York University at TREC 2006: Legal Track

Miao Wen[1] and Xiangji Huang[2]

[1]Department of Computer Science, York University, Toronto, Ontario, Canada
e-mail: mwen@cs.yorku.ca

[2]School of Information Technology, York University, Toronto, Ontario, Canada
e-mail: jhuang@yorku.ca

Abstract

York University participated in the legal track this year. For this track, we developed an Okapi-based Legal Search Engine (LSE) v1.0. Our experiments mainly focused on evaluating the effect of a probabilistic text retrieval model on the legal domain. In order to address the special problems in legal text retrieval, new automatic feedback methods and term weighting methods are proposed and tested.

1 Introduction

Legal text retrieval is a particular problem of information retrieval among a wide range of retrieval tasks in different domains. Retrieval in the legal domain is case oriented. Lawyers need to retrieve huge amount of evidence from accessible resources, which is relevant to the problem in litigation. Increasingly, lawyers use automated search and retrieval tools to find useful information from the vast amount of evidence in electronic form. To our best knowledge, most computer aided legal systems are kinds of expert system historically. So far almost all the legal information retrieval systems are based on the boolean retrieval model.

Probabilistic Information Retrieval (IR) model is one of the most classical models in IR. Sound statistic background of the model brings its outstanding performance. Based on this model, term weighting functions are proposed and evolved over the decades. The utilization of relevance information and query expansion are the most important factors of IR, which has been studied almost from the very beginning of IR. The efficiency of it to improve the performance of IR has been affirmed widely. However, applying the probabilistic IR model into legal text retrieval is relatively new. The 2006 legal track provides an uniform simulation of legal text requests in real litigation, which allows IR researchers to evaluate their retrieval systems in the legal domain. One major goal of us is to evaluate the effect of a probabilistic retrieval model on the legal domain.

The other major goal of us this year is to evaluate our new automatic pseudo-relevance feedback process in the legal text retrieval. Pseudo-relevance feedback, also known as blind feedback, is a practical technique commonly used to improve retrieval performance [3, 8]. The basic idea is to extract expansion terms from the top-ranked documents to formulate a new query term set for the second round retrieval. Through a query expansion, some relevant documents missed in the initial round can then be retrieved to improve the overall performance. As one of the most popular and practical relevance feedback approaches, it provides an easy way to obtain relevance document automatically. However, the negative side of blind feedback is its uncontrolled quality on relevance, which could degrades the performance of the retrieval greatly. A more robust and error tolerant feedback algorithm is investigated in our experiments.

Under the MLSRF-pack architecture, we developed an Okapi-based legal search engine LSE1.0 to process all the topics. The whole collection was only indexed at the document level in the experiments. We totally built 6 parallel indexes by using Okapi to facility the query process. A new feedback approach of β-approximation was also tested in our experiments. We submitted two automatic runs in this year. The run without feedback was submitted as our primary run. The other run with a new feedback term selection (TS) and a new term weighting (RW) method was also submitted

In the next section, we describe the overall system architecture. More details on query formulation, parallel databases, new relevant weighting and term selection methods are discussed in algorithm section. Experiments and results are provided in section 4. In the final section, more discussions on the experimental results and future work are given.

2 System Description

Figure 1 depicts the overall structure of our legal search system. We use Okapi 2.31 as the underlying retrieval system, based on which we develop our own multi-database utility model, topic processing and feedback modules.

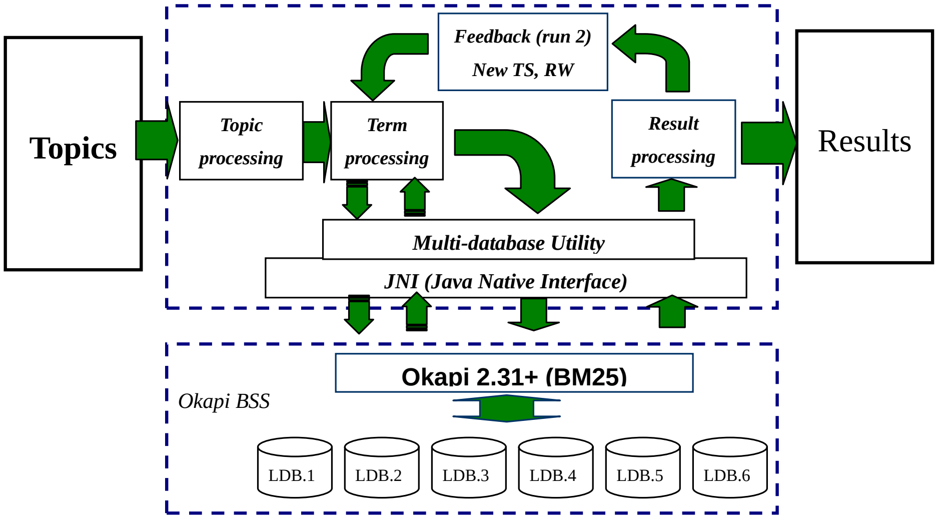

Figure 1: System Architecture

The LSE v1.0 was developed under the frame of MLSRF-pack. The differences between the LSE1.0 and our previous search engine are mainly in two aspects. First, no passage level index was built and used in LSE version 1.0. So the original dual level index and re-ranking was simplified into a single document level index retrieval. However, a room for dual index was reserved and could be implemented. Secondly, we found a lot of OCR errors in the corpus when we index the whole corpus. Because of the OCR errors, over 10 thousands tokens could be generated at the pseudo-relevance feedback stage. Therefore, a new term selection method was designed and evaluated at the pseudo-relevance feedback stage. Details will be given in the next section.

3 Algorithm

In this section, we first discuss how to generate the query terms automatically for each topic. Then we describe the indexing and related okapi-based parallel database. Finally, we present our new term weighting and term selection methods.

3.1 Query formulation

Total forty-six document production requests (ref as ’topics’ in the rest) in five ’Complaint’ sections were given in this year’s legal track. Different from other tracks, topics were not in traditional TREC format, but in a complex XML format with extensive topic description information. We formulated our query by extracting terms from <BooleanQuery> element. Precisely, three elements, <ProposalByDefendant>, <RejoinderByPlaintiff> and <FinalQuery>, were utilized in our query formulation. General algorithm of query formulation is shown as follows:

- 1. Remove all term or phrase defined by ’NOT’.
- 2. Remove all non-literal characters, such as ’”’, ’!’, ’(’,’)’
- 3. Remove all Boolean operators, including ”NOT/not”, ”OR/or”, ”W/x”, etc.
- 4. Tokened the remain string and formulate each term as a array of occurrence in above three elements.

3.2 Parallel databases retrieval model

To improve the flexibility and capability of Okapi interface, we implemented the parallel databases retrieval model so that the upper layer system could access multiple Okapi databases. These databases were indexed under the same schema as if they were indexed in a whole piece. We will discuss this issue in three parts: 1. Indexing parallel databases; 2. Extracting global statistic information and weighting; 3. Merging final results.

3.2.1 Indexing parallel databases

The first task was to separate corpus and generated and indexed okapi databases with same schema. In legal track 2006 case, we were given about 7 million records data as retrieval collection in approximately 57G of uncompressed XML files. After preprocessing XML files, the whole data set ware split into 6 partitions equally. The detail split schema is shown in 1.

	file ID	Num of records
P1	a.a-e.f	1105668
P2	e.g-i.l	1105702
P3	i.m-m.r	1105596
P4	m.s-q.x	1105645
P5	q.y-w.d	1381973
P6	w.e-z.z	1105608
total		6910192

Table 1: 2006 Legal corpus split schema

All partitions shared same extraction pattern and okapi indexing schema. For each document in the corpus, eleven contents were abstracted from source collection if it was presented. Abstraction and indexing schema were shown in 2.

More detailed discussion about XML elements were given on the 2006 legal track Web site [2]

XML element	description	indexing
<A ID>	ID of record	single
<ot>	OCR of record	joined with K
<K>	Record title	joined with ot
<ci>	ID of legal case	single
<d>	Description of record	single
<m>	Person or org. in record	single
<DS>	Document source	single
<bt>	Bates Number	single
<dt>	Document type	single
<lu>	Litigation Usage	single
<si>	Document Site	single

Table 2: 2006 Legal abstraction and indexing schema

3.2.2 Extracting probabilistic information and weighting

The traditional BM25 weighting function is shown as follows:

$$\omega = \frac{(k_1+1) * tf}{K + tf} * w^{(1)} * \frac{(k_3+1) * qtf}{k_3 + qtf} \quad \oplus \quad k_2 * nq * \frac{(avdl - dl)}{(avdl + dl)} \tag{1}$$

$$w^{(1)} = \log \frac{N - n + 0.5}{n + 0.5} \tag{2}$$

where w is the weight of a query term, N is the number of indexed documents in the collection, n is the number of documents containing the term, R is the number of documents known to be relevant to a specific topic, r is the number of relevant documents containing the term, tf is within-document term frequency, qtf is within-query term frequency, dl is the length of the document, $avdl$ is the average document length, nq is the number of query terms, the k_is are tuning constants (which depend on the database and possibly on the nature of the queries and are empirically determined), K equals to $k_1 * ((1-b) + b * dl/avdl)$, and $\oplus$ indicates that its following component is added only once per document, rather than for each term. In our experiments, the values of k_1, k_2, k_3 and b in the BM25 function are set to be 1.2, 0, 8 and 0.75 respectively.

In parallel databases solution, the key issue was to extract statistic information of query terms from all databases, globally, but from any individual database, so BM25 function could maintain valid. We abstracted :

N_i Number of record in i^{th} partition database

n_i Number of units containing a specific term in i^{th} partition database

By substituting N and n in equation 2 with following:

$$N = \sum_{k=1}^{n} N_i \tag{3}$$

$$n = \sum_{k=1}^{n} n_i \tag{4}$$

We obtained precise global statistic information of query terms, so that the term weighing of BM25 still holds in our parallel database solution.

3.2.3 Merging final results

After obtaining the global weight for each query term, the whole set of terms are queried upon each partition database. Each one will retrieve a ranked list of relevant records within each partition. The final result is then generated by merging all sub-results from all the partitions and ranking according to their relevant scores.

3.3 A new method for term weighting

3.3.1 Relevant weight of BM25

Without relevant information, term weighting function(2), was simplified to IDF-like function. However, the utilization of relevant information was one of the most important component in Probabilistic retrieval model. RSJ relevance weighting of query terms was proposed in 1976 [5] as an alternative term weighting of 2 when relevant information is available. As shown in 5,

$$w^{(1)} = \log \frac{(r)/(R-r)}{(n-r)/(N-n-R+r)} \tag{5}$$

R and r were introduced into term weighting, in which:

- R is the number of documents known to be relevant to a specific topic,
- r is the number of relevant documents containing the

The above weighting was applicable with idea relevance information available, and under the assumption of: (1) "The term distribution in the relevant items previously retrieved is the same as the distribution for the complete set of relevant items";(2) "all non-retrieved items can be treated as non-relevant"[1]. However, the idea relevance information only existed theoretically, to make the weighting function more practical, a point-5 version of approximation is suggested, when not all the relevance information is available.

$$w^{(1)} = \log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} \tag{6}$$

The function 6 is consistent with 2 when relevant information is not available, which R and r are reduce to 0.

3.3.2 β approximation

For the same reason of point-5 version of approximation in RSJ, relevant information was never ideal in practical case. The universal 0.5 approximation to every terms is somewhat an arbitrary solution. So we tried to find an alternative solution, which can adjust approximation according to terms' own probabilistic character in collection, so that weighting of terms could be more accurate.

Our new method was also based on the original theoretical principle of term relevance weighting [5, 1], same as BM25 relevance weighting was:

$$w = \log \frac{p(1-q)}{(1-p)q} \tag{7}$$

- p is the probability of a document contains a term, given that it is relevant.
- q is the probability of a document contains a term, given that it is not relevant.

Considering the property of relevance information obtained by pseudo-relevance feedback and the deduction fashion in [9], the approximation of p = 0.5 without any relevance information, we could more safely approximate that p = β, (0.5,1) for each terms in feedback information. So the weighting function with relevance information could be deduced as:

$$w^{\beta} = \log \frac{N - n\beta}{(1-\beta)n} \tag{8}$$

The left question was how to approximate β for each term based on relevant information. One of our proposed method was to utilize the change of term's 'density' in whole data set and in feedback information. Given following definition:

- N' is the number of documents in feedback collection,
- n' is the number of documents containing the term in feedback collection,

We believed that the density holds certain association to p, and if we assumed that there was a linear relationship between them. Then we have:

$$\frac{n}{N} : p = \frac{n'}{N'} : k\beta \tag{9}$$

$$\beta = \frac{N \cdot n'}{N' \cdot n} \times p \times \frac{1}{k} \tag{10}$$

Where p=0.5

Ideally, β should be in range of (0.5, 1). Because of the size of feedback information ware normally far smaller than that of original collection's, it was practically in range of (0, $+\infty$). To adapt it for formula (5). We applied following:

$$\beta' = \frac{e^{\beta}}{1 + e^{\beta}} \tag{11}$$

The above was the general idea of β approximation on term weighting with unreliable relevance information.

3.4 A new term selection method

Term selection was a practical as well as critical problem of feedback process, which related to query expansion strategy. Technically, more than hundreds of terms could be abstracted from feedback information, even only top 10 documents were chosen. To identify and select the most "useful" terms amount them automatically, selection criteria was needed to be designed very carefully. Robertson talked about criteria in [4, 7], shown in .

$$a_t = w_t * (p_t - q_t) \tag{12}$$

- w_t is the weight of term t,
- p_t is the probability of a document contains a term, given that it is relevant.
- q_t is the probability of a document contains a term, given that it is not relevant.

in which, p_t and q_t may be estimated from relevance feedback information.

When considering above criteria within RSJ weighting schema, a_t was actually equivalent to w_t mathematically, and by R-r relevant weighting, only partial probabilistic information of terms within feedback documents was used. More fully utilizing of terms' statistic information was explored in our experiment as a complement to traditional methods.

$$a_t = w^{\beta} * \bar{m} * \frac{1}{\sqrt{\sum_{i=1}^{n_{fb}} (o_{ti} - \bar{m})^2}} * \frac{(k_3 + 1) * qtf}{k_3 + qtf} \tag{13}$$

$$\bar{m} = \frac{\sum_{i=1}^{n_{fb}} o_{ti}}{n_{fb}} \tag{14}$$

where

- n_{fb} is the number of selected feedback documents in feedback collection. Traditionally it is 10 and can be increased to improve performance according different topics.
- o_{ti} is the number of occurrence of term t, in the i^{th} feedback document.

4 Experiments and Results

For the 2006 legal track, we submitted two runs "york06la01" and "york06la02" in total. The difference between those two runs is shown in Table 3. Their evaluation results are presented in Table 4. Although the retrieval results we submitted are based on all the 46 topics, only 39 topics are counted in the final evaluation. Therefore, the evaluation results shown in Table 4 are generated over these 39 topics instead of the whole 46 topics.

run ID	parallel DB	feedback	β-approximation	Term Selection (13)
york06la01	Yes	No	No	No
york06la02	Yes	Yes	Yes	Yes

Table 3: 2006 Legal York runs' setting

run ID	map	R-prec	bpref
york06la01	0.1031	0.1555	0.1684
york06la02	0.0952	0.1393	0.1628

Table 4: 2006 Legal York runs' result

According to the evaluation results of these 39 topics, the average number of relevant documents per topic is 111. Topic 19 has 502 relevant documents. There are 15 topics that have over 100 relevant documents. Comparing to the 2004 and 2005 topics of HARD track, the 2006 topics of legal track are much more rich in terms of the number of relevant documents. However, for the "Average Precision" measure, the upper boundary among all the 39 topics is 0.2539. Similarly, the median value is 0.0499. In terms of "R-prec", the upper boundary is 0.3270, while the median value is just 0.0863.

5 Conclusion and Future work

There are a lot of spaces to improve the retrieval performance. First, we find that there are many OCR errors in the 2006 legal corpus. Some errors can be recognized. But many errors are difficult to deal with. We tried to clean those errors by defining some rules. However, we found that it is almost impossible to find all the errors by this method. Without finding an effective method to handle this problem, those OCR errors will be put into our Okapi indexes. However, the incorrect statistic information for a term will have a negative impact on the retrieval performance. Secondly, the average length of legal documents is long. For example, the average length of documents for the 2004 HARD corpus is 2188 [10], while the average length of documents for the 2006 legal corpus is 4549. Therefore, the original settings for those tuning constants k_1, k_2, k_3 and b in the BM25, which were set to 1.2, 0, 8 and 0.75 in our experiment, might not be a good setting any more. Finally, to find a better term selection method is also our next step work.

6 Acknowledgement

This research is supported by research grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada and Atkinson Faculty of York University.

References

[1] G. Salton and C. Buckley. *Improving retrieval performance by relevance feedback.* Journal of the American Society for Information Science, 41(4):288-297, 1990.

[2] Legal discovery track 2006 official web site *http://trec-legal.umiacs.umd.edu*

[3] N.E.Efthimiadis *Query Expansion.* In Annual Review of Info. Systems and Technology, 31:121-187, 1996.

[4] S.E. Robertson, *On term selection for query expansion.* Journal of Documentation 46, 359-364, 1990.

[5] S.E. Robertson and K. Sparck Jones, *Relevance weighting of search terms.* Journal of the American Society for Information Science 27, 129-46, 1976.

[6] S.E. Robertson and S. Walker *Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval.* Presented at SIGIR 94, Dublin, 1994

[7] S.E. Robertson and K. Sparck Jones, *Simple, proven approaches to text retrieval.* University of Cambridge Computer Laboratory Technical Report no. 356, 1994 updated 1996,1997,2006.

[8] Thomas R. Lynam, Chris Buckley, etc. *A multi-system analysis of document and term selection for blind feedback.* Proceedings of the thirteenth ACM international conference on Information and knowledge management CIKM '04, November 2004

[9] W. Croft and D. Harper *Using probabilistic models of information retrieval without relevance information.* Journal of Documentation 35, 285-295, 1979.

[10] X. Huang, Y.R. Huang. *York University at TREC 2004: HARD and genomics Tracks* The proceedings of the Thirteenth Text REtrieval Conference, 2004.