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. In this context, a query is much longer than in other standard IR tasks. It can take the form of a sentence (i.e. title) or several paragraphs with thousands of words (i.e. the description). This has led the research to focus on query reformulation for patent search. In this paper, we carry out an intensive study of both patent specific and standard query reformulation methods for patent prior art search with partial patent applications. We found that query expansion methods are useful for short queries (title, abstract and claims), and query reduction methods are often useful for mid-long sections (abstract and claims). We also shown that the abstract is a good section to query with, suggesting that writing the abstract is useful in the early stages of a patent drafting.

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 [8]: (i) queries are (partial) patent applications, which

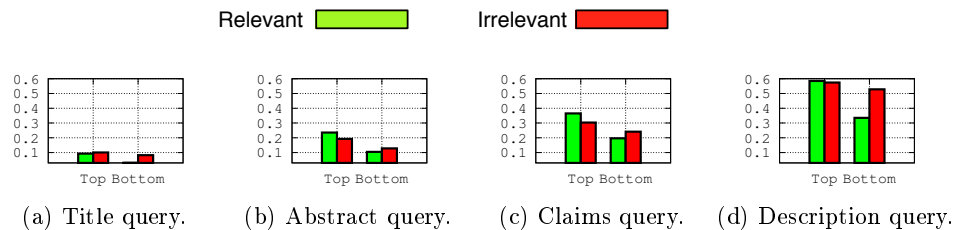(a) Title query.   (b) Abstract query.   (c) Claims query.   (d) Description query.

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

consist of documents with hundreds of words organized into several sections, 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 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 [14]). 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

---

[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 [10].

term overlap with the relevant documents than the top 100 queries; and (iii) the overlap of any relevant document set for any set of queries is less than one.

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 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 the evaluation results and 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) [4] 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 T_k of k terms, which are relevant to Q, then build Q' such as $Q' = Q \cup T_k$. As for *Query Reduction* (QR) [6], 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 $T_k \subset Q$ of k terms, which are 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 [16] is a classic algorithm of relevance feedback used mainly for query expansion.

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

Basically, it provides a way of incorporating relevance feedback information into the vector space model representing a query [12]. The underlying theory behind 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].

Similarly, Rocchio can be use as a QR method. Basically, the idea is that once the Rocchio modified query vector have been computed, it is possible to select only the terms that appear in the initial query Q and rank them using the Rocchio score and finally, select the top k terms with the highest score to built 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 — an adaptation of the *Maximal Marginal Relevance* (MMR) [3] algorithm for result set diversification. But, rather than use MMR for diverse document selection (as typically used), it is used it 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 in 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.

---

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

$$t_k^* =_{t_k \notin T_{k-1}^*} [\lambda \cos(Q, t_k) - (1 - \lambda) \max_{t_j \in T_{k-1}^*} \cos(t_j, t_k)] \tag{1}$$

Here, the first cosine similarity term measures relevance between the query Q and possible expansion term t_k while the second term penalizes the possible expansion term according to it's cosine similarity with any currently selected term in T_{k-1}^*. The parameter $\lambda \in [0, 1]$ trades off relevance and diversity. For MMRQE, we found that $\lambda = 0.5$ generally provide the best results, according to our experiments on the CLEF-IP training dataset collection.

Similarly, we can 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, MMRQR aims to select an optimal subset of k terms $T_k^* \subset Q$ (where $|T_k^*| = k$ and $k < |Q|$) 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 use all the sections of the patent documents in the PRF set to built the document-term matrix of n documents and m terms shown in Figure 2. For MMRQR, we found that $\lambda = 0.8$ generally provide the best results in our experiments on the CLEF-IP training dataset collection.

The key insight we want to highlight is that MMRQE does not select expansion terms independently as in practical usage of Rocchio, but rather it selects terms that have uncorrelated usage patterns across documents, thus hopefully encouraging diverse term selection that covers more documents for a fixed expansion budget k and ideally, higher recall.

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. [9] 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). They used 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).

Patent Lexicon for Query Expansion: Mahdabi et al. [11] 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 the 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 [5], the authors proposed a query reduction technique, which decomposes a query (a patent section) into constituent 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 as 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-StopWords.

We are aware that other patent specific methods have been explored for patent prior art search [9,2,18,11]. These approaches have not been implemented because of the following reasons: (i) their authors already pointed out about their poor performance [9,11]; (ii) throughout our experiments, we concluded that relying on other terms to form a query rather than those in the patent application, leads to poor retrieval quality [9,18]; and (iii) the method is computational too expensive [2].

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 dataset[4] [13,15] with the default stemming and stopword removal. We removed patent specific stop-words as described in [8]. CLEF-IP 2010 contains 2.6 million patent documents, and the English test sets of CLEF-IP 2010 correspond to 1303 topics. We also made the same experiments on the CLEF-IP 2011 dataset, but for lack of space we omit the results from the paper. However the obtained results was almost similar and presented the same trends.

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 [7,15]. 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.

3.2 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 are only few terms that we can collect from the title field in the PRF set.

---

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

– **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 [14] algorithm, as well as a vector space model (VSM) approach, TF-IDF [17]. 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 4 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 4d). 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) since no method outperforms the baseline. This indicates that in advanced stages of the patent preparation process, QE is not relevant (see Figure 3d and Figure 4d).

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 [11]) and SynSet (method of [9]) 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.
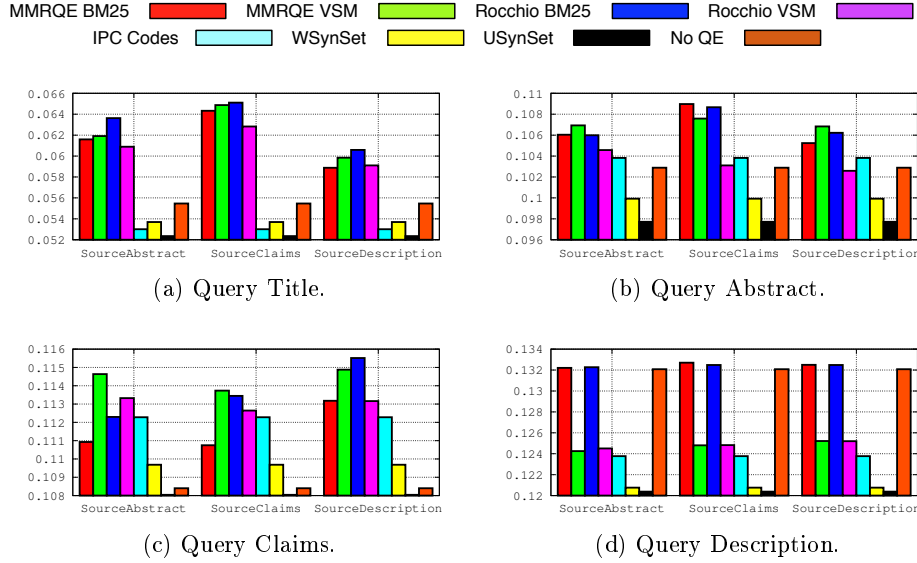
(a) Query Title.

(b) Query Abstract.

(c) Query Claims.

(d) Query Description.

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

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 the abstract, the claims or the description sections. 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-StopWords. What is the best QR method for patent search?

To summarize all the results obtained over all the above configurations, Figures 5, and 6 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 significantly (i.e. No QR). Since many confusing terms are used in the description part, probably current QR methods failed to distinguish useless terms to be removed. This can be an interesting research direction to investigate for future work.

(a) Query Title.

(b) Query Abstract.

(c) Query Claims.

(d) Query Description.

Fig. 4: PRES for QE methods on CLEF-IP 2010.

(a) Query Abstract.

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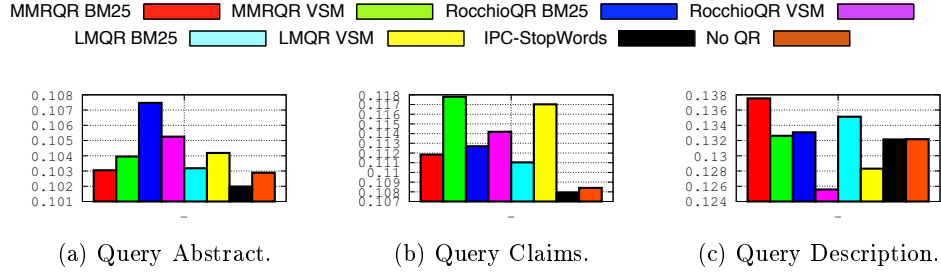
(c) Query Description.

Fig. 5: MAP for QR methods on CLEF-IP 2010.

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.

4   Conclusion and Future Work

In this paper we analyzed general and specific query reformulation methods for patent prior art search with partial (incomplete) patent applications on patent retrieval corpora of CLEF-IP. We demonstrated that QE methods are useful for short queries, i.e. title, abstract, and claims, but useless for very long queries, i.e. the description section. We also showed that the description is the best

(a) Query Abstract.   (b) Query Claims.   (c) Query Description.
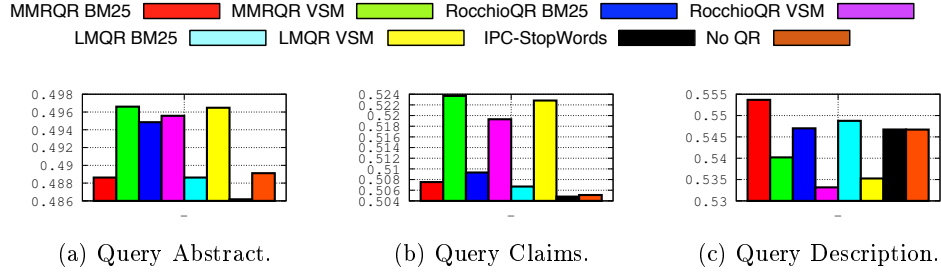
Fig. 6: PRES for QR methods on CLEF 2010.

section that works with QE to query with, followed by the claims, the abstract, then the title section. As for the source of query expansion terms, the claims seems to be the best section since it provides terms and terminology specific to the patent application domain. In the same vein, we also found that content within the patents is more suited as a source for expansion than external sources such as synonym dictionaries. For QE, future work concerns how can we exploit patent specific meta-data such as inventor and citation networks to retrieve more relevant terms to the query.

Regarding QR methods, we showed that these techniques are effective to some extent especially for the abstract and claims sections, which are considered as mid-long sections in a patent application. The improvement of QR method on the description section is not that significant. Future work may consist of exploiting query quality predictors to identify useless terms in a query using machine learning methods.

Finally, it is clear that for a patent examiner, using the description section of a patent application with perhaps a QR method like MMRQR or RocchioQR will help to find the most relevant documents (to invalidate the application). However, for inventors, before investing too much time in writing a full patent application, we believe that it is better to first write the abstract. Then, use Rocchio with expanded terms coming from the claims to get the most relevant documents to make a prior art search task (to position the current invention with related work).

References

1. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Publishing Company, USA, 2nd edition, 2010.
2. S. Bashir and A. Rauber. Improving retrievability of patents in prior-art search. In *ECIR*, 2010.
3. J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, 1998.
4. E. N. Efthimiadis. Query expansion. *Annual Review of Inf. Systems and Technology (ARIST)*, 31:121–187, 1996.

5. D. Ganguly, J. Leveling, W. Magdy, and G. J. Jones. Patent query reduction using pseudo relevance feedback. In *CIKM*, 2011.

6. 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.

7. 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.

8. W. Magdy. *Toward Higher Effectiveness for Recall-Oriented Information Retrieval: A Patent Retrieval Case Study*. PhD thesis, Dublin City University School of Computing, 2012.

9. W. Magdy and G. J. Jones. A study on query expansion methods for patent retrieval. In *PaIR*, 2011.

10. 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.

11. P. Mahdabi, S. Gerani, J. X. Huang, and F. Crestani. Leveraging conceptual lexicon: query disambiguation using proximity information for patent retrieval. In *SIGIR*, 2013.

12. C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

13. 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.

14. S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-2. In *TREC*, pages 21–34, 1993.

15. 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.

16. G. Salton. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

17. G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Comm. ACM*, 18(11):613–620, nov. 1975.

18. M. Verma and V. Varma. Patent search using ipc classification vectors. In *PaIR*, 2011.