Query Formulation for Prior Art Search - Georgetown University at CLEF-IP 2013

Jiyun Luo and Hui Yang

Georgetown University, Department of Computer Science,
37th and O Streets NW, Washington DC, USA
jl1749@georgetown.edu,huiyang@cs.georgetown.edu

Abstract. Our group participated in the CLEF-IP 2013 Passage Retrieval starting from Claims task. We focus on formulating representative queries from various metadata that is embedded in a patent document. We then submit the queries to a state-of-the-art search engine to perform document level retrieval. For passage level retrieval, we implement a TF-IDF algorithm that calculates the sum of query keywords' TF-IDF scores. We submitted six runs, which tested different uses of the metadata and different retrieval algorithms. We find that carefully constructed structured queries from titles and terms with mid-range IDF values are effective for patent prior art retrieval.

Keywords: Patent Search, Prior Art Search, Query Formulation

1 Introduction

A patent is a set of legal documents authorized by a government's patent office. It is used to grant exclusive rights for exploitation of the invention for a span of time, usually 20 years. A Patent Application document is written by the patent applicant to describe the background and the description of the invention, and to declare a set of claims. The claims are usually drafted with the help of a patent attorney, and are used to specify what exactly the patent should protect.

The novelty search, also called Prior Art Search, is the procedure that patent examiners search for existing patent documents, which are called prior arts, to prove that the all or part of the claims in a newly filed patent application document are not novel and hence can be rejected. CLEF-IP 2013 Passage Retrieval starting from Claims task exactly captures the procedure of the novelty search. Given one or a few claims, the participants are asked to retrieve relevant patent documents in the collection and mark out the relevant passages. The following is an example of a CLEF-IP query:

<tfile>EP-1752179-A2.xml</tfile>
<title>Needle guard clip with stylus</title>
<abstract>A needle guard (10) includes a clip (12) with a canting wall (16) to grip the needle shaft (56) and a distal wall (22) to block the tip (58) thereof, wherein the canting and distal walls may be interconnected by an angled strut (24,26)...

</abstract>
<tclaims>/patent-document/claims/claim[1][8]</tclaims>
<claim num="1">
A safety catheter device comprising a catheter hub and a catheter tube extending therefrom, a needle having a needle shaft terminating in a sharp tip, ...
</claim>
<claim num="8">
A needle protector device comprising a housing adapted to slidably receive a needle therethrough, a clip positioned in the housing and having a first wall with an aperture adapted to slidably receive a needle shaft of the needle...
</claim>

In the example, the participants are asked to search the given patent corpus and to retrieve prior arts for the patent file *EP-1752179-A2.xml* with relevant passages being marked out. These retrieved documents and passages should be evidence to help patent examiners reject the 1^{st} and the 8^{th} claims in the patent application.

The data collection [3] used in CLEF-IP 2013 Passage Retrieval starting from Claims task is consisted by XML patent documents from European Patent Office (EPO) prior to year 2002 as well as over 400,000 documents published by the World Intellectual Property Organization (WIPO). The documents are multilingual, including English, German and French. No images are kept in the collection.

Table 1. Data Fields Indexed by Lemur

Field	**Description**
abstract	the abstract section in a patent document
applicants	metadata about the applicants in a patent document, including the name, address and country of each applicant
application-date	the application date
application-reference	includes country code, application document number, patent kind code and application-date
claim-num	claim number used as the identification number of a claim
claims	the claims section in a patent, including all claim texts, and claim numbers
date	all date fields in a patent document
description	the description section in a patent document
inventors	metadata about the inventors in a patent document, including the name, address, and country of each inventor
priority-claims	the priority claim section in a patent document
publication-date	the publication date
publication-reference	includes country code, publication document number, patent kind code, and publication-date
title	the English title
docno	the external document id

2 Dataset Preparation and Document Indexing

We adopt the Lemur Search Engine[1] to build index and retrieve the patent documents. Specifically, we use Lemur to build inverted index for each word in the CLEF-IP collection except the stopwords. We stem the terms using the Krovetz stemmer [2]. To allow structured retrieval, we also index many fields that are present in the patent application documents. Table 1 gives a complete list of indexed fields and their detail descriptions.

The Lemur Search Engine implements many retrieval algorithms, including the vector space model, Language Modeling, and Okapi BM25 [1]. In our work, we focus at generating highly representative query keywords. During retrieval, we adopt the algorithms that implemented by Lemur directly. The two particular algorithms that we use are Language Modeling with Dirichlet smoothing and Okapi BM25.

The language modeling with Dirichlet smoothing can be shown as in the following scoring formula:

$$P(t|d) = \frac{tf_{t,d} + \mu P(t|M_C)}{\sum_{t' \in V} tf_{t',d} + \mu} \tag{1}$$

where $tf_{t,d}$ means term t's term frequency in document d, M_C is the corpus model, V is the Vocabulary. In order to get the best value for parameter μ, we fixed the input query keywords, and switched μ between {3000, 3500, 3700, 3800, 3900, 3950, 4000, 4050, 4100, 4200, 4300, 4500, 5000, 5500}. Our experiments shows that 4050 is the best μ value.

Okapi BM25 follows the scoring formula as below:

$$\sum_{t \in Q} (\log \frac{N - df_t + 0.5}{df_t + 0.5}) \frac{(k_1 + 1) tf_t}{k_1((1-b) + b\frac{doc\_len}{avg\_doc\_len}) + tf_t} \frac{(k_3 + 1) qtf_t}{k_3 + qtf_t} \tag{2}$$

where Q is the query keywords set, N is the number of documents in the corpus, df_t is term t's document frequency, tf_t is term t's term frequency in document d, doc_len is the length of document d as the number of terms, avg_doc_len is the average length of a document, qtf_t is term t's term frequency in query set Q. There are 3 parameters (k_1, b ,k_3) in Okapi's scoring formula. In order to evaluate the best value set for (k_1, b ,k_3), we fixed b=0.75, k_3=7, and varied k_1 through 1.0 to 10. We perform parameter tuning to b and k_3. The experiments show that the best value set for (k_1, b ,k_3) is (8.0, 0.85, 1000).

3 Query Formulation

In this section, we present several approaches to formulate queries from patent documents.

[1] http://www.lemurproject.org/

Extracting claim texts We directly extract task claims out of patent documents and use these claims as query keywords to retrieve documents. For patent *EP-1752179-A2*, we get query:

> *A safety catheter device comprising a catheter hub and a catheter tube extending therefrom, a needle having a needle shaft terminating in a sharp tip, ...*
> *A needle protector device comprising a housing adapted to slidably receive a needle therethrough, a clip positioned in the housing and having a first wall with an aperture adapted to slidably receive a needle shaft of the needle...*

Extracting hyphenating phrases We believe that hyphenating phrases like "water- bed' are usually representative words, hence we propose to form queries by extracting hyphenating phrases from the claims.

Extracting titles We believe that a document's title usually summarizes a document well. In this approach, we use patent title to generate queries. Some patent application documents may have multiple titles; each title is written in English, German, or French. We refer the title written in English as the English Title, and the title which shares the consistent language with the patent application document as the Consistent Title. For example, patent *EP-0195350-A2* is written in German and has three titles:

English Title: Method for regenerating carbon articles
German Title: Verfahren zur Regenerierung von Formkörpern aus Kohlenstoff
French Title: Procédé pour régénérer des corps en carbone

The first title is called the English Title and the second title is called the Consistent Title since it matches with the language document of the application. For a patent document written in English like *EP-1752179- A2* (see Section 1), its English Title is also the Consistent Title.

Through experiments on last year's data, we find out that if we retrieve results using the Language Modeling approach, we get better results when we use the Consistent Title; on the other hand, if we retrieve results using the Okapi BM25 Model, we get better results when we add both the Consistent Title and the English Title into the query keywords.

IDF filtering Each of the previous three approaches alone or their combinations can provide us with a set of query keywords. Even though we remove stopwords out of the query keywords, there are still many terms like "water" which are not stopwords but are also common in the document collection; hence they are not representative. In this approach, we propose that using IDF (inverse document frequency) [5] to filter query keywords. Terms with a very low IDF value are common words in the corpus, while terms with a very high IDF value has a high possibility to be a typo.

We propose a two layer filtering strategy. We believe that hyphenating phrases and terms extracted from title are better words than words extracted from claims. Based on this assumption, we split query keywords into two sets. One set is called the Standard Query, it contains terms extracted from claims. The

other set is called the Refined Query, it contains hyphenating phrases and terms extracted from titles. We conduct a stricter filtering strategy on the Standard Query and a looser one on the Refined Query. That is when we filter the Standard Query, we set the IDF's lower bound to be 0.7 and the upper bound to be 3.2, while when we filter the Refined Query, we set the best IDF lower bound to be 0.65 and the upper bound to be 3.2. These values are decided by a series of experiments based on CLEF-IP 2012 Passage Retrieval starting from Claims training tasks and testing tasks.

POS tagging Another way to filter query keywords is using POS [4] tagging. We use Stanford Log-linear Part-Of-Speech Tagger[2] to identify query keywords. Initially we thought only nouns and adjectives are representative words and need to be kept, but our experiment results shows that this strategy is too aggressive. In the end, we loosed it a bit that we kept verbs in the query keywords set too.

4 Experimental Results

The runs that we submitted to this year's CLEF-IP Prior Art Retrieval Task used a combination of some or all of the approaches that we list in Section 3. These runs are:

1. Run ID: OnlyClaimLM
   - extract the task claims to be query keywords
2. Run ID: coOnlyTtlLM
   - use patent application's Consistent Title as query keywords
3. Run ID: HypCoTtlNoIdfUpperBoundLM
   - extract claims, hyphenating phrases and the Consistent Title to form query keyword set
   - filter the Original Query with IDF lower bound 0.7 and the Refined Query with IDF lower bound 0.65, no IDF upper bound is set
   - POS tagging query keywords and only leave nouns, adjectives and verbs
4. Run ID: HypCoTtlWithIdfUpperBoundLM
   - the same as <HypCoTtlNoIdfUpperBoundLM>, but also filter the Standard Query and the Refined Query using IDF upper bound 3.2
5. Run ID: HypDuTtlNoIdfUpperBoundBM
   - the same as <HypCoTtlNoIdfUpperBoundLM>, but different in 2 ways:
     - add not only the Consistent Title but also the English title into the Refined Query
     - use Okapi Model instead of Language Model
6. Run ID: HypDuTtlWithIdfUpperBoundBM
   - the same as <HypDuTtlNoIdfUpperBoundBM>, but also filter the Original Query and the Refined Query using IDF upper bound 3.2

[2] http://nlp.stanford.edu/software/tagger.shtml

Each patent document has a kind code. The code can be A1, A2, A3, ... or B1, B2, ... The A* kind codes means that patent documents are published during the patent application phase, while the B* kind codes means that patent documents are published during the granting phase. In our runs, we filtered the retrieval list to make sure that only A1, A2 and A* type patent documents will show in the final result list.

The top run in the English sub task comes from our submission. Table 4 lists the official evaluation results from CLEF-IP 2013. The results show that our approaches of generating highly representative queries are effective. The results also show that the Okapi BM25 retrieval model outperforms the Language Modeling retrieval model in the Novelty Search Task.

Table 2. Evaluation Result for our runs & the statistical data for all 19 submitted runs in CLEF-IP 2013 Passage Retrieval starting from Claims - English Task

Runs	PRES@100	Recall@100	MAP@100	MAP(D)	Precision(D)
HypDuTtlWithIdfUpperBoundBM	0.433	0.540	0.191	0.132	0.213
HypDuTtlNoIdfUpperBoundBM	0.432	0.540	0.190	0.132	0.214
HypCoTtlWithIdfUpperBoundLM	0.403	0.497	0.167	0.132	0.210
HypCoTtlNoIdfUpperBoundLM	0.391	0.486	0.165	0.125	0.201
OnlyClaimLM	0.356	0.453	0.147	0.120	0.171
coOnlyTtlLM	0.132	0.198	0.064	0.038	0.092
best score	0.433	0.540	0.191	0.142	0.214
median	0.396	0.488	0.166	0.038	0.092
mean	0.357	0.444	0.146	0.064	0.081

Our experiments show that in the Novelty Search Task, using long queries helps to find more relevant patent documents than using short queries, e.g. only using document titles. Hence although titles and hyphenating phrases are good resources, we also added claim texts as one resource to extract qualified query keywords. But very long queries always contain noise, in our best run, our strategy to balance this is to control the query keywords in less than 20 words. Specifically, after filtering qualified terms from titles, hyphenating phrases and claims texts by using IDF and POS tagging, if the query keywords coming from titles and hyphenating phrases are less than 20 words, we ranked query keywords from claims texts by IDF score and added the top ranked terms into the query keywords set until it contains 20 words.

All the approaches we talked above are about document level retrieval. At passage retrieval level, in all runs, we used the sum of query keywords' $TF \times IDF$ score to rank passages, where TF is the query keyword's term frequency in a passage and IDF is its corpus inverse document frequency. Only the top 10 ranked passages will be returned. Our approaches have good passage MAP and Precision scores which proves that our approach is effective.

In summary, our approach is highly effective in finding Patent Prior Arts written in English, as well as effective in finding Patent Prior Arts written in German or French.

5 Conclusion

CLEF-IP 2013 Passage Retrieval starting from Claims task precisely captures the procedure of the Prior Art Search. Participants are given one or a few claims, and are asked to retrieve relevant patent documents in the collection and mark out the relevant passages.

In this paper we present an integrated process of Prior Art Search.We focus on formulating representative queries from various metadata that is embedded in a patent document. We then submit the queries to the Lemur search engine to perform document level retrieval. We mainly used two retrieval algorithms, Language Modeling and Okapi BM25. We tuned the parameters for CLEF-IP 2012 dataset. We believe that the parameter setting we list in Section 2 are also suitable for other Patent collections. Moreover, in the paper, we present six approaches to formulate queries from patent documents. The experimental results from CLEF-IP 2012 and 2013 both support that our approaches are effective to identify representative query keywords.

References

1. W. B. Croft, D. Metzler, and T. Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010.
2. R. Krovetz. Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–202. ACM, 1993.
3. F. Piroi, M. Lupu, A. Hanbury, A. P. Sexton, W. Magdy, and I. V. Filippov. Clef-ip 2012: Retrieval experiments in the intellectual property domain. In *CLEF-IP '12*.
4. A. Ratnaparkhi et al. A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on empirical methods in natural language processing*, volume 1, pages 133–142. Philadelphia, PA, 1996.
5. P. Treeratpituk and J. Callan. Automatically labeling hierarchical clusters. In *Proceedings of the 2006 international conference on Digital government research*, pages 167–176. Digital Government Society of North America, 2006.