is mainly because of the sequence of submitted catchphrases. The two results of the classification methods are sorted by alphabetical order, while the results of ranking are submitted in descending order of sorted scores.

In addition, we only choose the words not the phrases as the catchphrases. We tried to use some rule-based methods to construct phrases according to the word we extracted from the legal documents, but we have not achieved the improvement on performance. The low scores on MRP, MP@10, MR@100 and Overall Recall maybe caused by this reason.

4.2 Results of Precedence Retrieval

4.2.1 Dataset

Task 2 provides two data sets, the 200 current cases (Query_docs) which formed by removing the links to the 2000 prior cases and the prior cases which have been cited by the cases in Query_docs along with some random cases (not among Query_docs).

4.2.2 Experimental Settings

We do the same pre-processing for query extraction and index building. In order to discard some of the interference information in document, we filter the document through the lexical information only the nouns, verbs, adjectives, and Poter stemming, lower case and removing stop words are also implemented. For Language Model, we set the parameter mu=10000 and lambda=0.5, and for BM25, we set k1=1.8 and b=0.7.

4.2.3 Results

In Task 2, We have submitted three group of results Language Model(HLJIT2017_IRLeD_Task2_1), Vector Space Model(HLJIT2017_IRLeD_Task2_2) and BM25(HLJIT2017_IRLeD_Task2_3). The results are shown in Table 3.

Table 3: FIRE 2017 IRLeD Run Evaluation for Precedence Retrieval

Method	Language Model	BM25	Vector Space Model
MAP	0.3291	0.1784	0.2479
Mean reciprocal Rank	0.6325	0.4074	0.5246
Precision@10	0.2180	0.1290	0.1665
Recall@100	0.6810	0.5950	0.6710

From the experimental results, the language model is much better than the other two models. The MAP of the language model is 0.3291, the MAP of the vector space model is 0.2479, and the lowest of the probability model is 0.1784.

Some methods which can improve the performance of language model, such query extension and document extension, have not yet been applied in this evaluation. It may be our further work. In addition, to do the smoothing

of the document D, we only apply the given set of legal documents; Too small collection of documents, resulting in the sparse words, And documents and query models also failed to adjust to the optimal. We believe that these methods will improve the performance of the search model and will be tried in future research.

5 Conclusions

We described the approach to resolve the problems of Catchphrase Extraction and Precedence Retrieval in Fire2017 IRLeD task.

For the task of Catchphrase Extraction, we tried to the classification-based and ranking-based methods. Various type of features is integrated into our models. The experiments show that the ranking-based model achieved better performance.

For the task of Precedence Retrieval, we have only tried several basic language models, such as language model, probability model and vector space model. Experiments show that the language model is more excellent than vector space model and BM25.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.61772177) and the Special subject of State Key Laboratory of Digital Publishing Technology (The research on Plagiarism Detection-From Heuristic to Machine Learning).

References

[1] A. Mandal, K. Ghosh, A. Bhattacharya, A. Pal and S. Ghosh. Overview of the FIRE 2017 track: Information Retrieval from Legal Documents (IRLeD). In Working notes of FIRE 2017 - Forum for Information Retrieval Evaluation, Bangalore, India, December 8-10, 2017, CEUR Workshop Proceedings. CEUR-WS.org, 2017.

[2] Du P, Xia J, Zhang W, et al. Multiple classifier system for remote sensing image classification: A review[J]. Sensors, 2012, 12(4): 4764-4792.

[3] Liaw A, Wiener M. Classification and regression by randomForest[J]. R news, 2002, 2(3): 18-22.

[4] Joachims T. Learning to classify text using support vector machines: Methods, theory and algorithms[M]. Kluwer Academic Publishers, 2002.

[5] Hulth A. Improved automatic keyword extraction given more linguistic knowledge[C]. Proceedings of the 2003 conference on Empirical methods in natural language processing. Association for Computational Linguistics, 2003: 216-223.

[6] Toutanova K, Klein D, Manning C D, Singer Y, 2003. Fea-ture-rich part-of-speech tagging with a cyclic depend-ency network. In Proc. the 2003 Conference of the North American Chapter of the Association for Com-putational Linguistics on Human Language Technology, p. 173–180.

[7] Turney P D. Coherent keyphrase extraction via web mining[J]. arXiv preprint cs/0308033, 2003.

[8] Song F, Croft W B. A general language model for information retrieval[C]. Proceedings of the eighth international conference on Information and knowledge management. ACM, 1999: 316-321.

[9] Robertson S, Zaragoza H. The probabilistic relevance framework: BM25 and beyond[J]. Foundations and Trends in Information Retrieval, 2009, 3(4): 333-389.

[10] McCandless M, Hatcher E, Gospodnetic O. Lucene in Action: Covers Apache Lucene 3.0[M]. Manning Publications Co., 2010.