

# Effective Query Generation and Postprocessing Strategies for Prior Art Patent Search

Suleyman Cetintas and Luo Si

Department of Computer Sciences, Purdue University, West Lafayette, IN 47907.

E-mail: {scetinta, lsi}@cs.purdue.edu

**Rapid increase in global competition demands increased protection of intellectual property rights and underlines the importance of patents as major intellectual property documents. Prior art patent search is the task of identifying related patents for a given patent file, and is an essential step in judging the validity of a patent application. This article proposes an automated query generation and postprocessing method for prior art patent search. The proposed approach first constructs structured queries by combining terms extracted from different fields of a query patent and then reranks the retrieved patents by utilizing the International Patent Classification (IPC) code similarities between the query patent and the retrieved patents along with the retrieval score. An extensive set of empirical results carried out on a large-scale, real-world dataset shows that utilizing 20 or 30 query terms extracted from all fields of an original query patent according to their  $\log(\text{tf})\text{idf}$  values helps form a representative search query out of the query patent and is found to be more effective than is using any number of query terms from any single field. It is shown that combining terms extracted from different fields of the query patent by giving higher importance to terms extracted from the abstract, claims, and description fields than to terms extracted from the title field is more effective than treating all extracted terms equally while forming the search query. Finally, utilizing the similarities between the IPC codes of the query patent and retrieved patents is shown to be beneficial to improve the effectiveness of the prior art search.**

## Introduction

Increasing global competition demands increased intellectual property protection. As a result of the fact that patents are an important type of intellectual property document, patent search has recently attracted increasing attention. A patent is issued for an invention after an official organization (e.g., U.S. Patent and Trademark Office) judges its utility,

inventiveness, and novelty. In the process of preparing or judging a patent application, an essential task is to search for prior art technologies to identify related patents that may invalidate the patent application. Prior art patent search, a type of patent retrieval, is the task of identifying previously published relevant patents for a given application patent file, and is an important part of the process of validating a patent application.

Prior art patent search, unlike other retrieval tasks, has many of its own characteristics that require careful analysis. First, prior art search is a recall-oriented task (i.e., recall is more important than is precision), as missing a relevant patent that could invalidate the patent application may affect the decision of the patent examiner. Therefore, it is more important to retrieve as many relevant documents as possible. A patent examiner who uses a search engine often examines the first few hundred documents (e.g., 100, 200, or 500) in the ranked list of retrieved patents, unlike a traditional search engine scenario where users often examine only the first few documents (i.e., precision is more important than is recall).

Another characteristic of prior art patent search is that patents are structured documents that have specific fields such as title, abstract, claims, description, and so on. Among these fields, a patent examiner especially examines the contents of the claims field to check the patentability of the application patent since an application patent's claim of originality can be found in this field. Motivated by this, most prior research on prior art patent search has used the words from the claims field as the search query terms without examining other alternatives (Fujii, 2007; Itoh, 2004, 2005; Mase, Matsubayashi, Ogawa, Iwayama, & Oshio, 2005). Although the claims field is an important field and terms extracted from this field should be utilized; other fields also should be carefully taken into account to be able to construct a representative search query for a query patent. Recently, Xue and Croft (2009a, 2009b) studied where the terms should be extracted from, and considered the claims field as well as the other fields; however, they only considered extracting terms from a single field at a time and did not consider combining the query terms

---

Received April 11, 2010; revised October 18, 2011; accepted October 19, 2011

© 2011 ASIS&T • Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.21708

extracted from different fields to form a structured query. To our knowledge, this is the first work (along with some preliminary work in our previous study, Cetintas & Si, 2009) that (a) considers using terms extracted from all fields to form a single search query and (b) explores the important follow-up question of how terms extracted from different fields should be combined to form the search query (e.g., whether more terms should be extracted from some fields while less terms should be extracted from others, and whether terms extracted from some fields should be given higher importance than should terms extracted from some other fields, etc.). After the search queries are constructed, an important question is how to utilize the structure of the documents to search using the structured query. To date, there has been very limited research in this direction. Itoh (2004) considered searching the constructed search queries in specific fields of the patents, namely, (a) in the combination of abstract and claims fields and (b) in the combination of all fields (the whole patent; i.e., no structure information utilized). However, this work did not consider other fields except abstract and claims fields, and more important, did not consider complex queries (i.e., used all the terms from the claims field without any selection as the search query). This article proposes a novel automated query construction approach for prior art patent search that (a) utilizes terms extracted from all fields in a query patent, (b) gives more importance to terms extracted from some of the fields (e.g., the abstract, claims, and description fields) than to terms extracted from others (e.g., the title field), and (c) considers utilizing the similarity between terms extracted from the fields of the query patent and terms in the corresponding fields of the patents to be retrieved (i.e., utilizes the structure of the patents to be retrieved) along with the similarity between extracted terms and whole patents (i.e., without utilizing the structure information). It is shown that (a) utilizing terms extracted from all fields outperforms the approach of using terms extracted from any single field alone; (b) while combining the terms extracted from all fields, differentiating the fields from which terms should be extracted by giving higher importance to terms extracted from those fields leads to more effective prior art search results; and (c) the approach of searching query terms extracted from individual fields in their corresponding fields along with the combination of all fields (i.e., the whole document) is comparable to the approach of searching the query terms only in the combination of all fields (i.e., when no structure information is utilized).

Another distinct property of patent documents is the fact that patents are assigned International Patent Classification (IPC) codes that can be exploited to calculate the similarity between a query patent and retrieved patents in prior art search. Prior research has utilized the integration of IPC code similarity between a query patent and has retrieved patents to rerank the results in the prior art search literature (Cetintas & Si, 2009; Harris, Arens, & Srinivasan, 2011; Itoh, 2005; Konishi, 2005; Xue & Croft, 2009a). Konishi (2005) compared the IPC codes of a query patent and its retrieved patents. If retrieved patents had one or more IPC classes in

common with a query patent, he multiplied the retrieval score by some constant. Itoh (2005) followed a filtering approach of two steps. In the first step, he used the first four characters of the main IPC code (positioned in the beginning of the IPC description) of a query patent as a constraint over its retrieved documents. In the second step; he used the first six characters of the main IPC codes of the top-five patents in the retrieved patents, and used those IPC codes as a constraint over a baseline run (i.e., eliminated all retrieved patents that did not have any of those partial IPC codes). Xue and Croft (2009a) combined retrieval score features, low-level features, and category features, and showed consistent results with prior work that had combined retrieval and category features together to improve effectiveness of the prior art search. Recently, our previous work (Cetintas & Si, 2009), similar to the approach by Itoh, used two features from the IPC code similarity: the first four characters of the IPC code and the first 11 characters of the IPC code. Instead of Itoh's (2005) second strict constraint, our previous study (Cetintas & Si, 2009) followed a reranking approach and combined these two features along with the retrieval score in a weighted way to rerank the retrieved patents. However, our previous work used preset constants while combining the three features and did not consider learning the weights directly from data. Harris et al. (2011) also used a similar approach that weights features from the IPC code similarity by manually applying a large range of different weights. The present article extends the prior work and proposes two methods for reranking the retrieved patents: (a) combining the first four and the first 11 characters of the IPC code along with the retrieval score with preset weights and (b) combining the two IPC similarity features and the retrieval score with weights directly learned from data. The results will show that (a) both approaches of reranking the retrieved patents increase the effectiveness of prior art search, and (b) the combination approach that uses weights directly learned from data outperforms the approach of using preset weights.

The rest of the article is arranged as follows: First, we introduce the dataset used in this work, and then describe several approaches for query construction strategies and postprocessing strategies. Our experimental results are then presented, and we conclude with a discussion and suggestions for future research directions.

## Corpus

The patent corpus from the Text REtrieval Conference (TREC) 2009 Chemical IR track was used in this work. The patent corpus consists of 1,185,012 patent files from the chemical domain (classified under the IPC codes C and A61K), and covers patents in the field until 2007, registered at the European Patent Office (EPO), the U.S. Patent Office (USPTO), and the World Intellectual Property Organization (WIPO) (three major patent offices). To our knowledge, this is the largest domain-specific (chemical) patent corpus released so far. The patents are in XML form and are provided by the Information Retrieval Facility (IRF). Each contains three

significant pieces of text along with the title: the abstract, description, and claims fields. In total, the uncompressed size of the patent files is 98.22 GB. The query set consists of 100 patent files (i.e., 23 USPTO patents and 77 EPO patents) (IRF; Cetintas & Si, 2009; Harris et al., 2011; Lupu, Huang, & Zhu, 2011). The results on all 100 query patents files are reported in this work. The Indri search engine (Metzler & Croft, 2004; Strohman, Metzler, Turtle, & Croft, 2004) was utilized for indexing. To be able to do structured retrieval over different fields of patent and article files, Indri should be given the names of particular fields for indexing. In this work, all significant textual parts of the patent files (the “invention-title,” “abstract,” “claims,” and “description” fields) were indexed. Note that prior work (Xue & Croft, 2009a, 2009b) has observed that other fields such as the brief summary (or background summary) field are effective for patent search, but they do not exist in this dataset as tag-delimited fields, as noted earlier. The default Indri retrieval model (i.e., Indri language modeling and inference networks) was applied as the document-retrieval model (Strohman et al., 2004); the standard INQUERY stop words list (Allan et al., 2000) was used to eliminate stop words, and the Porter stemmer was utilized to do stemming (Porter, 1980). Note that it also is possible to manually collect a stop word set similar to the approach followed in Mase et al. (2005); however, manual selection of stop words requires careful analysis of the language in which the patent documents are written as well as the content of the patent files, and may degrade the retrieval performance if stop words are not selected well. Therefore, a standard stop words list (i.e., the INQUERY stop words list) was used in this work. For the evaluation, prior work has mainly used mean average precision (MAP) and recall since prior art patent search is a recall-oriented task (as mentioned earlier), and higher precision is desired as well (Itoh, 2005; Konishi, 2005; Mase et al., 2005; Taraki, Fujii, & Ishikawa, 2004; Xue & Croft, 2009a, 2009b). In this work, to better analyze the results, we report the MAP and recall at 100 and 200 (R@100, R@200) retrieved documents as well as binary preference (Bpref), normalized discounted cumulative gain (NDCG), and mean reciprocal rank (MRR).

## Methods

### *Query Construction Strategies*

*Extraction of query terms.* In prior art search, an application patent (i.e., a query patent) should be transformed into a search query since patents are too long to be used directly as a search query. During this transformation, several factors should be carefully analyzed: (a) the number terms that should be extracted, (b) how these terms should be selected, (c) where [i.e., which field(s) of the original query patent] these terms should be extracted, (d) whether terms extracted from single fields should be used directly as the search query or whether terms extracted from different fields should be combined to form the search query, (e) if terms extracted from different fields should be combined, and if

so, how they should be combined (e.g., whether all terms should be treated equally or whether some terms should have higher importance than should others). Most prior approaches have used only the claims field for extracting the query terms (Fujii, 2007; Itoh, 2004, 2005; Mase et al., 2005). Xue and Croft (2009a) analyzed the first three enlisted factors mentioned earlier and showed that fields other than the claims field also should be considered. However, they did not consider combining terms extracted from different fields or how they should be combined. Our previous study (Cetintas & Si, 2009) used strategies considering all of the aforementioned factors; however, we only reported the effectiveness of the combined techniques because the work mainly focused on system description. Therefore, it was not clear which techniques are more effective or whether the choices made at each step were optimal.

This work proposes to consider  $N = \{10, 20, \dots, 100\}$  number of terms to be extracted from each field to decide on the number of terms that should be selected from each field. From a particular field, terms are ranked according to their term frequency ( $tf$ ) in that field (i.e., the number of times a word/term occurs in a patent field), the inverse document frequency ( $idf$ ) according to that field (i.e., number of all patent documents divided by the number of patent documents with a particular term in the corresponding field), and a variant of the combination of  $tf$  and  $idf$  [i.e.,  $\log(tf)*idf$ ] scores. The last ranking criteria (i.e.,  $\log(tf)*idf$ ) is based on the fact that the importance of a term in a document is not directly proportional to the number of occurrences of the term in the document (e.g., 20 occurrences of a term in a document is very unlikely to truly carry 20 times the significance of a single occurrence in the document) (Manning, Raghavan, & Shtze, 2008). It is a common modification to the popular  $tf$ - $idf$  weighting scheme (Baeza-Yates & Ribeiro-Neto, 1999; Manning et al., 2008) that has been used in previous research of prior art patent search for selecting terms from fields of the query patents when constructing the search query (Cetintas & Si, 2009; Xue & Croft, 2009a). Using the three selection criteria, the top- $N$  terms are selected as the extracted terms from each field and are the representative set of terms for that field of the original query patent (Baeza-Yates & Ribeiro-Neto, 1999). Terms are extracted from the title (tagged as “invention-title”), abstract, claims, and description (tagged as “abstract,” “claims,” and “description,” respectively) fields. We first investigate whether terms extracted only from single fields should be used directly as the search query or whether terms extracted from different fields should be combined. Note that the main intuition for combining terms extracted from different fields of an original query patent is the fact that the original query patent is a patent document itself, and selecting a good set of terms for all fields in the query patent will form a search query that better represents the original query patent. Therefore, we study whether terms extracted from different fields should be combined, and if so, how they should be combined. Specifically, we examine whether more terms should be extracted from some particular fields and less terms from other fields, and whether

TABLE 1. Indri query formats for structured prior-art search queries.

<pre>(a) #weight (     W_title #combine( TERMS_FROM_TITLE_FIELD )     W_abst #combine( TERMS_FROM_ABST_FIELD )     W_claims #combine( TERMS_FROM_CLAIMS_FIELD )     W_desc #combine( TERMS_FROM_DESC_FIELD ) )  (b) #weight (     W_title #combine( TERMS_FROM_TITLE_FIELD )      W_abst_1 #combine[abstract]( TERMS_FROM_ABST_FIELD )     W_abst_2 #combine( TERMS_FROM_ABST_FIELD )      W_claims_1 #combine[claims]( TERMS_FROM_CLAIMS_FIELD )     W_claims_2 #combine( TERMS_FROM_CLAIMS_FIELD )      W_desc_1 #combine[description]( TERMS_FROM_DESC_FIELD )     W_desc_2 #combine( TERMS_FROM_DESC_FIELD ) )</pre>	<p>Indri structured query formats for prior-art patent search task. Note that the first query searches all extracted terms in the whole body of the target patents (i.e., does not use the structure information of the query patent) whereas the second query additionally searches the terms extracted from the abstract, claims, and description fields of the query patent in the corresponding fields of the target patents.</p> <p>TERMS_FROM_{TITLE, ABST, CLAIMS, DESC}_FIELD are the set of selected terms from the title, abstract, claims, and description fields of a PA query patent. The weights <math>W_{title}</math>, <math>W_{abst}</math>, <math>W_{claims}</math>, and <math>W_{desc}</math> are set depending on the approaches followed in the Methods.</p>
--	---

terms extracted from all fields should be treated equally or whether terms extracted from some particular fields should be of higher importance. Specifically, Table 1a shows an Indri-formatted structured query that (with the weights  $\langle W_{title}, W_{abst}, W_{claims}, W_{desc} \rangle$  being set as  $\langle 1, 2, 2, 2 \rangle$ ) gives higher importance to terms extracted from the abstract, claims, and description fields than to terms extracted from the title field. The main intuition for such a structured and weighted query is twofold. First, terms extracted from different fields are grouped together so that they can be weighted easily when needed. Second, terms extracted from different fields can be used as separate subqueries to search different fields of the target patent documents to be retrieved. For instance, terms extracted from the abstract can be used as a subquery to search the abstract fields of the target patents whereas terms extracted from the claims field can be used as a subquery to search the claims field; the results can be combined together in a weighted way to calculate the final retrieval score (for more detailed motivation and an explanation of complex Indri queries, see Strohmman et al., 2004).

*Structured retrieval over patents.* Patents are structured documents, and rich, distinct information coming from the structured nature of these documents can be exploited during retrieval in prior art search. While most previous research has focused on how to utilize the structure of an original query patent while transforming it into a shorter search query, there is very limited research on how to utilize the structure of target patents to be retrieved using the constructed search queries. Itoh (2004) considered the document structure by searching the queries: (a) in the combination of the abstract and claims fields and (b) in the combination of all fields (the whole patent; i.e., no structure information utilized). However, his work did not consider other fields except the abstract and claims fields, and more important, he did not consider more complex queries (i.e., used all the terms from the claims field as the search query without any selection). The current study

proposes to utilize the structure of target patents by considering the effect of searching the constructed queries in different fields of the documents in a weighted way. In particular, two main approaches are combined to form a weighted search query. First, query terms extracted from individual fields (i.e., claims, abstract, and description) are searched in their corresponding fields of target patents by utilizing the structure of the target patents (e.g., terms extracted from the abstract field of the query patent are searched in the abstract field of the target patents, etc.). Second, query terms extracted from individual fields are searched in the combination of all fields of target patents (the whole patent; i.e., no structure information utilized). The main intuition of this approach is the fact that the query patent and the target patents to be retrieved are both structured patent files, and for a pair of similar query patents and target patents, the same fields are more likely to have higher similarity than are different fields. That is, thinking of the constructed structured query as a shortened version of the original query patent, there may be more similarity between the terms of the structured search query that are extracted from a particular field of the query patent and the terms in the same field of the documents to be retrieved. In particular, we take into account the similarity between terms extracted from the abstract, claims, and description fields of a query patent with terms in the abstract, claims, and description fields of the patents to be retrieved, respectively. Note that we do not consider the similarity between terms in the title field of the query patent and terms in the title field of the patents to be retrieved because the number of terms in the title fields of query patents and the number of terms in the title fields of target patents are too limited to have a good assessment of similarity. Formally, Table 1b shows an Indri query formatted by following the aforementioned ideas. Specifically; with the weights  $\langle W_{title}, W_{abst\_1}, W_{abst\_2}, W_{claims\_1}, W_{claims\_2}, W_{desc\_1}, W_{desc\_2} \rangle$  being set as  $\langle 1, 1, 2, 1, 2, 1, 2 \rangle$  (These weights are used in this work.), the Indri query in Table 1b gives higher importance to terms

extracted from the abstract, claims, and description fields than it does to terms extracted from the title field and takes into account the similarity between terms extracted from the same fields (i.e., from the abstract, claims, and description fields) of the query patent and the retrieved patents.

### *Postprocessing Strategies*

*Filtering according to priority dates.* Many people often work simultaneously to invent a method/technique to solve a particular technical problem. In such cases, most countries follow the first-to-file-system in granting the patent to the one who first filed the application. To protect a patent for the same invention in several countries, the Paris Convention for the Protection of Industrial Property provides that once a patent application is filed in a member country of the Convention, the patent applicant is entitled to claim “priority” over other applications in other member countries for a period of 12 months after the application, and the filing date of the first application is considered the “priority date.”

In prior art search task, the set of retrieved patents for a search query can be published before or after the priority date of the query patent. Therefore, the retrieved patents that are filed after the query patent (i.e., that have the earliest priority date is later the latest priority date of the query patent) cannot invalidate the query patent because they are filed after the query patent, so this does not violate the originality of the query patent. In this work, retrieved patents whose earliest priority date is later than the latest priority date of the query patent are discarded from the ranked list of the retrieved patents for a search query. If a retrieved patent does not have a priority date(s), its publication date is used for comparison.

*Reranking based on IPC similarities.* Patents are assigned IPC codes that can be exploited to improve the effectiveness of the prior art search. The IPC, established by the 1971 Strasbourg Agreement under the WIPO, has as its primary purpose to create an effective search tool for the retrieval of patent documents by intellectual property offices and other users. It is currently being used by more than 100 patent-issuing bodies and is the most widely used patent classification system. The IPC provides a hierarchical system for obtaining an internationally uniform classification of patents and utility models via assigning one or more of the 70,000 IPC codes that indicate the technical field(s) the patent covers (World Intellectual Property Office, 2011a). The hierarchical system arranges the IPC codes in a treelike structure with five components: section, class, subclass, groups, and subgroup. IPC Edition 20110101 has a total of eight sections, 129 classes, 631 subclasses, 7,392 main groups, and 62,493 subgroups (for a total of 69,885 groups) (World Intellectual Property Office, 2011b). Table 2 illustrates the five hierarchical components of an IPC classification (for more details about the IPC classification system, see World Intellectual Property Office, 2011a).

A significant amount of prior research has utilized the similarity of IPC codes of a query patent and its retrieved patents to rerank the results showing its importance (Cetintas & Si, 2009; Harris et al., 2011; Itoh, 2005; Konishi, 2005; Xue & Croft, 2009a). However, most previous research has followed quite-strict approaches such as multiplying the retrieval score by some constant in the case of common IPC classes (Konishi, 2005), eliminating all the retrieved patents that do not (partially) match the IPC code of the query patent or the top retrieved patents (Itoh, 2005), and so on (discussed earlier). To rerank the retrieved patents, Cetintas and Si (2009) followed a more relaxed approach than have previous works by combining the retrieval score and two features of IPC code similarity—the first 4 and the first 11 characters of the IPC code—in a weighted way to balance precision and recall. However, our previous work did not learn the optimum weights, but used preset (i.e., empirically tuned) constants while combining the three features (i.e., the retrieval score and the two IPC similarity features). Harris et al. (2011) also used a similar approach that weights features from the IPC code similarity by manually applying a large range of different weights. The current study significantly extends previous research and proposes two reranking approaches by combining the first 4 characters of the IPC code and the first 11 characters of the IPC code along with the retrieval score (a) with preset weights and (b) with weights directly learned from data.

In particular, two features of IPC code similarity are used in this work: the first 4 characters of the IPC code and first 11 characters of the IPC code. The first four characters of the IPC code include section symbol, class number, and subclass letter; and the first 11 characters (including spaces) additionally include a one- to three-digit group number, an oblique stroke, and a number of at least two digits representing a main group or subgroup. For instance, the two IPC codes “C01B 13/00” and “C01B 7/09” share the same section, class, and subclass (represented by the first four characters: “C01B”), yet belong to a different main group or subgroup (represented by all of the first 11 characters “C01B 13/00” and “C01B 7/09”). The intuition behind using both the first 4 and first 11 characters as a feature is to balance the trade-off between precision and recall (Cetintas & Si, 2009; Harris et al., 2011). The similarity calculated using the first 11 characters indicates a much more detailed match that requires the two compared patents to have exactly the same section, class, subclass, group, and subgroup in their IPC codes. Therefore having all of these constraints match with each other is quite useful to achieve high precision. However, IPC codes are placed manually, and the placements therefore are subject to noise and error. Requiring a strict match criterion (i.e., the first 11 characters) may prevent identifying similar patents that have different groups or subgroups (while having identical sections, classes, and subclasses), and this may lead to low recall. Hence, the similarity calculated using the first four characters also is utilized and indicates less-detailed similarity between two patents. Formally, the IPC code similarity between a query patent  $QP_i$  and a retrieved patent  $RP_j$

TABLE 2. An example illustrating the five hierarchical levels of an IPC classification provided for the IPC code C01B 13/00.

C	01	B	13/00	Main group: 4th level
Section: 1st level	Class: 2nd level	Subclass: 3rd level	or	
			13/02	Subgroup: 5th level
			Group/subgroup: 4th/5th levels	
C				Chemistry; Metallurgy
C	01			Inorganic Chemistry
C	01	B		Non-metallic elements; compounds thereof
C	01	B	13/00	Oxygen; Ozone; Oxides or hydroxides in general
C	01	B	13/02	Preparation of oxygen

using the first four characters [i.e.,  $IPC^4Sim(QP_i, RP_j)$ ] is calculated as follows:

$$IPC^4Sim(QP_i, RP_j) = \frac{\sum_{n=1}^{|S_{QP_i}^4|} \sum_{m=1}^{|S_{RP_j}^4|} \delta(S_{QP_i}^4(n) == S_{RP_j}^4(m))}{|S_{QP_i}^4|}, \quad (1)$$

where  $S_{QP_i}^4$  is the set of partial IPC codes (i.e., the first four characters) of a query patent  $QP_i$ , and similarly,  $S_{RP_j}^4$  is the set of partial (i.e., the first four characters of) IPC codes of a retrieved patent  $RP_j$ ,  $|S_{QP_i}^4|$  is the number of unique partial IPC codes of  $QP_i$ , and similarly,  $|S_{RP_j}^4|$  is the number of unique partial IPC codes of  $RP_j$ ,  $\delta$  is the indicator function that returns 1 if the two compared IPC codes are the same and 0 otherwise. The IPC code similarity between query patent  $QP_i$  and a retrieved patent  $RP_j$  using the first 11 characters [i.e.,  $IPC^{11}Sim(QP_i, RP_j)$ ] is calculated in a similar way.

After learning the two IPC code similarity features [i.e.,  $IPC^4Sim(QP_i, RP_j)$  and  $IPC^{11}Sim(QP_i, RP_j)$ ], there are two ways of reranking the retrieved documents, as discussed earlier: The first approach is using preset (i.e., empirically set) weights as used in Cetintas and Si (2009), and the second approach extends the first approach by learning the weights directly from data.

The first reranking approach utilizes preset weights to calculate the retrieval score between  $QP_i$  and  $RP_j$  [i.e.,  $RetScore^{old}(QP_i, RP_j)$ ] in a linear way as follows:

$$\begin{aligned} RetScore^{new}(QP_i, RP_j) &= RetScore^{old}(QP_i, RP_j) (1 - \alpha(\lambda * IPC^4Sim(QP_i, RP_j) \\ &+ (1 - \lambda)IPC^{11}Sim(QP_i, RP_j))) \end{aligned} \quad (2)$$

where  $\alpha$  is a constant that controls the effect of IPC code similarity on the updated retrieval score, and  $\lambda$  is a constant that controls the relative effect of  $IPC^4Sim(QP_i, RP_j)$  and  $IPC^{11}Sim(QP_i, RP_j)$  over the overall IPC similarity score. Our previous study (Cetintas & Si, 2009) empirically

(i.e., trying out a large range of potential values) set  $\alpha$  to 0.75 and  $\lambda$  to 0.2 for its experiments in the prior art patent search task of the TREC 2009 Chemical IR track because no relevance judgment data were available during the experiments (i.e., relevance judgment data were provided after all participants submitted their results to the TREC tracks, and the manual evaluation process was completed weeks/months after the submission deadline). [Note that  $RetScore^{old}(QP_i, RP_j)$  has a negative value.]. In the current study, we use the same values since the experiments are run on the same dataset provided by the TREC 2009 Chemical IR track.

The second approach that is proposed in this work models the combination of the old retrieval score and the two IPC similarity features in a probabilistic way, as follows:

$$\begin{aligned} P(RetScore^{new}(QP_i, RP_j) = 1 | RetScore^{old}(QP_i, RP_j), \\ IPC^4Sim(QP_i, RP_j), IPC^{11}Sim(QP_i, RP_j), \beta_0, \beta_1, \beta_2, \beta_3) \\ = \frac{\exp(\beta_0 + \beta_1 RetScore^{old}(QP_i, RP_j) + \\ \beta_2 IPC^4Sim(QP_i, RP_j) + \beta_3 IPC^{11}Sim(QP_i, RP_j))}{1 + \exp(\beta_0 + \beta_1 RetScore^{old}(QP_i, RP_j) + \\ \beta_2 IPC^4Sim(QP_i, RP_j) + \beta_3 IPC^{11}Sim(QP_i, RP_j))} \end{aligned} \quad (3)$$

where  $RetScore^{new}(QP_i, RP_j) = 1$  indicates that the retrieved patent is a patent that potentially may invalidate the query patent,  $\beta_1$  is the weight of the old retrieval score [i.e.,  $RetScore^{old}(QP_i, RP_j)$ ],  $\beta_2$  is the weight of  $IPC^4Sim(QP_i, RP_j)$ , and  $\beta_3$  is the weight of  $IPC^{11}Sim(QP_i, RP_j)$ . In this approach, twofold cross-validation is used to learn the weights; that is, the first half of the queries (the first half of the European patents and the first half of the U.S. patents) is used to learn the weights for the other half, and then the second half is used to learn the weights for the first half. Therefore, the results on the whole query patent set (i.e., on all 100 query patents that are provided in the TREC 2009 dataset) are reported for this set of experiments as well. An important configuration for this set of experiments is what portion of the irrelevant patents in the ranked lists should be used. Note that in the prior art patent search, the ranked list

of retrieved patents of a query patent has many more irrelevant patents than relevant patents (unlike a traditional learning scenario where the classes to be classified would be roughly balanced). Using all of the negative data instances (i.e., irrelevant patents) along with the positive data instances (i.e., relevant patents) will lead to an imbalanced dataset that will affect how well the parameters of the probabilistic learning model given in Equation 3 (i.e.,  $\beta_0, \beta_1, \beta_2, \beta_3$ ) are learned during the optimization process. Therefore, in this work, we also explore the effect of undersampling the negative data instances (i.e., the irrelevant patents) to better observe the effect of using a more balanced dataset between the positive and negative data instances (i.e., between relevant and irrelevant patents) on learning the model parameters.

## Results

We now present the results of the approaches that were presented earlier in the article. All approaches were evaluated on the dataset as described earlier.

An extensive set of experiments is conducted to address the following questions:

**RQ1:** How should query terms be selected during the transformation of the initial query patent into a search query? From which fields should terms be extracted? How many terms should be extracted from each field? How (i.e., according to what criteria) should the terms be selected?

**RQ2:** What is the effect of combining terms extracted from different fields to form the search query? Should terms extracted from single fields be used directly as the search query or should terms extracted from different fields of the query patent be combined? If they should be combined, how many terms should be extracted from each field?

**RQ3:** If terms extracted from different fields of the original query patent should be combined, how should they be combined? Should all terms be treated equally or should some of the terms have higher importance than should others?

**RQ4:** How effective is the approach of searching query terms extracted from individual fields (i.e., the claims, abstract, description) of the query patent in their corresponding fields along with the combination of all fields (i.e., the whole document) compared to the approach of searching the query terms only in the combination of all fields (i.e., whole document)? In other words, how effective is the approach of utilizing the structure of the patents to be retrieved while searching with the constructed structured query?

**RQ5:** How effective is the approach of using learned weights compared to using preset weights while reranking the retrieved patents by utilizing the IPC similarities?

### *Effect of Different Strategies for Selecting Query Terms*

The first set of experiments (discussed earlier) was conducted to explore how query terms should be selected during the transformation of the query patent into a search query. More specifically, the following questions are explored:

(a) How many terms should be selected from each field? (b) How should these terms be selected (i.e., according to what criteria)? (c) From which fields should these terms be extracted? Results for six different evaluation metrics—MAP, recall at 100 and 200 (R@100 & R@200), Bpref, NDCG, and MRR—are reported for detailed analysis in Tables 3 to 8, respectively. It can be seen that extracting 20 or 30 terms from each field is the best configuration overall for the number of terms to be extracted. Although extracting more terms slightly improves the results for some metrics (e.g., NDCG and MRR), the gains for these metrics are not very significant because including more terms in the search query significantly increases the time spent on search. The reason why extracting more than 20 or 30 terms from each field does not significantly improve the results is that after a specific number of (i.e., 20 or 30) terms, the set of extracted terms becomes less descriptive of the query patent due to term verbosity and therefore does not help construct a more representative and discriminative set of terms for forming the search query from the query patent. These results are consistent with the results reported in Xue and Croft (2009a).

It also can be seen from Tables 3 to 8 that selecting the terms according to the  $\log(\text{tf})\text{idf}$  values outperforms selecting them according to  $\text{idf}$  values, and selecting the terms according to  $\text{idf}$  values outperforms the approach of selecting them according to the  $\text{tf}$  values. The difference in performances of  $\log(\text{tf})\text{idf}$ ,  $\text{idf}$ , and  $\text{tf}$  (for extracting the terms) is more apparent for the description field, followed by the claims field, and not as much apparent for the abstract field. This is due to the fact that the description field is much more detailed than is the claims field, and the claims field is more detailed than is the abstract field. As there are more terms in a field among which the query terms are selected, the  $\text{idf}$  shows its discriminative power; combined with  $\text{tf}$  (i.e.,  $\log(\text{tf})\text{idf}$ ), it becomes the most effective term-extraction criterion. This set of results also is consistent with the results reported in Xue and Croft (2009a).

Results in Tables 3 to 6 also show that the description field of the original query patents is the best field from which to extract terms, followed by the claims, abstract, and title fields for MAP, R@100, R@200, and Bpref metrics, respectively. On the other hand, Tables 7 and 8 show that the abstract field is the best field from which the terms should be extracted for NDCG and MRR metrics, respectively. However, note that NDCG and MRR both favor high precision at top ranks (i.e., NDCG gives more importance to the precision of the top-ranked documents, and MRR only deals with the relevance of the most relevant document). Yet, as noted earlier, prior art patent search is a recall-oriented task, and the goal is to optimize recall while also having high precision. Therefore, for prior art patent search, the description field is the best field of the original query patent from which to select the terms for the search query. On the other hand, the observation that the abstract field is more successful in providing higher precision at top-ranked documents can be explained by the fact that the abstract field

TABLE 3. Mean Average Precision (MAP) results of several configurations for selecting query terms from the {Title, Abstract, Claims, Description} fields of the query patents.

MAP	Selection criteria	Selected terms ( <i>n</i> )									
		10	20	30	40	50	60	70	80	90	100
Title	n/a	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039
Abstract	tf	0.0253	0.0492	0.0564	0.0561	0.0557	0.055	0.0549	0.0549	0.0549	0.0549
	idf	0.0484	0.0534	0.0565	0.055	0.0546	0.0551	0.0549	0.0549	0.0549	0.0549
	log(tf)idf	0.0326	0.0531	<b>0.0583</b>	0.0569	0.0556	0.0551	0.0549	0.0549	0.0549	0.0549
Claims	tf	0.0011	0.0048	0.013	0.0247	0.0378	0.0439	0.0495	0.0534	0.0559	0.0584
	idf	0.0359	0.0478	0.0498	0.0533	0.0569	0.0619	0.0636	0.06	0.0583	0.0593
	log(tf)idf	0.0515	0.064	<b>0.0675</b>	0.0668	0.0641	0.0629	0.0607	0.0606	0.0563	0.0566
Description	tf	0.0012	0.0011	0.0014	0.0018	0.0045	0.0072	0.0076	0.0072	0.0106	0.0116
	idf	0.031	0.0466	0.0498	0.0492	0.0479	0.0487	0.048	0.0468	0.0484	0.0497
	log(tf)idf	0.0535	0.0687	<b>0.0718</b>	0.0696	<b>0.0743</b>	<b>0.0747</b>	<b>0.0769</b>	0.074	0.0733	0.0721

Note. Some of the best performances are shown in boldface.

TABLE 4. Recall at 100 (R@100) results of several configurations for selecting query terms from the {Title, Abstract, Claims, Description} fields of the query patents.

R@100	Selection criteria	Selected terms ( <i>n</i> )									
		10	20	30	40	50	60	70	80	90	100
Title	n/a	0.128	0.128	0.128	0.128	0.128	0.128	0.128	0.128	0.128	0.128
Abstract	tf	0.081	0.1376	0.1538	0.1572	0.1566	0.154	0.1556	0.1556	0.1556	0.1556
	idf	0.1265	0.1505	0.1562	0.1554	0.1552	0.1563	0.1556	0.1556	0.1556	0.1556
	log(tf)idf	0.0967	0.1483	<b>0.1582</b>	<b>0.1586</b>	0.1573	0.1556	0.1556	0.1556	0.1556	0.1556
Claims	tf	0.0059	0.0273	0.0505	0.095	0.1358	0.1508	0.1686	0.1784	0.1857	0.1999
	idf	0.1383	0.1657	0.1761	0.1956	0.2061	0.2202	0.2169	0.2137	0.2094	0.2036
	log(tf)idf	0.1776	<b>0.2173</b>	<b>0.2155</b>	<b>0.2196</b>	0.213	0.2139	0.2055	0.2106	0.2052	0.2056
Description	tf	0.0046	0.0055	0.0062	0.0102	0.0202	0.0251	0.0283	0.0343	0.0446	0.0502
	idf	0.1111	0.1437	0.1638	0.1675	0.156	0.1538	0.1504	0.1522	0.1567	0.1586
	log(tf)idf	0.1985	0.2328	<b>0.2471</b>	0.2305	0.2327	0.2402	0.2439	0.2307	0.231	0.2311

Note. Some of the best performances are shown in boldface.

TABLE 5. Recall at 200 (R@200) results of several configurations for selecting query terms from the {Title, Abstract, Claims, Description} fields of the query patents.

R@200	Selection criteria	Selected terms ( <i>n</i> )									
		10	20	30	40	50	60	70	80	90	100
Title	n/a	0.1684	0.1684	0.1684	0.1684	0.1684	0.1684	0.1684	0.1684	0.1684	0.1684
Abstract	tf	0.1239	0.191	0.2211	0.2225	0.2229	0.2203	0.2203	0.2203	0.2203	0.2203
	idf	0.1877	0.2189	0.215	0.222	0.2195	0.2216	0.2203	0.2203	0.2203	0.2203
	log(tf)idf	0.1357	0.2026	<b>0.2196</b>	<b>0.2232</b>	0.2229	0.221	0.2203	0.2203	0.2203	0.2203
Claims	tf	0.0108	0.0355	0.0662	0.1261	0.1751	0.2051	0.215	0.2227	0.2318	0.2561
	idf	0.1777	0.2241	0.2457	0.2504	0.2636	0.2797	0.2823	0.2731	0.2715	0.2621
	log(tf)idf	0.2237	0.2692	<b>0.2834</b>	0.2786	0.277	0.2711	0.2644	0.2671	0.262	0.2625
Description	tf	0.0086	0.0099	0.009	0.0145	0.0233	0.0324	0.0415	0.0456	0.062	0.0671
	idf	0.1245	0.1781	0.192	0.188	0.1837	0.1794	0.1762	0.1747	0.1804	0.1839
	log(tf)idf	0.246	0.2823	<b>0.2919</b>	0.2879	<b>0.3</b>	<b>0.2952</b>	0.2877	0.2862	0.2873	0.2845

Note. Some of the best performances are shown in boldface.

TABLE 6. Binary preference (Bpref) results of several configurations for selecting query terms from the {Title, Abstract, Claims, Description} fields of the query patents.

Bpref	Selection criteria	Selected terms ( <i>n</i> )									
		10	20	30	40	50	60	70	80	90	100
Title	n/a	0.2893	0.2893	0.2893	0.2893	0.2893	0.2893	0.2893	0.2893	0.2893	0.2893
Abstract	tf	0.2234	0.3464	<b>0.3981</b>	0.392	0.3932	0.3921	0.3901	0.3901	0.3901	0.3901
	idf	0.3414	0.3863	0.3833	0.393	0.3884	0.3905	0.3901	0.3901	0.3901	0.3901
	log(tf)idf	0.2773	0.3539	<b>0.3937</b>	<b>0.3967</b>	<b>0.3936</b>	0.3905	0.3901	0.3901	0.3901	0.3901
Claims	tf	0.0214	0.0611	0.1204	0.1879	0.257	0.2961	0.329	0.3407	0.3559	0.3751
	idf	0.3014	0.3672	0.3782	0.3861	0.395	<b>0.4097</b>	<b>0.413</b>	0.4014	0.402	0.4002
	log(tf)idf	0.3645	<b>0.4017</b>	0.3972	<b>0.4068</b>	<b>0.4049</b>	<b>0.4069</b>	<b>0.4082</b>	<b>0.4042</b>	<b>0.4029</b>	0.3989
Description	tf	0.0152	0.0227	0.0271	0.0303	0.0407	0.0534	0.0638	0.0811	0.0973	0.1058
	idf	0.2048	0.2394	0.2401	0.2353	0.2287	0.2228	0.2211	0.2089	0.2222	0.2171
	log(tf)idf	0.3538	0.3835	<b>0.4091</b>	<b>0.4103</b>	0.4034	<b>0.4113</b>	0.4051	0.3949	0.3965	0.3733

Note. Some of the best performances are shown in boldface.

TABLE 7. Normalized discounted cumulative gain (NDCG) results of several configurations for selecting query terms from the {Title, Abstract, Claims, Description} fields of the query patents.

NDCG	Selection criteria	Selected terms ( <i>n</i> )									
		10	20	30	40	50	60	70	80	90	100
Title	n/a	0.1597	0.1597	0.1597	0.1597	0.1597	0.1597	0.1597	0.1597	0.1597	0.1597
Abstract	tf	0.1341	0.2279	0.2599	0.2599	0.2593	0.2575	0.2566	0.2566	0.2566	0.2566
	idf	0.2255	0.251	0.2553	0.2572	0.2554	0.257	0.2566	0.2566	0.2566	0.2566
	log(tf)idf	0.1694	0.2339	<b>0.2621</b>	<b>0.2621</b>	0.2592	0.2569	0.2566	0.2566	0.2566	0.2566
Claims	tf	0.0111	0.0329	0.0664	0.1126	0.1515	0.1731	0.1897	0.2014	0.2116	0.221
	idf	0.1626	0.1994	0.2062	0.2167	0.2219	<b>0.2313</b>	<b>0.2364</b>	0.2296	0.2279	0.2291
	log(tf)idf	0.2004	0.2306	<b>0.2362</b>	<b>0.238</b>	<b>0.2337</b>	<b>0.2331</b>	0.2307	0.23	0.2255	0.2253
Description	tf	0.009	0.011	0.0133	0.0155	0.0229	0.0308	0.037	0.0442	0.0565	0.0599
	idf	0.1244	0.1588	0.1647	0.1638	0.1585	0.1603	0.1592	0.154	0.1598	0.1582
	log(tf)idf	0.2033	0.2301	<b>0.2416</b>	0.2381	<b>0.2424</b>	<b>0.2437</b>	<b>0.2429</b>	0.2378	0.2387	0.2313

Note. Some of the best performances are shown in boldface.

TABLE 8. Mean reciprocal rank (MRR) results of several configurations for selecting query terms from the {Title, Abstract, Claims, Description} fields of the query patents.

MRR	Selection criteria	Selected terms ( <i>n</i> )									
		10	20	30	40	50	60	70	80	90	100
Title	n/a	0.1714	0.1714	0.1714	0.1714	0.1714	0.1714	0.1714	0.1714	0.1714	0.1714
Abstract	tf	0.148	0.2914	0.3614	0.3642	0.3531	0.3473	0.3474	0.3474	0.3474	0.3474
	idf	0.3587	0.3358	<b>0.381</b>	0.3424	0.3476	0.3474	0.3474	0.3474	0.3474	0.3474
	log(tf)idf	0.1737	0.3124	<b>0.4076</b>	0.3588	0.3536	0.3476	0.3474	0.3474	0.3474	0.3474
Claims	tf	0.0031	0.0326	0.079	0.1682	0.1913	0.2088	0.2154	0.2473	0.2672	0.2667
	idf	0.1965	0.221	0.2277	0.2377	0.2378	0.2419	0.2565	0.2487	0.2439	0.2537
	log(tf)idf	0.2302	0.265	<b>0.2753</b>	0.2653	0.2456	0.2594	0.2462	0.2397	0.2361	0.2459
Description	tf	0.0019	0.0064	0.0059	0.0098	0.0154	0.0291	0.0403	0.0497	0.0806	0.0812
	idf	0.1967	0.2457	0.2525	<b>0.2673</b>	0.2339	<b>0.2677</b>	<b>0.2644</b>	0.246	0.2481	0.246
	log(tf)idf	0.2134	0.2439	0.2581	0.2541	<b>0.2608</b>	0.2528	0.252	0.2556	<b>0.2689</b>	<b>0.2655</b>

Note. Some of the best performances are shown in boldface.

TABLE 9. Results of combining terms extracted from all fields.

	Best scores by terms extracted from a single field	Terms extracted from each field					
		5	10	20	30	40	50
MAP	0.0769	0.0663	0.0733	<b>0.0825</b>	0.0822	0.0808	0.08
R@100	0.2471	0.216	0.2381	<b>0.2657</b>	<b>0.2659</b>	0.2619	0.2572
R@200	0.3000	0.2724	<b>0.3138</b>	<b>0.3382</b>	<b>0.3422</b>	0.3317	0.3225
Bpref	0.413	0.4037	0.4448	<b>0.4647</b>	<b>0.4789</b>	<b>0.4797</b>	<b>0.4681</b>
NDCG	0.2621	0.2376	0.2561	<b>0.2740</b>	<b>0.2777</b>	<b>0.2760</b>	0.2728
MRR	<b>0.4076</b>	0.2818	0.2561	0.2819	0.2838	0.2953	0.2861

Note. Some of the best performances are shown in boldface.

MAP = mean average precision; Bpref = binary preference; NDCG = normalized discounted cumulative gain; MRR = mean reciprocal rank.

is more succinct than are the claims and description fields, and that having a more targeted set of selected terms becomes effective in finding patents relevant to the query patent. However, as observed in Tables 3 to 6, it leads to lower recall since many relevant patents that use different wordings do not match this limited (yet, more targeted) set of terms that are extracted from the abstract (due to data sparsity). Therefore, the set of terms extracted from the description field becomes more successful in achieving higher recall (i.e., R@100 and R@200) or overall precision (i.e., MAP and Bpref) levels than does the set of terms extracted from the abstract field. This result is clearly different from the majority of prior approaches that have favored selecting terms only from the claims field (Fujii, 2007; Itoh, 2004, 2005; Mase et al., 2005), and is consistent with some later works (Cetintas & Si, 2009; Mahdabi, Keikha, Gerani, Landoni, & Crestani, 2011). Note that Xue and Croft (2009a, 2009b) found a different field than the claims field—the “background summary” or “brief summary” field—to be effective for from which to select the query terms. Yet, as noted earlier, the examined fields are the only ones with significant text data in the patent corpus used in this work.

Results in this section clearly show that for transforming a query patent into a search query, the extraction of query terms is a very important step and should be carefully analyzed; and although the claims field is an important one from which to extract the query terms, other fields also should be taken into account. Specifically, it is shown that selecting 20 or 30 terms according to the log(tf)idf criterion from fields of the original query patent has been found to be the most effective way for constructing the search query, and the description field is shown to be the most effective in contrast to the claims field that has been shown to be effective by prior research.

#### *Effect of Combining Terms Extracted From Different Fields*

The second set of experiments (discussed earlier) was conducted to see the effect of combining terms extracted from different fields to form the search query. More specifically, questions of whether terms extracted only from the description field (i.e., the best performing field, as shown earlier) should be used directly as the search query (as in prior

research), whether other terms from other fields also should be utilized, and whether more terms should be extracted from particular fields (e.g., the description field) rather than other fields to form the final search query are explored. It can be seen from Table 9 that selecting around 20 or 30 terms from each field clearly outperforms the approach of using terms extracted only from a single field (i.e., the description field for MAP, R@100, R@200, and Bpref, and the abstract field for NDCG as shown earlier), except for MRR. It also can be seen that although selecting only five terms is not enough to be effective, selecting 10 terms is comparable with the best results of using terms from only one field (except for MRR). Selecting 20 and more terms outperforms the best results of using terms extracted from only one field for all evaluation metrics (except for MRR). The achievements in the performance of the approach of utilizing terms extracted from all fields are more apparent for Bpref, R@100, R@200, and MAP, and not that apparent for NDCG. For MRR, the approach of combining terms extracted from individual fields performs much lower than does the approach of selecting terms from a single field (i.e., the abstract). As mentioned previously, NDCG and MRR both favor high precision at top ranks, and the set of terms selected from the abstract is a quite targeted set that is effective at achieving high precision at top ranks in the ranked list of the retrieved patents. Supporting this set of targeted terms with other terms extracted from other fields does not make a significant difference for NDCG (and can be ignored considering the significant increase in the time spent on search), and dramatically deteriorates the performance for MRR. Yet, complementing this set of targeted terms with other terms extracted from other fields helps to achieve a more broader set of terms, and becomes more successful in achieving higher recall (i.e., R@100 and R@200) or overall precision (i.e., MAP and Bpref) levels by being able to deal with relevant patents that are not easy to identify due to different wordings used by different authors and by helping to create a search query that is more representative of the original query patent. To our knowledge, this is the first work to explicitly show that utilizing terms extracted from every field of the original query patent is a more effective approach than is utilizing terms extracted only from a single (i.e., the best performing) field for constructing the search query in prior art patent search.

TABLE 10. Results of selecting different number of terms from different fields when combining terms extracted from all fields.

Title	No. of terms extracted from each field			MAP	R@100	R@200	Bpref	NDCG	MRR
	Abstract	Claims	Description						
10	10	10	10	0.0733	0.2381	0.3138	0.4448	0.2561	0.2561
10	10	10	20	0.0777	0.2443	0.3261	<b>0.4652</b>	0.2671	0.2772
10	10	20	20	0.0814	0.2620	<b>0.3386</b>	0.4615	0.2714	0.2823
10	20	20	20	<b>0.0825</b>	<b>0.2657</b>	<b>0.3382</b>	<b>0.4647</b>	<b>0.2740</b>	0.2819
20	20	20	20	<b>0.0825</b>	<b>0.2657</b>	<b>0.3382</b>	<b>0.4647</b>	<b>0.2740</b>	0.2819
20	20	20	10	0.0779	0.2563	0.3277	0.4616	0.2677	0.2815
20	20	10	10	0.0745	0.2405	0.3162	0.4486	0.2589	0.2732
20	10	10	10	0.0733	0.2381	0.3138	0.4448	0.2561	0.2561
20	20	20	20	<b>0.0825</b>	<b>0.2657</b>	<b>0.3382</b>	<b>0.4647</b>	<b>0.2740</b>	0.2819
20	20	20	30	<b>0.0824</b>	0.2628	0.3355	<b>0.4718</b>	<b>0.2762</b>	0.2923
20	20	30	30	<b>0.0822</b>	<b>0.2652</b>	<b>0.3440</b>	<b>0.4775</b>	<b>0.2777</b>	0.2823
20	30	30	30	<b>0.0822</b>	<b>0.2659</b>	<b>0.3422</b>	<b>0.4789</b>	<b>0.2777</b>	0.2838
30	30	30	30	<b>0.0822</b>	<b>0.2659</b>	<b>0.3422</b>	<b>0.4789</b>	<b>0.2777</b>	0.2838
30	30	30	20	<b>0.0843</b>	<b>0.2651</b>	<b>0.3454</b>	<b>0.4689</b>	<b>0.2765</b>	0.2785
30	30	20	20	<b>0.0834</b>	<b>0.2674</b>	<b>0.3411</b>	<b>0.4689</b>	<b>0.2770</b>	0.2847
30	20	20	20	<b>0.0825</b>	<b>0.2657</b>	<b>0.3382</b>	<b>0.4647</b>	<b>0.2740</b>	0.2819
Best scores by terms extracted from a single field				0.0769	0.2471	0.3000	0.413	0.2621	<b>0.4076</b>

Note. Some of the best performances are shown in boldface.

MAP = mean average precision; Bpref = binary preference; NDCG = normalized discounted cumulative gain; MRR = mean reciprocal rank.

Note that although the approach of using terms extracted from all the fields of the original query patent has been shown to outperform the approach of using terms extracted from a single field, it may not be optimal because it utilizes the same number of terms extracted from each field. Therefore, we also explore the question of whether more or less terms should be extracted from particular fields (e.g., the description field) than from other fields to form the final search query. However, trying all possible combinations (e.g., trying to extract  $x$  terms, for  $x$  in  $\{5, 10, 20, 30, 40, 50\}$ , from the title, abstract, claims, and description fields) requires testing  $6^4$  different term-selection combinations, which is too costly. Yet, it is possible to try some variations in the number of terms extracted from each field to test whether extracting more or less terms from more or less important fields has any significant impact on the retrieval effectiveness of the constructed query. Table 10 shows the results of 16 different combinations. It can be seen that extracting more or less terms from more or less important fields (the description field being the most important, followed by the claims, abstract, and title fields) does not make a significant difference for any of the six performance metrics (i.e., MAP, R@100, R@200, Bpref, NDCG, and MRR). Previous results observed in Table 9 showed that extracting 20 or 30 terms from each field of the query patent is enough to construct an effective search query, and the results in Table 10 show that increasing or decreasing the number of extracted terms from some particular fields does not make a significant difference

in constructing an effective search query. Therefore, for the rest of the experiments, the approach of selecting 20 terms from each field is followed.

#### *Effect of Different Strategies for Combining Terms Extracted From Different Fields*

The third set of experiments was conducted to see whether terms extracted from some fields of the original query patent should be given higher importance than should terms extracted from other fields (discussed earlier). More specifically, while combining terms extracted from every field of the original query patent to construct an effective and more representative search query, the question of whether terms extracted from all fields should have the same importance or terms extracted from particular fields should be given more importance than should others is explored. Table 11 lists the results for all possible combinations of giving higher importance to the four fields (i.e.,  $2^4$  different combinations for the four different fields). It can be seen that giving higher importance to terms extracted from the description field generally leads to higher accuracy in the results for all metrics, except for MRR. On the other hand, it also is observed that terms extracted from the title field should not be given higher importance than should terms extracted from other fields to achieve higher results for all metrics, except for MRR. Results in Table 11 also show that the configuration that gives higher importance to terms extracted from the abstract, claims, and

TABLE 11. Results of selecting fields of higher importance (from which to extract terms).

Field of higher importance (shown with *)				MAP	R@100	R@200	Bpref	NDCG	MRR
Title	Abstract	Claims	Description						
				0.0825	0.2657	0.3382	0.4647	0.2740	0.2819
			*	0.0848	0.2665	<b>0.3486</b>	<b>0.4707</b>	0.2772	0.2730
		*		0.0831	0.2678	0.3395	0.4682	0.2765	<b>0.2920</b>
		*	*	0.0853	<b>0.2720</b>	<b>0.3474</b>	<b>0.4795</b>	<b>0.2813</b>	0.2827
	*			0.0823	0.2605	0.3361	0.4609	0.2727	0.2860
	*		*	0.0854	0.2664	<b>0.3468</b>	0.4690	0.2778	0.2713
	*	*		0.0837	0.2702	0.3391	0.4676	0.2769	0.2860
	*	*	*	<b>0.0861</b>	<b>0.2730</b>	<b>0.3472</b>	<b>0.4788</b>	<b>0.2823</b>	0.2827
*				0.0794	0.2423	0.3148	0.4509	0.2659	<b>0.2936</b>
*			*	0.0818	0.2603	0.3288	0.4578	0.2702	0.2836
*		*		0.0807	0.2588	0.3349	0.4649	0.2722	0.2881
*		*	*	0.0820	0.2658	0.3370	0.4617	0.2723	0.2822
*	*			0.0793	0.2387	0.3126	0.4484	0.2651	<b>0.2918</b>
*	*		*	0.0818	0.2583	0.3285	0.4573	0.2697	0.2797
*	*	*		0.0804	0.2558	0.3333	0.4624	0.2709	0.2838
*	*	*	*	0.0825	0.2657	0.3382	0.4647	0.2740	0.2819

Note. Some of the best performances are shown in boldface.

MAP = mean average precision; Bpref = binary preference; NDCG = normalized discounted cumulative gain; MRR = mean reciprocal rank.

description fields than to terms extracted from the title field outperforms all other configurations for all metrics, except for MRR. For MRR, as opposed to all the other metrics, the approach that gives higher importance to terms extracted from the title field than to terms extracted from the other fields outperforms the approach that gives higher importance to terms extracted from the abstract, claims, and description fields than to terms extracted from the title field. Note that in Tables 3 to 8, the title field is shown to be the least effective field when extracting terms from single fields. In this section, it is shown that the configuration that gives higher importance to all terms except terms extracted from the title field performs the best for all metrics, except the MRR. When transforming a query patent into a search query, the aim is to select the most representative terms to help the prior art search engine identify patents that may potentially invalidate the query patent. The title field, unlike other fields, is the shortest field in a patent, and all terms in the titles are extracted without any selection process [i.e., selected regardless of  $\log(\text{tf}/\text{idf})$  scores], unlike terms to be extracted from other fields, and are observed to form a quite targeted set of terms. Therefore, the set of terms in the title field is too limited to represent the whole query patent well, which also is validated in Table 11 as the configuration that gives higher importance to all terms extracted from all fields, but the title field becomes the best performing configuration for all metrics other than the MRR. Yet, as discussed earlier, a targeted set of terms is quite useful to achieve higher MRR since MRR deals only with precision of the top-ranked document

whereas other metrics deal with recall or overall precision. Supporting the small set of targeted terms extracted from the title with terms extracted from the abstract, claims, and description fields becomes quite effective for achieving the highest MRR levels of all configurations. To our knowledge, this is the first work to explicitly show that terms extracted from all fields should be combined in a weighted way such that terms extracted from the title field should be of lower importance for achieving higher recall and overall precision levels whereas terms extracted from the title field should be of higher importance than should terms extracted from other fields for achieving higher MRR levels.

#### Effect of Using Structured Retrieval Over Patents

Unlike previous experiments that have focused on how to utilize the structure of an original query patent when transforming it into an effective search query, the set of experiments (discussed earlier) in this section was conducted to explore the question of how to utilize the structure of the target patents to be retrieved via the constructed search queries. More specifically, unlike prior work by Itoh (2004) that searched simple, unstructured queries (consisting of all terms extracted from the claims field) in the combination of the abstract and the claims fields, the approach followed in this work searches a constructed search query that consists of the terms extracted from the abstract, claims, and description fields in those same fields (i.e., the field from which they were extracted) of the patents to be retrieved, respectively, as well

TABLE 12. Results of searching the constructed queries in different fields of the target patents as well as in the combination of all fields (i.e., the whole patent documents).

Configurations of approaches		MAP	R@100	R@200	Bpref	NDCG	MRR
Searching in corresponding fields	Fields of higher importance for extracted terms						
No	No	0.0825	0.2657	0.3382	0.4647	0.2740	0.2819
Yes	No	0.0842	0.2676	0.3478	0.4717	0.2777	0.2767
Yes	Yes	0.0839	0.2694	0.3393	0.4658	0.2760	0.2845

MAP = mean average precision; Bpref = binary preference; NDCG = normalized discounted cumulative gain; MRR = mean reciprocal rank.

TABLE 13. Results of re-ranking the retrieved documents with weights learned from data for MAP, R@{100,200,500 and 1000}, Bpref, NDCG, and MRR values.

Reranking configuration	MAP	R@100	R@200	R@500	R@1000	Bpref	NDCG	MRR	% of negative data instances
No reranking	0.0861	0.2730	0.3472	0.4259	0.4788	0.4788	0.2823	0.2827	n/a
Preset weights	0.0786	0.2703	0.3444	0.4501	0.4997	0.4997	0.2844	0.3099	n/a
With weights learned from data	<b>0.0867</b>	<b>0.2797</b>	<b>0.3611</b>	<b>0.4629</b>	<b>0.5008</b>	<b>0.5008</b>	<b>0.2947</b>	<b>0.3280</b>	%100
	0.0819	0.2744	0.3544	0.4566	0.5004	0.5004	0.2888	0.3129	%20
	0.0789	0.2705	0.3444	0.4562	0.4990	0.4990	0.2848	0.3119	%4
	0.0772	0.2644	0.3386	0.4476	0.4969	0.4969	0.2819	0.3047	% 2
	0.0763	0.2548	0.3360	0.4454	0.4968	0.4968	0.2801	0.2997	%1
	0.0753	0.2533	0.3307	0.4434	0.4968	0.4968	0.2785	0.2951	%0.5
	0.0569	0.1963	0.2850	0.4118	0.4808	0.4808	0.2459	0.2455	%0.25

Note. Some of the best performances are shown in boldface.

MAP = mean average precision; Bpref = binary preference; NDCG = normalized discounted cumulative gain; MRR = mean reciprocal rank.

as in combination of all fields (i.e., the whole patent document without using any structure information). Furthermore, a weighted structured query (giving more importance to terms extracted from the abstract, claims and description fields than to terms extracted from the title field) is searched similarly also utilizing the structure of the patents to be retrieved. It can be seen from Table 12 that both approaches that utilize the structure of the patents to be retrieved perform comparably with the approach that does not utilize the structure of the patents to be retrieved. Although the main intuition for this set of experiments was that there may be more similarity between the same fields of a query patent and its retrieved patents that are relevant to the query patent, the results show that the similarity between query terms extracted from a particular field and terms in the corresponding field of the patents to be retrieved is not significantly different than the similarity between query terms extracted from a particular field and terms in all fields of the patents to be retrieved. Similar to the results shown in Table 9 (that a combination set of terms extracted from all fields is a more representative set of terms for the whole query patent and is a more effective set of terms than are terms that are extracted from a single field), the results in this section show that the combination set of terms in all fields of the patents to be retrieved is enough to achieve more effective results than are terms in specific fields (when searched with the constructed search query). Therefore, for utilizing the structure of the patents in prior art patent search, a combination of (extracted) terms in all fields

is not only an effective way of constructing a search query but also an effective target for retrieval using the constructed search queries.

#### Effect of Learning Weights for IPC-Based Reranking

The last set of experiments (discussed earlier) was conducted to evaluate the approaches of using preset or learned (from data) weights while reranking the retrieved patents with respect to IPC similarities compared to an approach that does not do reranking.

This section particularly explores the effect of reranking the retrieved patents based on IPC similarities via preset or learned (from data) weights. It can be seen from Table 13 that the reranking approach with preset weights outperforms the approach that does not do reranking for recall at 500 (R@500), recall at 1000 (R@1000), Bpref, and MRR. On the other hand, its performance is comparable to the approach without reranking for R@100, R@200, and NDCG, and lower for MAP. This interesting set of results shows that although reranking strategy seems to work regarding the whole ranked list, it does not seem to make an effective change at top ranks (e.g., R@100 and R@200). Yet, for reranking the top relevant patent for each query (i.e., for MRR), it is effective.

The approach that does reranking with weights learned from data clearly outperforms both approaches for all measures. Also note that the number of negative data instances

used while learning the weights affects the reranking effectiveness. Table 13 shows detailed results of using different undersampling configurations for the negative data instances (i.e., the set of irrelevant patents in the TREC-provided relevance judgment data) that are used along with all positive data instances (i.e., the set of relevant patents in the TREC-provided relevance judgment data). Note that in a prior art search scenario, the number of irrelevant (i.e., negative) patent instances in a ranked list is much higher than is the number of relevant (i.e., positive) patent instances, leading to an imbalanced dataset. In our experiments, the configuration that randomly samples 1% of the negative data instances has roughly an equal number of positive and negative data instances from which to learn the weights. However, the configuration that uses all negative data instances outperforms all other configurations, as shown in Table 13. This is because random undersampling methods potentially can remove certain important examples, which leads to significant performance loss (Chawla, Japkowicz, & Kolcz, 2004). The approach of using all irrelevant documents along with all positive instances is consistent with the approaches followed in previous studies for other applications (Craswell, Hawking, Wilkinson, & Wu, 2003; Qin, Liu, Xu, & Li, 2008, 2010).

This set of experiments shows that (a) reranking can improve the effectiveness of prior art search, (b) the weights that are used to combine the retrieval score and the IPC similarity features should be learned from data, and (c) all negative data instances should be used along with all positive instances to learn the weights.

## Conclusions

This article proposes an automated prior art search approach that first constructs structured queries by combining terms extracted from different fields of a query patent in a weighted way, and then utilizes the similarity between IPC codes of the query patent and its retrieved patents along with the retrieval score to rerank the retrieved documents. Extensive experiments were conducted on a large-scale dataset to explore the questions of how many query terms should be extracted from the fields of the original query patent to form a search query, from which fields of the original query patent these terms should be extracted, according to what criteria the terms should be selected, whether terms extracted from different fields should be combined (If so, how they should be combined.), whether terms extracted from individual fields should be searched in their corresponding fields along with the combination of all fields of the patents to be retrieved (i.e., whether utilizing the structure of the patents to be retrieved is essential to achieve higher effectiveness). Furthermore, we also explored how IPC similarities can be exploited (with preset weights or with weights learned from data) to rerank the retrieved documents to increase the effectiveness of the prior art search. Specifically, the results show that extracting 20 or 30 terms according to  $\log(\text{tf})\text{idf}$  scores from every field of the query patent is an effective way of constructing

a search query from the original query patent. Combining terms extracted from every field outperformed common prior approaches that have used terms extracted from only a single field (e.g., claims, description, etc.), extracting more terms from some fields while extracting less terms from others was observed to be not significantly different than extracting the same number of terms from all fields, and giving higher importance to terms extracted from the abstract, claims, and description fields than to terms extracted from the title field was found to be more effective. The results also show that searching query terms extracted from individual fields in their corresponding fields along with the combination of all fields (i.e., utilizing the structure of the patents to be retrieved) performs comparably with the approach of searching query terms only in the combination of all fields (i.e., without utilizing the structure of the patents to be retrieved). Furthermore, we also showed that (a) utilization of IPC similarities to rerank the retrieved patents increases the effectiveness of the prior art search, (b) the approach of using weights that are directly learned from data outperforms the approach of using user-set weights to combine the retrieval score and IPC similarity features, and (c) utilizing all of the negative data instances (i.e., the majority set of irrelevant patents) along with all the positive instances (i.e., the minority set of relevant patents) is beneficial when learning the weights from the data.

## Future Work

There are several possibilities to extend this research. One direction is that the approach of combining terms extracted from different fields is heuristic (i.e., assigning preset higher weights to important fields), and the weights that are used to give more importance to some fields can be better optimized (as shown in the case of IPC similarities). One possible direction is to separately search terms extracted from each field, and combine the ranked lists for the query terms of each field intelligently to form a final ranked list similar to the results-merging techniques available in the federated search (Callan, 2000; Cetintas & Si, 2007; Lu, Meng, Shu, Yu, & Liu, 2005). Following a query independent approach, the combination weights for the document scores in each ranked list can be learned before calculating the final scores to form the final ranked list. To our knowledge, this direction has not yet been explored and may be worth pursuing comprehensively. A second potentially important direction stems from the structured nature of the target patents (i.e., the patents to be retrieved). Robertson, Zaragoza, and Taylor (2004) showed that when searching structured documents with a search query, scores independently calculated for each field and (linearly) combined to compute the final score for a document may lead to poor performance due to nonlinear term-frequency saturation, depending on the retrieval model (e.g., in the case of BM25). Although it also was noted to not be a serious issue for language-modeling-based retrieval models that separately index each field (as in the case of this work), previous work only has considered unstructured

(i.e., simpler) search queries to search the structured documents. It may be quite interesting to carefully study the interaction between structured search queries (as in the case of patent retrieval) to search structured documents, and how different retrieval models would behave under different conditions (e.g., separately indexing each field, linearly combining the retrieval scores of different fields, etc.). A final direction is that the dataset used in this work consists of chemical (i.e., domain-specific) patents; it is possible to use chemical molecule mining approaches that can utilize the similarity between the molecules in the query patent, and the retrieved patents similar to the approaches followed in related research (Corbett & Murray-Rust, 2006; Klinger, Kolárik, Fluck, Hofmann-Apitius, & Friedrich, 2008; Lupu et al., 2011; Mukherjea & Bamba, 2004; Sun, Mitra, & Giles, 2008). However, because utilizing such similarities may not be efficient enough to be able to consider all the patent files in the corpus for a query patent, it also may be worthwhile to explore the tradeoff between effectiveness and efficiency via a study that addresses those possible challenges.

## Acknowledgments

Suleyman Cetintas and Luo Si have been supported by National Science Foundation Grants IIS-0746830 and IIS-1017837, a research grant from Indiana Economic Development Corporation, and a research grant from Purdue University.

## References

- Allan, J., Connell, M.E., Croft, W.B., Feng, F.F., Fisher, D., & Li, X. (2000). INQUERY and TREC-9. In Proceedings of the 9th Text REtrieval Conference (TREC '09) (pp. 551–562). NIST Special Publication 500-249. Gaithersburg, MD: National Institute of Standards and Technology.
- Baeza-Yates, R., & Ribeiro-Neto, B. (Eds.). (1999). *Modern information retrieval*. New York: ACM Press.
- Callan, J. (2000). Distributed information retrieval. In B. Croft (Ed.), *Advances in information retrieval* (pp. 127–150). Dordrecht, The Netherlands: Kluwer.
- Cetintas, S., & Si, L. (2007). Exploration of the tradeoff between effectiveness and efficiency for results merging in federated search. In Proceedings of the 30th International Conference on Research and Development on Information Retrieval (ACM SIGIR'07) (pp. 707–708). New York: ACM Press.
- Cetintas, S., & Si, L. (2009). Strategies for effective chemical information retrieval. In Proceedings of the 18th Text REtrieval Conference (TREC '09). NIST Special Publication 500-278. Gaithersburg, MD: National Institute of Standards and Technology.
- Chawla, N.V., Japkowicz, N., & Kolcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *ACM Special Interest Group on Knowledge Discovery and Data Mining Explorations Newsletter*, 6(1), 1–6.
- Corbett, P., & Murray-Rust, P. (2006). High-throughput identification of chemistry in life science texts. In Proceedings of the Second International Symposium on Computational Life Science (CompLife '06) (pp. 107–118). Berlin, Germany: Springer-Verlag.
- Craswell, N., Hawking, D., Wilkinson, R., & Wu, M. (2003). Overview of the TREC 2003 web track. In Proceedings of the 12th Text REtrieval Conference. Gaithersburg, MD: National Institute of Standards and Technology.
- Fujii, A. (2007). Enhancing patent retrieval by citation analysis. In Proceedings of the 30th International Conference on Research and Development on Information Retrieval (ACM SIGIR'07) (pp. 793–794). New York: ACM Press.
- Harris, C.G., Arens, R., & Srinivasan, P. (2011). Using classification code hierarchies for patent prior art searches. In M. Lupu, K. Mayer, J. Tait, A.J. Trippe, & W.B. Croft (Eds.), *Current challenges in patent information retrieval* (pp. 287–304). Berlin, Germany: Springer-Verlag.
- Itoh, H. (2004, June). NTCIR-4 Patent retrieval experiments at RICOH. Paper presented at the NII Test Collection for IR Systems Workshop (NTCIR-4), Tokyo, Japan.
- Itoh, H. (2005, June). NTCIR-5 Patent retrieval experiments at RICOH. Paper presented at the NII Test Collection for IR Systems Workshop (NTCIR-5), Tokyo, Japan.
- Klinger, R., Kolárik, C., Fluck, J., Hofmann-Apitius, M., & Friedrich, C.M. (2008). Detection of IUPAC and IUPAC-like chemical names. *Bioinformatics*, 24, i268–i276.
- Konishi, K. (2005, June). Query terms extraction from patent document for invalidity search. Paper presented at the NII Test Collection for IR Systems Workshop (NTCIR-5), Tokyo, Japan.
- Lu, Y., Meng, W., Shu, L., Yu, C., & Liu, K.-L. (2005). Evaluation of result merging strategies for metasearch engines. In Proceedings of the Sixth International Conference on Web Information Systems Engineering (pp. 53–66). Berlin, Germany: Springer-Verlag.
- Lupu, M., Huang, J., & Zhu, J. (2011). Evaluation of chemical information retrieval tools. In M. Lupu, K. Mayer, J. Tait, A.J. Trippe, & W.B. Croft (Eds.), *Current challenges in patent information retrieval* (pp. 109–124). Berlin, Germany: Springer-Verlag.
- Mahdabi, P., Keikha, M., Gerani, S., Landoni, M., & Crestani, F. (2011). Building queries for prior-art search. In Proceedings of the Second Information Retrieval Facility Conference (IRFC'11) (pp. 3–15). Berlin, Germany: Springer-Verlag.
- Manning, C.D., Raghavan, P., & Schtze, H. (Eds.). (2008). *Introduction to information retrieval*. New York: Cambridge University Press.
- Mase, H., Matsubayashi, T., Ogawa, Y., Iwayama, M., & Oshio, T. (2005). Proposal of two-stage patent retrieval method considering the claim structure. *ACM Transactions on Asian Language Information Processing*, 4(2), 186–202.
- Metzler, D., & Croft, B. (2004). Combining the language model and inference network approaches to retrieval. *Information Processing & Management*, 40(5), 735–750.
- Mukherjea, S., & Bamba, B. (2004). BioPatentMiner: An information retrieval system for BioMedical patents. In Proceedings of the 30th International Conference on Very Large Data Bases (VLDB'04) (pp. 1066–1077). San Francisco: Morgan Kaufmann.
- Porter, M.F. (1980). An algorithm for suffix stripping. *Program: Electronic Library and Information Systems*, 14(3), 130–137.
- Robertson, S., Zaragoza, H., & Taylor, M. (2004). Simple BM25 extension to multiple weighted fields. In Proceedings of the 13th ACM International Conference on Information and Knowledge Management (ACM CIKM'04) (pp. 42–49). New York: ACM Press.
- Qin, T., Liu, T.-Y., Xu, J., & Li, H. (2008). How to make LETOR more useful and reliable. In Proceedings of the ACM Special Interest Group on Information Retrieval 2008 Workshop on Learning to Rank for Information Retrieval (pp. 52–58). New York: ACM Press.
- Qin, T., Liu, T.-Y., Xu, J., & Li, H. (2010). LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4), 346–374.
- Strohman, T., Metzler, D., Turtle, H., & Croft, W. (2004). Indri: A language model-based search engine for complex queries. In Proceedings of the International Conference on Intelligent Analysis.
- Sun, B., Mitra, P., & Giles, L. (2008). Mining, indexing, and searching for textual chemical molecule information on the web. In Proceedings of the 17th International Conference on the World Wide Web (ACM WWW'08) (pp. 735–744). New York: ACM Press.
- Taraki, T., Fujii, A., & Ishikawa, T. (2004). Associative document retrieval by query subtopic analysis and its applications to invalidity patent search. In Proceedings of the 13th International Conference on Information and

- Knowledge Management (ACM CIKM'04) (pp. 399–405). New York: ACM Press.
- World Intellectual Property Office (WIPO). (2011a). International Patent Classification (IPC). Retrieved from [http://www.wipo.int/export/sites/www/classifications/ipc/en/guide/guide\\_ipc.pdf](http://www.wipo.int/export/sites/www/classifications/ipc/en/guide/guide_ipc.pdf)
- World Intellectual Property Office (WIPO). (2011b). International Patent Classification (IPC) statistics. Retrieved from <http://www.wipo.int/classifications/ipc/en/ITsupport/Version20110101/transformations/stats.html>
- Xue, X., & Croft, B. (2009a). Automatic query generation for patent search. In Proceedings of the 18th International Conference on Information and Knowledge Management (ACM CIKM'09) (pp. 2037–2040). New York: ACM Press.
- Xue, X., & Croft, B. (2009b). Transforming patents into prior-art queries. In Proceedings of the 32nd International Conference on Research and Development on Information Retrieval (ACM SIGIR'09) (pp. 808–809). New York: ACM Press.