Formulating Good Queries for Prior Art Search

José Carlos Toucedo and David E. Losada

Grupo de Sistemas Inteligentes
Dept. Electrónica y Computación
Universidad de Santiago de Compostela, Spain
{josecarlos.toucedo,david.losada}@usc.es

Abstract. In this paper we describe our participation in CLEF-IP 2009 (prior art search task). This was the first year of the task and we focused on how to build effectively a prior art query from a patent. Basically, we implemented simple strategies to extract terms from some textual fields of the patent documents and gave more weight to title terms. We ran experiments with the well-known BM25 model. Although we paid little attention to language-dependent issues, our performance was usually among the top 3 groups participating in the task.

Key words: Patent Retrieval, Prior Art Search, Query Formulation.

1 Introduction

The main task of the CLEF-IP09 track is to investigate information retrieval techniques for patent retrieval, specifically for *prior art search*. Prior art search, which consists of retrieving any prior record with identical or similar contents to a given patent, is the most common type of retrieval in the patent domain.

The track provides the participants with a huge collection of more than one million patents from the European Patent Office (EPO). Every patent in the collection consists of several XML documents (generated at different stages of the patent's life-cycle) that can be written in French, English or German[1].

In an information retrieval setting the patent to be evaluated can be regarded as the information need and all the patent documents (granted patents and applications) filed prior to the topic application as the document collection. However, the query patent provided is a very long document which contains many ambiguous and vague terms [1]. Therefore, this year our main objective has been to formulate a concise query that effectively represents the underlying information need.

The rest of the paper is organized as follows. Section 2 describes the approach we have taken, specifically how the query is built and what experiments we designed; the runs we submitted are explained in Sect. 3 and the results are analysed in Sect. 4. Finally, in Sect. 5 we expose our conclusions and discuss future work.

[1] Further information is available in [6].

2 Approach Taken

The track requires that retrieval is performed at patent level but provides several documents per patent. We decided to work with an index built at document level and then post-process the result in order to obtain a ranking of patents (each patent receives the score of its highest ranked document). This follows the intuition that the patent document that is the most similar to the query patent reflects well the connection between the query and the underlying patent.

The index we used[2] was built from all the textual fields of a query patent, i.e. invention-title, abstract, description and claims. Although the documents contain terms from three different languages, no language-oriented distinction was made at index construction time. This means that the index contains all terms in any language for each patent document. Furthermore, stemming was not applied and an English stopword list (with 733 stopwords) was used in order to remove common words. This makes sense because almost 70% of the data was written in English.

2.1 Query Formulation

A query patent contains about 7500 terms on average and, therefore, using them all would yield to high query response times. Furthermore, there are many noisy terms in the document that might harm performance. A good processing of the query patent document is a key factor in order to achieve good effectiveness.

Our experiments focused on extracting the most significant terms from the query patent, i.e. those terms that are discriminative. To this aim, we used *inverse document frequency* (idf). In our training, we concentrated on deciding the number of terms that should be included into the query. We ran this process in both a language-independent and language-dependent way (i.e. a single ranking of terms vs. three rankings of terms, one for each language).

The number of query terms is difficult to set because few query terms make that the query processing is fast but the information need might be misrepresented; on the other hand, if many terms are taken the query will contain many noisy terms and the query processing time might be prohibitive. We have studied two methods to choose a suitable number of terms: (i) establishing a fixed number of terms for all queries and (ii) establishing a fixed percentage of the query patent length. Observe that those terms that appear several times in a query patent have been considered only once in the final selection. Because of this, both the number of terms to extract and the query patent length refer to the number of unique terms.

Once the number of query terms has been selected, we must determine how they are extracted. We explored two strategies: language-independent and language-dependent. Suppose that we select n terms from the original query

[2] We deeply thank the support of Erik Graf and Leif Azzopardi, from University of Glasgow, who granted us access to their indexes [2].

patent regardless of the language. This means that all query patent terms (English, French and German terms) are ranked together and we simply select the n terms with the highest *idf* from this list. Because of the nature of the languages, it is likely that the three languages present different *idf* patterns. Besides, there are fewer German/French documents than English documents in the collection and, therefore, this introduces a bias in terms of *idf*. We therefore felt that we needed to test other alternatives for selecting terms. We tried out an extraction of terms where each language contributes with the same number of terms. In this second strategy we first grouped the terms of a query patent depending on their language (no classification was needed since every field in the XML is tagged with language information). Next, the highest $n' = \lfloor n/3 \rfloor$ terms from each group are extracted. The query is finally obtained by compiling the terms from the three groups.

In Sect. 2.3 we will explain how different configurations combining these strategies behave in terms of performance.

2.2 Retrieval Model

Initially, we used the well-known BM25 retrieval model [5] with the usual parameters ($b = 0.75$, $k_1 = 1.2$, $k_3 = 1000$). However, as shown below, we also tested several variations for b and k_1 in the submitted runs (in order to check the stability of the model w.r.t. the parameter setting).

The platform under which we executed our experiments was the Lemur Toolkit[3]. All experiments were run in the Large Data Collider (LDC), a supercomputer offered by the Information Retrieval Facility (IRF). This system, with 80 Itanium processors and 320GB of random access memory, provides a suitable environment for large-scale experiments.

2.3 Training

With the training data provided by the track, we studied two dimensions: query length and language. Query length refers to the way in which query size is set. As argued above, this can be done in a query-dependent (i.e. a given percentage of the query patent terms are selected) or query-independent way (i.e. a fixed number of terms are selected for all queries). The language dimension reflects the way in which terms are ranked (language-independent, i.e. a single rank for all terms; language-dependent, one rank of terms for every language). Hence, our training consisted of studying how the four combinations of these dimensions perform in terms of three well-known performance measures: MAP, Bpref and P10.

The results were obtained with the large training set (500 queries) of the main task, which contains queries in the three languages. We used the usual parameters for the BM25 retrieval model.

[3] `http://www.lemurproject.org/`

Figures 1(a), 2(a) and 3(a) report results for the case where the number of terms is fixed for all queries. Surprisingly, we get better performance when the language is not taken into account. However, Figs. 1(b), 2(b) and 3(b), where terms are selected using a percentage of the query length, show a different trend. Figures 1(b) and 3(b) show that no significant difference can be established in terms of MAP and P10, respectively. In contrast, Fig. 2(b) shows that the language-dependent choice is slightly more consistent than the language-independent one in terms of Bpref.

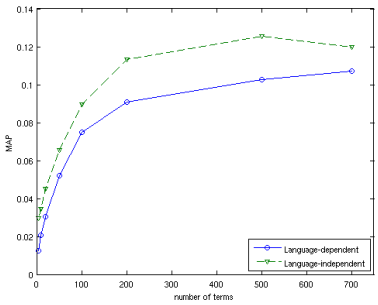
Summing up, there are two competing configurations that perform well: a) the model that consists of combining the query-dependent and language-dependent strategies, and b) the model that considers the query-independent and language-independent strategies together. If we observe carefully the plots we will note that these two models do not differ much in MAP and P10 values but, in terms of Bpref, the model that is language and query-dependent performs the best. Furthermore, according to this training, we can state that a 40% of query length is a good trade-off between performance and efficiency. We therefore fixed the query-dependent and language-dependent as our reference model.

To further check that our final query production strategy is actually selecting good terms, we compared it against a baseline method. The baseline we used consists of the same retrieval system with no query formulation strategy. In this case, the queries were built by appending all the textual fields of the patents (invention-title, abstract, description and claims). This leads to very long queries with no term selection (and many terms appear more than once). Table 1 shows that our approach outperforms the baseline for each measure. So, we achieve better performance and, additionally, the query response time is expected to be significantly lower by selecting those terms that we consider more important.

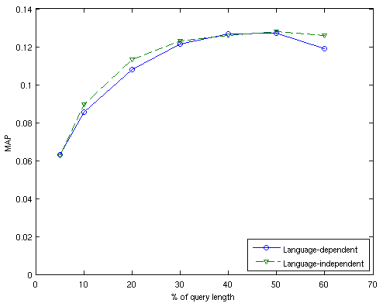
Table 1. Query formulation improvement over a baseline with no term selection.

	avg(#terms/query)	MAP	BPREF	P10
baseline	5656.270	.1077	.4139	.0834
query formulation	439.128	**.1269**	**.5502**	**.1012**
Δ%		+17,83%	+32,93%	+21,34%

BM25 Parameters. In parallel, we performed some experiments to tune the BM25 parameters b and k_1. To this aim, we chose the small training set (5 queries) with the query-independent (500 terms) and the language-independent strategies. First, we tried several values for b keeping the other parameters fixed ($k_1 = 1.2$, $k_3 = 1000$). The observed results are described in Table 2. On the other hand, we studied the effect of the k_1 parameter for two different values of b: the recommended one ($b = 0.75$) and the value yielding the best MAP

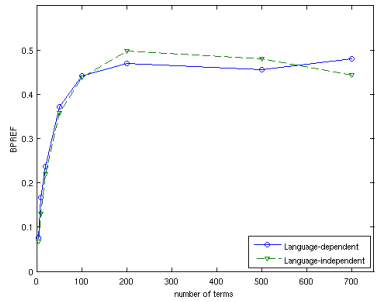


(a) Query-independent experiments

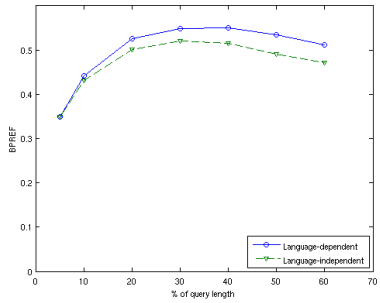


(b) Query-dependent experiments

Fig. 1. MAP performance

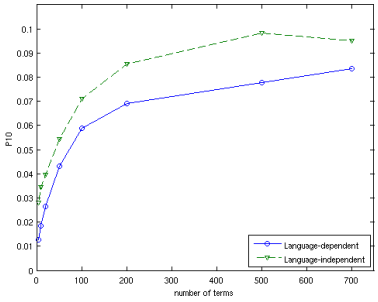


(a) Query-independent experiments

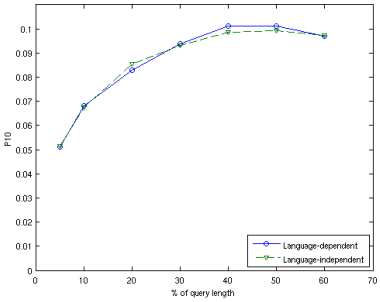


(b) Query-dependent experiments

Fig. 2. BPREF performance



(a) Query-independent experiments



(b) Query-dependent experiments

Fig. 3. P10 performance

performance ($b = 1$). Again, we used $k_3 = 1000$. Tables 3 and 4 report the results.

Table 2. b tuning.

b	MAP	BPREF	P10
0.1	.0708	.6184	.0600
0.2	.0952	**.6302**	.0800
0.3	.1071	.6265	.0800
0.4	.1129	.6229	.1200
0.5	.1397	.6193	**.1400**
0.6	.1422	.6120	**.1400**
0.7	.1470	.5995	**.1400**
0.8	.1445	.5995	**.1400**
0.9	.1442	.5922	**.1400**
1.0	**.1529**	.6047	.1200

Table 3. k_1 tuning ($b = 0.75$).

k_1	MAP	BPREF	P10
0.2	.1020	**.6302**	.1000
0.4	.1120	.6265	.1000
0.6	.1154	.6229	.1200
0.8	.1408	.6120	**.1400**
1.0	.1400	.6120	**.1400**
1.2	.1470	.5995	**.1400**
1.4	.1465	.5959	**.1400**
1.6	**.1591**	.5922	**.1400**
1.8	.1555	.6047	**.1400**
2.0	.1513	.6047	.1200

Table 4. k_1 tuning ($b = 1$).

k_1	MAP	BPREF	P10
0.2	.1067	**.6302**	.1000
0.4	.1130	.6229	.1200
0.6	.1412	.6120	**.1600**
0.8	.1459	.5995	.1400
1.0	.1446	.5995	.1400
1.2	.1529	.6047	.1200
1.4	**.1532**	.6172	.1400
1.6	.1471	.6136	.1400
1.8	.1449	.6099	.1200
2.0	.1445	.6027	.1200

According to this data, if we want to promote BPREF measure we should choose low values for both b and k_1. MAP, however, reaches its best performance at different places, specifically for $b = 0.75$ and $k_1 = 1.6$.

3 Submitted Runs

We participated in the *Main* task of this track with eight runs for the *Small* set of topics, which contains 500 queries in different languages.

First, we submitted four runs considering the scenario that best worked for our training experiments. These four runs differ on the retrieval model parameters. We included the recommended BM25 configuration but also tried out some variations in order to incorporate the trends that were detected in Sect. 2.3: *uscom_BM25a* ($b = 0.2$, $k_1 = 0.1$, $k_3 = 1000$), *uscom_BM25b* ($b = 0.75$, $k_1 = 1.2$, $k_3 = 1000$), *uscom_BM25c* ($b = 0.75$, $k_1 = 1.6$, $k_3 = 1000$) and *uscom_BM25d* ($b = 1$, $k_1 = 1.2$, $k_3 = 1000$).

Furthermore, we submitted four additional runs where the queries were expanded with the title terms of the query patent. In this way, the query term frequency of these terms is augmented and the presence of the title terms in the final queries is guaranteed. These new runs are labeled as the previous ones plus an extra "t".

4 Results

The official evaluation results of our submitted runs are summarized in Table 5. For each run and measure, we show both the value we got and its position among the 48 runs submitted by all groups.

The first conclusion we can extract from the evaluation is that our decision to force the presence of title terms worked well. Regardless of the configuration

Table 5. Submitted runs for CLEF-IP 09

	P	P5	P10	P100	R	R5	R10	R100	MAP	nDCG
uscom_BM25a	.0029	.0948	.0644	.0141	.4247	.0900	.1183	.2473	.0837	.4466
	#36	#31	#32	#38	#39	#30	#32	#36	#30	#12
uscom_BM25b	.0041	.1184	.0858	.0205	.5553	.1082	.1569	.3509	.1079	.4410
	#13	#11	#6	#12	#22	#12	#6	#11	#7	#15
uscom_BM25c	.0042	.1180	.0858	.0206	.5563	.1104	.1564	.3504	.1071	.4341
	#9	#12	#7	#10	#20	#8	#7	#13	#9	#20
uscom_BM25d	.0042	.1188	.0852	.0206	.5630	.1113	.1558	.3500	.1071	.4346
	#10	#10	#9	#11	#18	#6	#8	#14	#10	#18
uscom_BM25at	.0031	.1004	.0680	.0151	.4549	.0937	.1223	.2637	.0867	.4331
	#32	#25	#31	#35	#35	#22	#30	#34	#26	#21
uscom_BM25bt	.0042	**.1280**	**.0908**	**.0213**	.5729	**.1176**	**.1631**	**.3610**	**.1133**	**.4588**
	#11	#3	#3	#4	#15	#2	#2	#5	#3	#6
uscom_BM25ct	.0042	.1268	.0898	.0212	.5722	.1172	.1611	.3602	.1132	.4544
	#12	#4	#4	#6	#16	#3	#3	#7	#4	#8
uscom_BM25dt	**.0043**	.1252	.0892	**.0213**	**.5773**	.1163	.1606	.3609	.1121	.4455
	#8	#5	#5	#5	#14	#4	#4	#6	#5	#13

of the BM25 parameters, the run with the title terms always obtains better performance than its counterpart.

Furthermore, among the configurations with the title terms the best run is the one labeled as *uscom_BM25bt*. This run corresponds to the usual parameters of the BM25 retrieval model, i.e. $b = 0.75$, $k_1 = 1.2$, $k_3 = 1000$.

On the other hand, Table 6 shows another view on how good our runs performed in the evaluation. For each measure, we compare our best run with the best run and the median run in the track.

Table 6. Comparative results for CLEF-IP 09

	P	P5	P10	P100	R	R5	R10	R100	MAP	nDCG
best run	.0431	.2780	.1768	.0317	.7588	.2751	.3411	.5800	.2714	.5754
our best run	.0043	.1280	.0908	.0213	.5773	.1176	.1631	.3610	.1133	.4588
	#8	#3	#3	#4	#14	#2	#2	#5	#3	#6
median run	.0034	.1006	.0734	.01785	.53535	.09275	.1309	.30855	.0887	.42485

Note that our run is ranked third for the reference measures P10 and MAP. However, in absolute terms, our results are not comparable to the results achieved by the best run. Actually, regardless of the measure, there is always a large gap between the top 1 run and the remaining ones. The first position was recurrently ocuppied by the team from Humboldt University, with their *humb_1* run [4]. Among the techniques they applied we can highlight the usage of multiple retrieval models and term index definitions, the merging of different results (and the posterior re-ranking) based on regression models and the exploitation of patent metadata[4] for creating working sets. They used prior art information from the *description* field (patents explicitly cited) as the seed of an iterative

[4] The problem of comparing results based on text retrieval and re-ranking and filtering methods based on utilization of meta-data has been also outlined in [3].

process for producing the working set. In the future, it would be interesting to compare the systems according to how good they retrieve *hidden* patents (not explicitly mentioned in the *description*).

5 Conclusions and Future Work

We have designed a query production method that outperforms a baseline with no query formulation and ranked among the top three systems for most performance measures. This method selects a number of terms that depends on the length of the original query and forces a fixed number of terms per language.

The original query patent has much noise that adversely affects retrieval performance. An appropriate method for estimating the importance of the terms should be designed and applied to the patent query in order to remove noise. Nevertheless, prior art search is a recall-oriented task and reducing the query too much may harm recall.

This was our first participation in CLEF and we did not pay much attention to the cross-language retrieval problem. In the near future, we want to conduct research in this direction. We will study how to separate the patent contents by language, maintaining different indexes, etc. Furthermore, we would like to experiment with link analysis, entity extraction and structured retrieval.

Acknowledgements We are deeply grateful to Erik Graf and Leif Azzopardi, from University of Glasgow, for their help during our experiments. We also thank the support of the IRF. This research was co-funded by FEDER and *Xunta de Galicia* under projects 07SIN005206PR and 2008/068.

References

1. Graf, E., Azzopardi, L.: A methodology for building a patent test collection for prior art search. In: Proceedings of the Second International Workshop on Evaluating Information Access (EVIA) (2008)
2. Graf, E., Azzopardi, L., van Rijsbergen, K.: Automatically generating queries for prior art search. Working notes for the CLEF 2009 Workshop
3. Kando, N.: Overview of the fifth NTCIR workshop. In: Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access (2005)
4. Lopez, P., Romary, L.: Multiple retrieval models and regression models for prior art search. Working notes for the CLEF 2009 Workshop
5. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. pp. 109–126 (1996)
6. Roda, G., Tait, J., Piroi, F., Zenz, V.: CLEF-IP 2009: Retrieval experiments in the intellectual property domain. In: Multilingual Information Access Evaluation Vol. I Text Retrieval Experiments (2010)