DUTIR at TREC 2009: Chemical IR Track

Song Jin, Zheng Ye, Hongfei Lin
Information Retrieval Lab, Dalian University of Technology
No. 2 LingGong Road GanJingZi District, Dalian 116023, China
{jinsong, zye}@mail.dlut.edu.cn, hflin@dlut.edu.cn

Abstract

This paper presents the DUTIR submission to TREC 2009 Chemical IR Track. This track included two tasks: Prior Art (PA) and Technical Survey (TS) tasks. We present a series of experiments on two text retrieval models, BM25 and Language Model for IR (LMIR). For Prior Art task, we focused on formulating the queries from the query patents and date filtering. Moreover, some traditional search techniques are used for Technical Survey task.

1. Introduction

In this paper, we describe the work done by members at Dalian University of Technology in China for TREC 2009 Chemical IR Track. In particular, we present a series of experiments for Technical Survey (TS) and Prior Art (PA) tasks in Chemical IR Track 2009. Our work mainly focuses on the following aspects: (1) the performances of traditional retrieval models work for discovery of the chemical patents and academic journal articles; (2) the performances of several search techniques (e.g. query expansion, weighting model) for the tasks.

1.1 Chemical IR Track

TREC 2009 was the first year of the Chemical IR Track. This newly organized track addresses challenges in building large chemical datasets to evaluate the state of the art in chemical and patent information retrieval tools. This track included two sub-tasks: Technical Survey (TS) and Prior Art (PA) tasks.

Technical Survey Task: This task contains 18 topics created by chemical patent experts based on their information needs. Participating systems are asked to return a set of documents that answer the information need as best as possible. The aim of this task is to understand the pros and cons of the participating systems in finding relevant chemical documents and how effectiveness can be improved.

Prior Art task: The second task asked participating systems to find relevant patents with respect to a set of 1,000 existing patents. As the query topics, the 1,000 patents are automatically generated from the patent dataset. This task was intended to investigate how to design both effective and efficient systems that can retrieve high quality relevant patents for a rather large number of topics.

1.2 Collection

The datasets of Chemical IR Track consist of about 1.2 million chemical patent files (approximately 98GB in size) and 59 thousand scientific articles (around 3GB in size). The patent collection covers patents

in the field until 2007, registered at three major patent offices (EPO, USPTO, and WIPO). The chemical scientific articles are extracted from 31 journals published by the RSC (Royal Society of Chemistry). All the data is in XML format. More detailed information about the data collection can be found in[1].

The remainder of this paper is organized as follows. In Section 2, we describe the methods for technical survey task. In Section 3, we introduce our approach for prior art task. In Section 4, we simple present our official results in TREC 2009 Chemical IR Track. In Section 5, we conclude the paper and discuss future work.

2. Our Methods for Technical Survey Task

The TS task is similar to a traditional ad hoc retrieval task. Both chemical patent files and scientific articles datasets are used in this task. We respectively use the information in title and narrative fields of the topics for different methods.

2.1 Baseline Model

Our baseline run, *DUT09TSRun1*, is a simple title-only query-likelihood run. In the experiment, the Indri 2.6 search engine [1] is used as our basic retrieval system, and documents are retrieved for a query by the query-likelihood language model [4] with Dirichlet smoothing [5]. We set the Dirichlet prior empirically at 1,500 as recommended in [2].

For example, Topic 15 “Betaines for peripheral arterial disease” is converted into the following Indri query:

#*combine*(betaines for peripheral arterial disease)

which produces results rank-equivalent to a simple query likelihood language modeling run.

2.2 Query Expansion Methods

Automatic query expansion is a widely used technique in IR. In the experiments, to select useful expansion terms, we use two heterogeneous resources. One is the pseudo-relevance documents, and the other is the contents in narrative field of topics.

Pseudo-Relevance Feedback (PRF) has been shown to be an effective way of improving retrieval performance. In this track, we use a modified version of Lavrenko and Croft’s relevance model [3]. This model is a multinomial distribution which estimates the likelihood of term q given a query Q.The query terms $q_1, q_2, ..., q_m$ and the term t in relevant documents are sampled identically and independently from a distribution R. The relevance model is then estimated as follows:

$$P(t \mid R) \approx \sum_{D \in F} \frac{P(t \mid D)P(Q \mid D)P(D)}{P(Q)} = \sum_{D \in F} P(t \mid D)P(Q \mid D) \quad (1)$$

where F denotes the feedback documents. Based on this estimation, the most likely expansion term e from $P(t \mid D)$ is chosen for the original query. The above relevance model is used to enhance the original query model by the following interpolation:

$$P(t \mid Q') = (1-\lambda)P(t \mid Q) + \lambda P(t \mid R) \quad (2)$$

[1] https://wiki.ir-facility.org/index.php/Data

DUT09TSRun3 was carried out using the Indri search engine. Given an original query Q, we retrieve a set of N documents and form a relevance model from them. Then we form Q_{RM} by wrapping a $\#combine$ around the k most likely terms from the relevance model that are not stopwords.

The contents in narrative field are used to describe the topics more detailed. Thus, in *DUT09TSRun2*, we extract the non-stopword terms with the largest probabilities from narrative field for each query. Using these expansion terms, we also form Q_{RM} for each original query. Finally, an expanded query is formed in the following form:

$$\#weight(\ \lambda_{fb}\ Q\quad (1.0-\lambda_{fb})\ Q_{RM}\ )$$

In the experiments, we set the parameters $\lambda_{fb}=0.7$, $k=50$, $N=10$.

2.3 Weighting Models

In our experiments, we explore two traditional weighting models, BM25 and DFR, which perform well on a large number of IR collections.

2.3.1 BM25

In BM25, the weight of a query term is computed based on its statistics of a document and the whole collection. The corresponding weighting function is as follows:

$$w(q_i,d)=\frac{(k_1+1)*tf}{k_1*((1-b)+b*dl/avdl)+tf}*\log\frac{N-n+0.5}{n+0.5}*\frac{(k_3+1)*qtf}{k_3+qtf} \tag{3}$$

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, 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, the $k_i s$ are tuning constants (which depend on the database and possibly on the nature of the queries and are empirically determined).

In our experiments, the values of k_1, k_3 and b in the BM25 function are empirically set to be 1.2, 8 and 0.75 respectively, which has been proven to perform well on a large number of test collections.

2.3.2 BM25F

Zaragoza et al. [6] introduced a field-based (e.g. title, abstract, body) version of BM25, called BM25F, in which the frequencies of each field are combined together, and then the resulting *pseudo frequency* are used in the BM25 weighting function. In particular, the term frequencies in each field are combined in a linearly weighted sum to obtain the final term *pseudo frequency*, which is then used in the usual BM25 saturation function. The corresponding weighting function is as follows:

$$BM25F(t,d)=\frac{tf_{d,t}}{K_1+tf_{d,t}}w_t^{(1)}$$
$$tf_{d,t}=\sum_{f\in F}w_f*\frac{tf_{d,f,t}}{(1-b_f)+b_f\dfrac{sl_f}{avsl_f}},0\le b_f\le 1 \tag{4}$$

where $w_t^{(1)}$ is the usual RSJ relevance weight for term t, which reduces to an idf weight in our experiments, f is a field in document d, b_f is a normalization factor for field f, w_f is a weight for field f, F is a set of field, K_1 is single saturating parameter. In equation (4), there are $2|F|+1$ parameters in total.

In our experiments, we divide the patents into four fields, namely "title" "abstract" "description" and "claims". For the parameters in equation (4), we empirically set b_f, to be 0.75 for each field. For the weight of each field, we empirically set to be 3 (title), 2 (abstract), 1 (description) and 0.5 (claims).

2.3.3 DFR (In_expB2)

In our study, we apply the In_expB2 weighting model [7], derived from the Divergence From Randomness (DFR) framework. The relevance score of a document d to a query Q is given by

$$\begin{aligned} score(d,Q) &= \sum_{t \in Q} TF * qtf * NORM * \log_e\left(\frac{N+1}{n\_\exp}\right) \\ TF &= tf * \log_2(1 + avdl / dl) \\ NORM &= (tf+1)/(df * (TF+1)) \\ n\_\exp &= df * (1 - e^{-qtf/df}) \end{aligned} \tag{5}$$

where N is the number of indexed documents in the collection, 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. Since it is a parameter-free model, there is no parameter required to be tuned.

For our official run *DUT09TSRun4*, we rank the documents according to a weighted sum of relevance score of each field. For the relevance score of a field to a query, the field level statistics of each term are used in equation (5).

For our official run *DUT09TSRun6*, the In_expB2 weighting model is used. Note that, in this run, the term frequency is a *pseudo frequency* described in Section 2.3.2, since we index the collection on field level.

For our official run *DUTIR09BM25F*, we use BM25F structure-based weighting model and the term frequency is a pseudo frequency described in Section 2.3.2.

3. Our Methods for Prior Art Task

The PA task asked participating systems attempt to identify all relevant documents with respect to a set of 1,000 existing patents. It also contained a mini-task, where the participants were invented to submit the results to only the first 100 patents in the list. This year, we only submitted the results for the short PA task.

The challenge of the PA task is that the query, a whole patent document, is typically quite long. To solve this problem, we use some query processing techniques to formulate the queries. The second step of this task is data filtering. Once a set of relevant patents obtained, the system will filter out the patents with priority data after the latest priority date of the query patent. Notice that, only the chemical patent files dataset is required for this task.

The query patent contains four fields at least, namely "title", "abstract", "claims" and "description". Obviously, the words in title field are more important than them in other fields. We used all title words, and selected sets of terms from the abstract, claims and description fields to construct the queries. In

DUTIRRun1, we use m words in title field and select top n words with highest *TF-IDF* scores from other three fields that are not stopwords.

In *DUTIRRun1*, we consider the abstract, claims and description fields as the whole sample to select the words. After the observation of new queries, we found many expansion terms from the description field, because this field contains more words than other two. In *DUTIRRun2*, we use the $fieldLength_i$ to solve this problem.

$$\underset{Field_i}{Score}(e) = \log(tf_i(e) \cdot idf \ / \ fieldLength_i) \tag{6}$$

$$Score(e) = \sum_{i=1}^{3} \underset{Field_i}{Score}(e) \tag{7}$$

where $tf_i(e)$ is the term frequency in the field i of query patent, and $fieldLength_i$ is the length of the field i. The $Score(e)$ represents the importance of word e to some extant. We select the words with the weight as follows:

$$Weight(e) = \frac{Score(e)}{MaxScore} \tag{8}$$

where $MaxScore$ is the maximum score of all the words. When word w is from title field, we set $Weight(w) = 1$. In the experiments, we set the parameters: $m + n = 60$.

In *DUTIRRun3*, we first obtain the top ranked 1,500 patents from the target collection to be searching using the query-likelihood language model. Then we re-rank the 1,500 patents using the words in title field of query patent as the query by the query-likelihood language model. The language model treats documents themselves as models and a query as strings of text generated from these documents models. The query likelihood model estimates document language models using the maximum likelihood estimator. The documents can be ranked by their likelihood of generating or sampling the query from document language models: $P(Q \mid D)$.

$$P(Q \mid D) = \prod_{i=1}^{m} P(q_i \mid D) \tag{9}$$

where q_i is the i th query term, m is the number of words in a query Q, and D is a document model.

Dirichlet smoothing is used to estimate non-zero values for terms in the query which are not in a document. It is applied to the query-likelihood language model as follows:

$$\begin{aligned} P(w \mid D) &= \frac{|D|}{|D| + \mu} P_{ML}(w \mid D) + \frac{\mu}{|D| + \mu} P_{ML}(w \mid Coll) \\ P_{ML}(w \mid D) &= \frac{freq(w, D)}{|D|}, P_{ML}(w \mid Coll) = \frac{freq(w, Coll)}{|Coll|} \end{aligned} \tag{10}$$

where $P_{ML}(w \mid D)$ is the maximum likelihood estimate of word w in the collection D, $Coll$ is the first retrieval collection (2,000 patents), and μ is the smoothing parameter. $freq(w \mid D)$ and $freq(w \mid Coll)$ respectively denote the frequency of a word w in D and $Coll$. $|D|$ and $|Coll|$ are the lengths of a document D and collection $Coll$.

4. Experiments

In the preprocessing of the collection, we remove some connectives (e.g. "/", "-", "_"), then use blank delimiter to segment words. Porter stemming and stopword removal are conducted in indexing and searching processes. Beside these simple steps, no further technologies have been used. In the following, we present our official experimental results.

Table 1 and Table 2 show our official runs for Technical Survey and Prior Art Task respectively.

Table 1. Results from each team in terms of xinfAP and the inferred NDCG.

Run	xinfAP	infNDCG
DUT09TSRun1	0.211250859	0.458570893
DUT09TSRun2	0.24820495	0.488650961
DUT09TSRun3	0.23876087	0.479486965
DUT09TSRun4	0.247897016	0.453953889
DUT09TSRun6	0.301352563	0.535624287
DUTIR09BM25F	0.245096335	0.480607796

Table 2. Results for 5 popular measures for the short PA task.

Run	MAP	b-pref	MRR	P_30	Recall_100	NDCG
DUTIRRun1	0.0195	0.0932	0.1060	0.0397	0.0491	0.0683
DUTIRRun2	0.0203	0.0969	0.0924	0.0420	0.0508	0.0695
DUTIRRun3	0.0204	0.0984	0.0932	0.0397	0.0517	0.0702

5. Conclusions

In this paper, we present our participation in Chemical IR Track 2009. For technical survey task, our experiments were conducted on two text retrieval models, BM25 and Language Model for IR (LMIR). We attempted the PRF via Lavrenko's relevance model for query expansion in this task, and we used a combination of different weighting schemes (BM25 and DFR) together. For prior art task, we focused on formulating the queries from the query patents. Top 60 terms with the largest probabilities (TF-IDF scores) from different fields (e.g. title, abstract, description, claims) were selected as the original query to retrieval the relevance documents. However, the results are not satisfactory, and we will explore more distinctive methods for the query patent process. We leave these limitations as our future work.

6. Acknowledgements

This research is jointly supported by Natural Science Foundation of China (No.60673039 and 60973068), National High Tech Research and Development Plan of China (2006AA01Z151) and Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

References

[1] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. In Proceedings of the International Conference on Intelligence Analysis, 2004.

[2] D. Metzler, T. Strohman, H. Turtle, and W. B. Croft. Indri at TREC 2005: Terabyte track. In Proceedings of TREC 2004, 2004.

[3] V. Lavrenko and W. B. Croft. Relevance based language models. In Proceedings of SIGIR 2001, pages: 120–127, 2001.

[4] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In Proceedings of SIGIR 1998, pages: 275-281, 1998.

[5] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems, 22(2), pages: 179-214.

[6] H. Z. Nick, N. Craswell, M. Taylor, S. Saria, and S. Robertson. Microsoft Cambridge at TREC-13: Web and hard tracks. In Proceedings of TREC 2004, 2004.

[7] G. Amati. Probabilistic models for information retrieval based on divergence from randomness. PhD thesis, Department of Computing Science, University of Glasgow, 2003.