A Study of Query Reformulation Methods for Patent Prior Art Search with Partial Patent Applications

No Author Given

No Institute Given

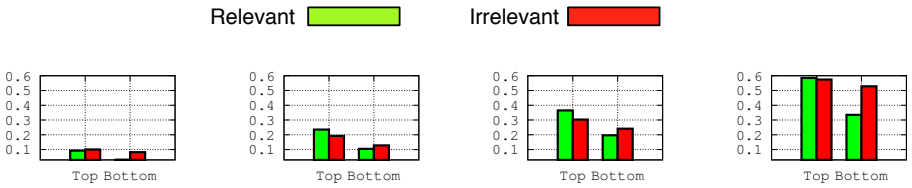
Abstract. Patents are used by entities to legally protect their inventions and represent a multi-billion dollar industry of licensing and litigation. In 2012, 276,788 patent applications were approved in the US alone – a number that has doubled in the past 15 years. While much of the literature inspired by the evaluation framework of the CLEF-IP competition has aimed to assist patent examiners in assessing prior art for complete patent applications, less of this work has focused on patent search with queries representing (partial) applications to help inventors to assess the patentability of their ideas prior to writing a full application. Hence, in this context, a query is much longer than in other standard IR tasks. It can take the form of a long paragraph, or even a very long document. This has led the research to focus on query reformulation for patent search. Therefore, in this paper, we carry out an intensive study and evaluation of both patent specific and standard query reformulation methods for patent prior art search with partial patent applications. We intend to mainly answer the following questions: *What are these query reformulation methods? How do they work? What is the best section in a patent application to use as a query? What is the best query reformulation method?*

Keywords: Query Reformulation, Patent Search, Experimentation.

1 Introduction

Patents are used by entities to legally protect their inventions and represent a multi-billion dollar industry of licensing and litigation. In 2012, 276,788 patent applications were approved in the US alone, a number that has doubled in the past 15 years. Hence, helping both inventors and patent examiners assess the patentability of a given patent application through a patent prior art search is a critical task.

Patent prior art search involves finding previously granted patents that may be relevant to a new patent application. The objective and challenges of standard formulations of patent prior art search are different from those of standard text and web search since [7]: (i) queries are (partial) patent applications, which consist of documents with hundreds of words organized into several sections,



(a) Title query. (b) Abstract query. (c) Claims query. (d) Description query.

Fig. 1: Average Jaccard similarity between fields of topics and the corresponding (ir)relevant documents for different queries that perform the best/worst.

while typical queries in text and web search constitute only a few words; (ii) patent prior art search is a recall-oriented task, where the primary focus is to retrieve all relevant documents at early ranks, in contrast to text and web search that are precision-oriented, where the primary goal is to retrieve a subset of documents that satisfy the query intent. Another important characteristic about patent patent prior art search is that, in contrast to scientific and technical writers, patent writers tend to generalize and maximize the scope of what is protected by a patent using a combination of abstract and specific terminology, which make also hard to find any relevant prior work by the patent examiner.

While much of the literature inspired by the evaluation framework of the CLEF-IP competition has aimed to assist patent examiners in assessing prior art for complete patent applications, less work has focused on assessing the patentability of inventions before writing a full patent application. Prior art search with queries that represent unfinished patent applications is desirable, since writing a full application is time-consuming and costly, especially if lawyers are hired to assist.

To assess the difficulty of querying with partial patent applications, we refer to Figure 1. Here we show an analysis of the average Jaccard similarity[1] between different queries (representing the title, abstract, claims, or descriptions intended to represent a partial patent application) and the labeled relevant (all) and irrelevant documents (top 10 irrelevant documents ranked by BM25 [13]). We show results for the top 100 and bottom 100 queries (100 queries that perform the best, and 100 queries that perform the worst) of CLEP-IP 2010 evaluated according to Mean Average Precision (MAP). Note that, while the title section is usually composed by an average of six terms, the other sections are longer, ranging from ten to thousands of terms. There are three notable trends here: (i) term overlap increases from title to description since the query size grows accordingly; (ii) the bottom 100 performing queries tend to have much smaller term overlap with the relevant documents than the top 100 queries; and (iii) the the overlap of any relevant document set for any set of queries is less than one.

[1] The Jaccard similarity is used to measure the term overlap between two sets. Before applying the Jaccard similarity, patent-specific stopwords were removed, as suggested by [9].

While these results suggest the description section is the best part of a partial patent application to use as query, they also point out that the term overlap between the queries and the relevant documents can be very low. Also, in this context, a query is much longer than in other standard IR tasks. It can take the form of a long paragraph (e.g. the case of the abstract used for querying), or even a very long document (e.g. the case of claims or the description used for querying). This has led the research to focus on query reformulation for patent search. Therefore, we suggest an investigation of *query reformulation* [1] methods as a means for improving the term overlap between queries that represent partial patent applications and relevant documents, with the objective of assessing not only the performance of standard query reformulation methods, but also the effectiveness of query reformulation methods that exploit patent-specific characteristics. In summary, the contributions of this paper are the following:

1. A review of the of both patent specific and standard query reformulation methods for patent prior art search with partial patent applications.
2. A thorough comparative analysis of these query reformulation methods along several dimensions (including query type, IR model, term expansion source, etc.) on standardized datasets of CLEF-IP.

The rest of the paper is organized as follows: in Section 2 we present a number of patent specific query reformulation methods; in Section 3 we present our evaluation and the results analysis; and in Section 4 we conclude with possible directions for future work.

2 Query Reformulation for Patents

Query Reformulation is the process of transforming an initial query Q to another query Q'. This transformation may be either an expansion or a reduction of the query. *Query Expansion* (QE) [3] enhances the query with additional terms likely to occur in relevant documents. Hence, given a query representation Q, QE aims to select an optimal subset of k terms T_k relevant to Q, then build Q' such as $Q' = Q \cup T_k$. As for *Query Reduction* (QR) [5], it is the process that reduces the query such that superfluous information is removed. Hence, given a query representation Q, QR aims to select an optimal subset of k terms $T_k \subset Q$ relevant to Q, then build Q' such as $Q' = T_k$.

In the following, in Section 2.1 we first describe standard query reformulation methods, then, in Section 2.2, we describe few patent specific query reformulation methods.

2.1 Standard Query Reformulation Methods

The Rocchio Algorithm for Relevance Feedback The Rocchio algorithm [15] is a classic algorithm of relevance feedback used mainly for query expansion. Basically, it provides a way of incorporating relevance feedback information into the vector space model representing a query [11]. The underlying theory behind

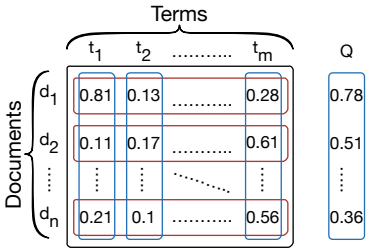


Fig. 2: Notation used in MMR QE/QR.

Rocchio is to find a query vector $\overrightarrow{Q'}$, that maximizes similarity with relevant documents while minimizing similarity with irrelevant documents. Typically, a pseudo-relevance feedback (PRF) set of k top ranked documents obtained after an initial run of the query is considered as the set of relevant documents to build $\overrightarrow{Q'}$. We refer to this method as RocchioQE[2].

On the other hand, Rocchio we can think to use Rocchio as a QR method. Basically, the idea is once we have computed the Rocchio modified query vector, we take only terms that appear in the initial query Q and rank them using the Rocchio score. Then, we select top k terms with the highest score to build Q'. We refer to this approach as RocchioQR.

Maximal Marginal Relevance for Query Reformulation As a general method for query reformulation, we also consider a method of "diverse" term selection — such as the *Maximal Marginal Relevance* (MMR) [2] algorithm for result set diversification. The idea is to use MMR for diverse term selection.

In the case of query expansion, we call this method MMR Query Expansion (MMRQE). MMRQE takes as input a PRF set, which is used to build a document-term matrix of n documents and m terms as shown in Figure 2 (the TF-IDF is used to populate the matrix for each document vector). To represent the query Q is the documents' dimension as in Figure 2, we use the BM25 or TF-IDF score between each document d_i and the query. Hence, given a query representation Q, MMRQE aims to select an optimal subset of k terms $T_k^* \subset D$ (where $|T_k^*| = k$ and $k \ll |m|$) relevant to Q but inherently different from each other (i.e., diverse). This can be achieved by building T_k^* in a greedy manner by choosing the next optimal term t_k^* given the previous set of optimal term selections $T_{k-1}^* = \{t_1^*, \ldots, t_{k-1}^*\}$ (assuming $T_0^* = \emptyset$) using the MMR diverse selection criterion.

Similarly, we can imagine greedily rebuild the query from scratch, while choosing diversified terms (i.e. terms of the query). Here, we call this approach MMR Query Reduction (MMRQR). Formally, given a query representation Q, we aim to select an optimal subset of k terms $T_k^* \subset Q$ (where $|T_k^*| = k$ and

[2] We used the LucQE module, which provides an implementation of the Rocchio method for Lucene. `http://lucene-qe.sourceforge.net/`

$k < |Q|$) relevant to Q but inherently different from each other (i.e., diverse). This can be achieved by building T_k^* in a greedy manner by choosing the next optimal term t_k^* given the previous set of optimal term selections $T_{k-1}^* = \{t_1^*, \ldots, t_{k-1}^*\}$ (assuming $T_0^* = \emptyset$) using an adaptation of the MMR diverse selection criterion. Note that the we used all the sections of the patent documents of the PRF set to built the document-term matrix of n documents and m terms shown in Figure 2.

2.2 Patent Specific Query Reformulation Methods

In this section we first describe two patent specific query expansion methods, then, we describe two patent specific query reduction methods.

Synonyms Sets for Patent Query Expansion Magdy et al. [8] proposed a patent query expansion method, which automatically generates candidate synonyms sets (SynSet) for terms, and use it as a source of expansion terms. The idea for generating the SynSet comes from the characteristics of the CLEF-IP patent collection, where some of the sections in some patents are translated into three languages (English, French, and German). The idea is to use these parallel manual translations to create possible synonyms sets. Hence, for a word w in one language which has possible translations to a set of words in another language $w_1, w_2, \ldots, w_n$, this set of words can be considered as synonyms or at least related to each other. The generated SynSet is used for query expansion in two ways: (i) The first one use the probability associated with the SynSet entries as a weight for each expanded term in the query (denoted WSynSet). Therefore, each term was replaced with its SynSet entries with the probability of each item in the SynSet acting as a weight to the term within the query. (ii) The second one neglected this associated probability and used uniform weighting for all synonyms of a given term (denoted USynSet). This strategy is similar to adding synonyms from WordNet where no probability is assigned.

Patent Lexicon for Query Reformulation Mahdabi et al. [10] proposed to build a query-specific patent lexicon based on definitions of the International Patent Classification (IPC). The lexicon is simply build by removing general and patent stop-words from the text of IPC definition pages. Each entry in our lexicon is composed of a key and a value. The key is an IPC class and the value is a set of terms representing the mentioned class. Then, the lexicon build is used to extract expansion concepts related to the context of the information need of a given query patent. To this end, the IPC class of the query patent is searched in the lexicon and the terms matching this class are considered as candidate expansion terms. The approach proposed tries to combine these two complementary vocabularies. In this paper we refer to this patent query expansion method as IPC Codes.

Language Model for Query Reduction In [4], the authors proposed a query reduction technique, which decomposes a query (a patent section) into con-

stituent text segments and computes a Language Modeling (LM) similarities by calculating the probability of generating each segment from the top ranked documents (PRF set). Then, the query is reduced by removing the least similar segments from the query. We refer to this method by LMQR.

IPC Codes for Query Reduction Based on the intuition that, terms in the IPC code definition may represent "stop-words", especially if they are rare (infrequent in the patent application), one can think to reduce a patent query as follows: (i) For each patent application, take the definitions of the IPC codes which are associated to it. Then, (ii) rank the terms of the query according to both their frequency in the class code definition, and their frequency in the query. Finally, (iii) remove bottom terms of this ranking from the query (i.e. good terms are terms that occur a lot in the query, and few in the class code definition, whereas bad terms are those that occur few in the query, and a lot in the class code definition). In the evaluation section we denote this approach IPC Codes.

3 Experimental Evaluation

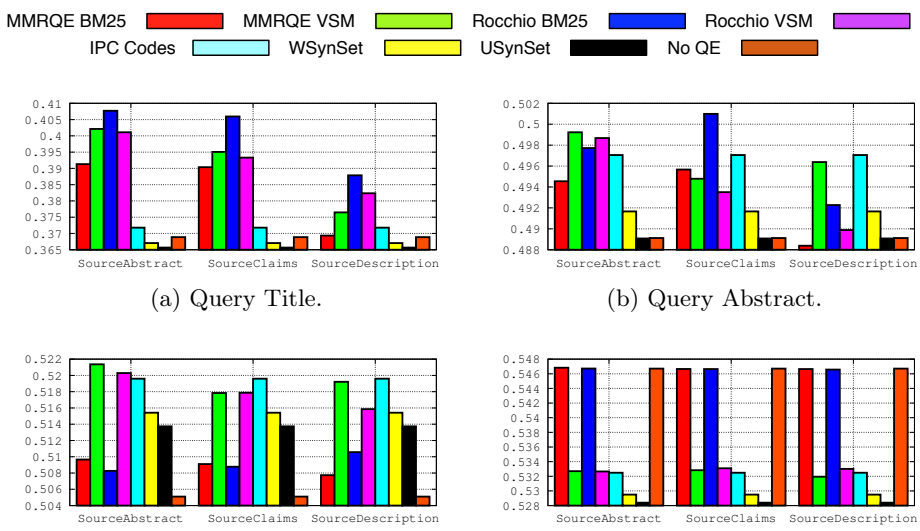
In this section we first explain the experiments setup for evaluating the effectiveness of the different methods described above. Then, we discuss the results of QE and QR methods in Sections 3.2 and 3.3 respectively.

3.1 Experimental Setup

For our experiments we used used the Lucene IR System[3] to index the English subset of CLEF-IP 2010 and CLEF-IP 2010 datasets[4] [12,14] with the default stemming and stop-word removal. We removed patent-specific stop-words as described in [7]. CLEF-IP 2010 contains 2.6 million patent documents and CLEF-IP 2011 consists of 3 million patent documents. The English test sets of CLEF-IP 2010 and CLEF-IP 2011 correspond to 1303 and 1351 topics respectively. In our implementation, each section of a patent (title, abstract, claims, and description) is indexed in a separate field, so that different sections can be used, for example, as source of expansion terms. But, when a query is processed, all fields in the index are targeted, since it is sensible to use all available content.

We also used the patent classification (IPC) for filtering the results by constraining them to have common classifications with the patent topic as suggested in previous works [6,14]. Finally, we report MAP, and PRES, which combines Recall with the quality of ranking and weights relevant documents lower in the ranking more highly than MAP. We report the evaluation metrics on the top 1000 results.

[3] `http://lucene.apache.org/`
[4] `http://www.ifs.tuwien.ac.at/~clef-ip/`



(a) Query Title.

(b) Query Abstract.

(c) Query Claims.

(d) Query Description.

Fig. 3: PRES for QE methods on CLEF-IP 2010.

3.2 Query Expansion Results

In this section, we discuss the results of the evaluation performed on the QE methods described in Section 2. During the exploration of query expansion for patent search with partial patent applications, there are many configuration options and associated questions that we can consider:

- **Partial patent query type:** We consider that a query of a partial patent application consist of either the title, the abstract, the claims or the description section. Critical questions are: what part of a partial application an inventor should write to obtain the best search results? and what part of a partial application suits the best for QE?
- **Query expansion source:** We consider the abstract, claims, and description sections as different term sources to determine which section offers the best source of expansion terms, e.g., are the claims words of particularly high value as expansion terms? Note that we consider that there is no interest to use the title as source for the expansion since there is only few terms that we can collect from the title field in the PRF set.
- **Relevance model:** For initial retrieval of documents in the *pseudo-relevant* feedback set (PRF) and subsequent re-retrieval, there are various options for the relevance ranking model. In this work, we explore a probabilistic approach represented by the popular BM25 [13] algorithm, as well as a vector space model (VSM) approach, TF-IDF [16]. A natural question is which relevance model works best for query expansion for patent prior art search?

– **Term selection method:** We consider the different query expansion methods described above, i.e. RocchioQE, MMRQE, IPC Codes, WSynSet, USynSet. What is the best QE method for patent search?

To summarize all the results obtained over all the above configurations, Figure 3 shows the PRES obtained for all the QE methods, while selecting the optimal number of terms used for the expansion (number of terms that maximizes the performance for each method). From these results, we make the following observations:

1. The best section to use for querying is the description section (see Figure 3d). We attribute this to the fact that the description section has more content along with relevant terms that define the invention since a detailed summary of the invention is described therein.
2. The best source for query expansion is the claims section. We attribute this to the fact that, the claims contain not only relevant, but also, specific terminology, since the scope of the invention is described therein. However, when querying using the claims, other sources of query expansion provided better performance. This may be because claims are very similar between them and contained specific terms; consequently, the queries lack of diversity and general terms or synonyms that are used to describe similar inventions.
3. The description section is not either a good source for expansion, since its content is too broad, therefore, it contains many irrelevant terms that hurt the performance.
4. Query expansion is not useful for very long queries (i.e. description), indicating that in advanced stages of the patent application process, QE is not relevant.
5. When dealing with more mid-long queries such as abstract or claims, MMRQE is more effective than Rocchio, which suggest that diverse term selection is not crucial for short queries.
6. Using the IPC code definitions (as suggested by [10]) and SynSet (method of [8]) as a source of expansion, gave poor performance (see IPC Codes and SynSet bars along the Figures).
7. Finally, regarding the best term selection method, we conclude that in general, MMRQE provides the best performance, followed by RocchioQE.

To give an insight of the effect of MMRQE and Rocchio over the performance, Table 1 shows two queries where QE methods improved the performance. First of all, it is interesting to notice that even if there are common terms selected to expand the queries by both MMRQE and Rocchio, the lists of MMRQE contain more diversified terms (at least in the first example). For the first example, relevant patents talk about a similar idea than the application, but using different complex and ambiguous terms. Hence, for the first query, key terms like: *rotor, blend, and suction*, were able to capture the scope of the relevant patents to allow either retrieving them (improving PRES), or pushing them to the top of the ranking (improving MAP). As for the second query, MMRQE expand the query with general terms, e.g. *result, includ, extend, plural*, which probably encourage retrieving irrelevant patents.

Table 1: Samples of queries extracted from CLEF-IP 2011, where QE improves the performance (P: Precision, R: Recall, AP: Average Precision, PRES: Patent Retrieval Evaluation Score).

1- Topic: EP-1921264-A2									
Abstract: An article of manufacture having a nominal profile substantially in accordance with Cartesian coordinate values of X, Y and Z set forth...									
Baseline performance:	**P@5:**	0.000	**P@10:**	0.000	**R@10:**	0.000	**AP:**	0.043	**PRES:** 0.777
MMRQE expanded terms: airfoil, rotor, blend, substanti, root, portion, includ, suction, form, tip									
MMRQE performance:	**P@5:**	0.000	**P@10:**	0.200	**R@10:**	0.666	**AP:**	0.124	**PRES:** 0.872
Rocchio expanded terms: airfoil, trail, edg, cool, form, blade, side, portion, root, lead									
Rocchio performance:	**P@5:**	0.000	**P@10:**	0.100	**R@10:**	0.333	**AP:**	0.100	**PRES:** 0.822
3- Topic: EP-1754935-A1									
Abstract: The fire-rated recessed downlight includes a mantle. A radiating mouth (4) is defined in the mantle. A dilatable fireproof piece (5) is fixed in the radiating mouth (4). Radiating apertures (6 or 6') corresponding to...									
Baseline performance:	**P@5:**	0.200	**P@10:**	0.100	**R@10:**	0.111	**AP:**	0.086	**PRES:** 0.801
MMRQE expanded terms: mmateri, adapt, 2, hous, light, compris, result, form, support, includ, side, mount, 4, 3, 5, plural, fit, 1, extend, recess									
MMRQE performance:	**P@5:**	0.000	**P@10:**	0.100	**R@10:**	0.111	**AP:**	0.044	**PRES:** 0.767
Rocchio expanded terms:materi, 2, compris, light, adapt, support, form, 3, 1, surfac, 5, 4, side, recess, hous, fire, 10, mount, resist, wall									
Rocchio performance:	**P@5:**	0.400	**P@10:**	0.200	**R@10:**	0.222	**AP:**	0.146	**PRES:** 0.821

3.3 Query Reduction Results

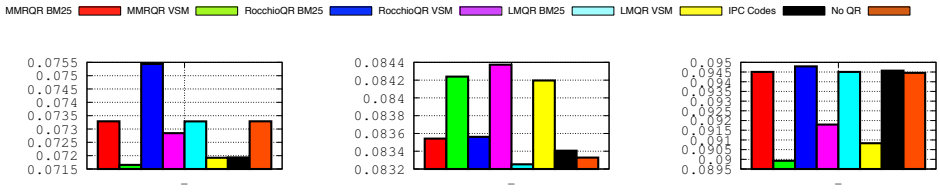
In this section, we discuss the results of the evaluation performed on the QR methods described in Section 2. Similarly, we carry out comprehensive experiments with the following specific options and associated questions:

- **Partial patent query type:** We apply QR methods to a query of a partial patent application, which consist of either the abstract, the claims or the description section. A critical question is what part of a partial application suits the best for QR? Note that we consider that there is no interest in reducing a title query since it contains only few terms.
- **Term selection method:** We consider the different query reduction methods described above, i.e. RocchioQR, MMRQR, LMQR, IPC Codes. What is the best QR method for patent search?

To summarize all the results obtained over all the above configurations, Figures 4, and 5 show the performance obtained for all the QR methods, when selecting the optimal number of terms removed from the original queries (number of terms removed that maximizes the performance for each method).

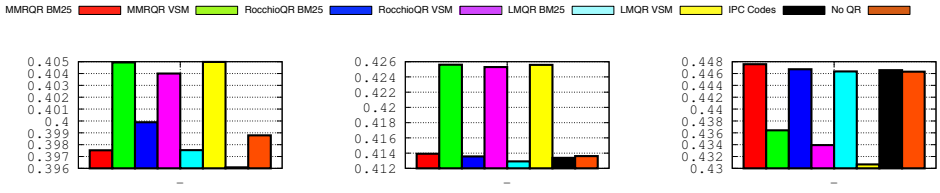
From these results, we make the following observations:

1. Query reduction is often useful for mid-long queries (i.e. abstract and claims), but not useful for very long queries since no method outperforms the baseline



(a) Query Abstract. (b) Query Claims. (c) Query Description.

Fig. 4: MAP for QR methods on CLEF-IP 2011.



(a) Query Abstract. (b) Query Claims. (c) Query Description.

Fig. 5: PRES for QR methods on CLEF 2011.

(i.e. No QR). Since many confusing terms are used in the description part, probably QR methods failed to distinguish useless terms to be removed.
2. Dealing with mid-long queries, VSM performs better than BM25, while for very long query (i.e. description), BM25 based QR methods perform better than VSM based QR methods. (Reda: Any suggestion to explain that?)
3. In general, RocchioQR and MMRQR provide better performance than the other methods.

To give an insight of the effect of MMRQR and RocchioQR over the performance, Table 2 shows some queries where QR methods are helpful. First, we notice that even if there is common terms removed from the original queries by both MMRQR and RocchioQR, the lists of MMRQR contain more similar terms (e.g. *laser, light, interferometer* for the first example). For the first example, MMRQR removed similar terms from the queries, which favor finding more diverse patent relevant to the patent applications. However, for the third query, MMRQR removed the main terms from the query (*motor*, and thermal *load*), which likely decrease the quality of the query.

4 Conclusion and Future Work

In this paper we analyzed general and specific query reformulation methods for patent prior art search for partial (incomplete) patent applications on two patent retrieval corpora, namely CLEF-IP 2010 and CLEF-IP 2011. We demonstrated that QE methods are critical for short queries, i.e. title, abstract, and claims,

Table 2: Samples of queries extracted from CLEF-IP 2011, where MMRQR and RocchioQR improve the performance. (P: Precision, R: Recall, AP: Average Precision, PRES: Patent Retrieval Evaluation Score).

1- Topic: EP-1424597-A2									
Abstract: Measurements of an interferometric measurement system are corrected for variations of atmospheric conditions such as pressure, temperature and turbulence using measurements from a second harmonic interferometer (10). A ramp, representing the dependence of...									
Baseline performance:	**P@5:**	0.000	**P@10:**	0.000	**R@10:**	0.000	**AP:**	0.022	**PRES:** 0.648
MMRQR removed terms: temperatur, **detector**, **path**, **laser**, **light**, **interferometr**, **brewster**, **sensit**, **repres**, **sourc**									
MMRQR performance:	**P@5:**	0.000	**P@10:**	0.100	**R@10:**	0.166	**AP:**	0.053	**PRES:** 0.761
RocchioQR removed terms: minim, conduct, variat, shi, turbul, condit, pressur, remov, ramp, thick									
RocchioQR performance:	**P@5:**	0.000	**P@10:**	0.000	**R@10:**	0.000	**AP:**	0.036	**PRES:** 0.724
3- Topic: EP-1314594-A1									
Abstract: An air conditioner for air conditioning the interior of a compartment includes a compressor (C) and an electric motor (84). The compressor (C) compresses refrigerant...									
Baseline performance:	**P@5:**	0.600	**P@10:**	0.400	**R@10:**	0.307	**AP:**	0.301	**PRES:** 0.777
MMRQR removed terms: refer, **motor**, **current**, **relat**, **condit**, **constant**, **suppli**, **compress**, **load**, **match**									
MMRQR performance:	**P@5:**	0.400	**P@10:**	0.500	**R@10:**	0.384	**AP:**	0.221	**PRES:** 0.774
RocchioQR removed terms: compart, **suppli**, **current**, ga, **refer**, **compress**, **relat**, interior, thermal, **match**,									
RocchioQR performance:	**P@5:**	0.400	**P@10:**	0.400	**R@10:**	0.307	**AP:**	0.266	**PRES:** 0.802

but useless for very long queries, i.e. the description section. We also showed that claims is the best section that works with QE both to query with and to use as a source of query expansion terms, suggesting that claims should be written at early stages of the patent application drafting so that they can be use to performed patent prior art search. In the same vein, we also found that the patent specific fields are more suited as a source for expansion than external sources such as synonym dictionaries. Here, future work concerns how can we exploit patent specific meta-data such as inventor and citation networks for query expansion

Regarding QR methods, we showed that these techniques are effective to some extent for claims and description sections, which are considered the longest sections in a patent application. Future work may consist of exploiting query quality predictors to identify useless terms in a query using machine learning methods.

References

1. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley Publishing Company, USA, 2nd edition, 2010.

2. J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, 1998.
3. E. N. Efthimiadis. Query expansion. *Annual Review of Inf. Systems and Technology (ARIST)*, 31:121–187, 1996.
4. D. Ganguly, J. Leveling, W. Magdy, and G. J. Jones. Patent query reduction using pseudo relevance feedback. In *CIKM*, 2011.
5. G. Kumaran and V. R. Carvalho. Reducing long queries using query quality predictors. In *SIGIR*, SIGIR '09, pages 564–571, New York, NY, USA, 2009. ACM.
6. P. Lopez and L. Romary. Patatras: retrieval model combination and regression models for prior art search. CLEF'09, pages 430–437, Berlin, Heidelberg, 2009. Springer-Verlag.
7. W. Magdy. *Toward Higher Effectiveness for Recall-Oriented Information Retrieval: A Patent Retrieval Case Study.* PhD thesis, Dublin City University School of Computing, 2012.
8. W. Magdy and G. J. Jones. A study on query expansion methods for patent retrieval. In *PaIR*, 2011.
9. P. Mahdabi, L. Andersson, M. Keikha, and F. Crestani. Automatic refinement of patent queries using concept importance predictors. In *SIGIR*, pages 505–514, New York, NY, USA, 2012. ACM.
10. P. Mahdabi, S. Gerani, J. X. Huang, and F. Crestani. Leveraging conceptual lexicon: query disambiguation using proximity information for patent retrieval. In *SIGIR*, 2013.
11. C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
12. F. Piroi, M. Lupu, A. Hanbury, and V. Zenz. Clef-ip 2011: Retrieval in the intellectual property domain. In *CLEF (Notebook Papers/Labs/Workshop)*, 2011.
13. S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-2. In *TREC*, pages 21–34, 1993.
14. G. Roda, J. Tait, F. Piroi, and V. Zenz. Clef-ip 2009: Retrieval experiments in the intellectual property domain. In C. Peters, G. Nunzio, M. Kurimo, T. Mandl, D. Mostefa, A. Penas, and G. Roda, editors, *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, volume 6241 of *Lecture Notes in Computer Science*, pages 385–409. Springer, 2009.
15. G. Salton. *The SMART Retrieval System: Experiments in Automatic Document Processing.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.
16. G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Comm. ACM*, 18(11):613–620, nov. 1975.