IRNLP@KAIST in the Subtask of Research Papers Classification in NTCIR-8

Bashar Al-Shboul
Korea Advanced Institute of Science and Technology
119, Munji-ro, Yuseong-gu, Daejeon, 305-732, Korea
bashar@kaist.ac.kr

Sung-Hyon Myaeng
Korea Advanced Institute of Science and Technology
119, Munji-ro, Yuseong-gu, Daejeon, 305-732, Korea
myaeng@kaist.ac.kr

ABSTRACT
In this paper, we present a novel query expansion approach based on splitting the user query into a set of N-grams, and expanding them separately utilizing a set of research articles. Our approach is based on retrieving a set of relevant research articles, process their abstracts to expand the query/searched term or phrase. We aim to expand terms that a regular relevance feedback might ignore. Our work shows an improvement over several classification levels compared to several methods of expansion.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: *Information filtering, Query formulation*, *Relevance feedback*

General Terms
Algorithms

Keywords
Query Expansion, Relevance Feedback

1. INTRODUCTION
Patent classification is important in real-life applications such as patent examination, invalidity search, trend detection, and technology map generation. The International Patent Classification (IPC) is a global standard hierarchical patent classification system, containing more than 50,000 classes at the most detailed level.

In this paper we propose a new query expansion model utilizing research articles. We use k-Nearest Neighboring (kNN) to measure the document similarity score between the topics and training data, based on the well-known OKAPI BM25. Expecting to achieve a good result we use query expansion using WordNet and DBpedia. In the following parts of this paper, we will present a detailed description of our system. This paper is organized as follows. In section 2 we present problem statement. In section 3 we review some related work. Our approach is discussed in section 4. Section 5 describes our experiments. Evaluation results are listed in section 6. Finally we conclude and present our future work in section 7.

2. PROBLEM STATEMENT
NTCIR Patent Mining task is about experimenting classifying cross-genre documents to IPC different levels of generalization. Participants are provided patent documents for training their systems, and topics representing research papers for testing. Given (*428*) Sub-Classes, (*6588*) Main-Groups, (*38491*) Sub-Groups, a training data consisting of (*0.9* Million) USPTO patents, (*3.7* Million) PAJ patents, and (*0.4* Million) research papers, and a set of (*95*) Topics, for the dry run, representing research papers from different domains, it is required to assign IPCs to the given topics based on relevancy/similarity between topics and given training documents. At the formal run, a set of (*633*) different topics were given. Some overlappings between topics were detected, thus topics were clustered, and the remaining (*549*) topics were used for last evaluation.

In this work, we consider the search query as an optimizing problem, hereby, we try to optimize the quality of the search query and its expansion using research articles index. Our main concern is that using one query to apply relevance feedback using the research articles index may ignore some phrases by preferring the whole query weight over a single phrase weight. This can occur due to the sparse distribution of the classes in the corpus. In that sense, a split search for each query is performed to assure the expansion of each phrase.

After all, multiple IPCs are to be assigned for each topic, ordered by score. We considered feeding the system with topics to output three classifications for the three different IPC levels (Sub-Class, Main Group, and Sub-Group).

3. RELATED WORK
The most related work is [1], where a KNN classifier was trained using PAJ or/and USPTO patents. Given set of topics, the system calculates similarity in vector space. Several similarity measures were tested separately, namely: Cosine, OKAPI, Pivoted Document Length Normalization, and Log-Linear, then, the system selects the highest *1000* similar documents from each similarity measure for ranking. Several ranking methods were tested either. Number of class occurrences in the list, the order of class occurrences, summation of scores for similar IPC documents, and summation of penalized scores for similar IPC documents. This work was ranked first in NTCIR-7 PT-MN task.

Another related work [2] considered extracting verbs, nouns, and adjectives from a topic. Terms are weighted using TFIDF, and compared to all other documents in the index. Top k similar documents are retrieved, and their IPCs are scored based on the document similarity. Similar IPCs scored are summed, and then the list is ranked in descending order. However, an addition was using 3 different query expansion methods. The three methods share the same steps except for one. Generally, query expansion is done by extracting all terms from a topic file, and weight them

using TFIDF. Further, a number of terms with the highest TFIDF values are selected for searching a paper or patent index for re-weighting. Among the retrieved documents list, those with the highest scores are selected, and their abstracts are retrieved for further processing. Then, terms in the abstracts are extracts, and weighted. Terms with highest TF values are selected to be added (first QE approach), or averaged and multiplied by a constant (second QE approach), or even to replace the original set of terms for another round of search (third QE approach).

4. RETRIEVAL APPROACH

Our system consists of two major parts: an offline part, and an online part. Indexing part is done offline using PAJ and research articles provided separately by NTCIR to generate two different indices, one for each document type. Querying part is done using a set of topics, representing research articles, also provided by NTCIR. Figure 1 shows the schema of step-by-step process.

The figure only shows the PAJ index; however, research articles index is included in the Query Expansion process explained in figure 2. The icons above the query expansion process are for WordNet and DBpedia respectively.

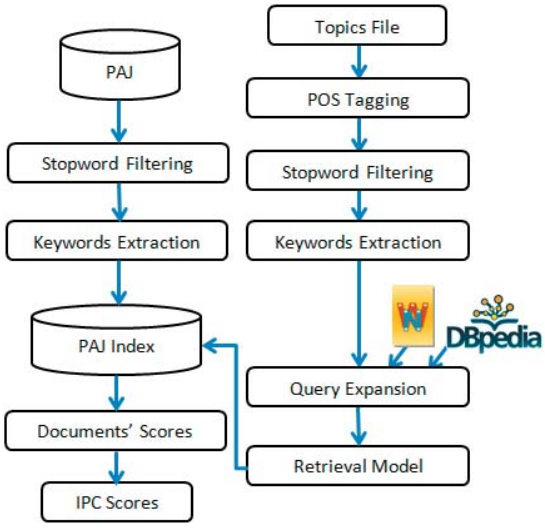

Figure 1: Retrieval Framework

4.1. Preprocessing

In both parts, indexing and querying, we used almost the same steps for preprocessing. First, we used POS tagging for keyword extraction. For that purpose we used Stanford POS Tagger. Specifically, we were targeting Nouns, Verbs, and Adjectives. Before extracting them, we considered filtering stopwords using a customized list collected from different resources. As for the extraction part, we considered Regular Expressions (RegEx).

RE was chosen to extract keywords and keyphrases due to the brief amount of information provided by PAJ documents, as a title and abstract might not be enough for statistic-based (TFIDF) or probabilistic-based methods (Bayes). Furthermore, PAJ documents are machine translated, following a strict grammatical rules; given that, we realized that parsing before POS tagging will probably give accurate tags.

Nouns, Verbs and Adjectives were extracted from topics, then phrases were extracted using RegEx, then all unigrams that were covered by a phrase or more has been removed. After stopword filtering all remaining keywords and keyphrases were considered for processing. Following is the RegEx used to extract keyphrases.

(VBG|VBN|JJ|JJR|JJS)(NN|NNP|NNPS|NNS)

Where VBG and VBN are verbs. JJ, JJR and JJS represent Adjectives. NN, NNS, NNP and NNPS represent Nouns and Noun phrases. This RegEx was built based on our observations over the tagged topics.

4.2. Query Expansion

After extracting and filtering keywords/keyphrases from topic files to generate queries, each query was split into Bag-of-Words (BOW) and Bag-of-Phrases (BOP). WordNet was used for expanding keywords in the BOW. As for BOP we followed the steps as in figure 2.

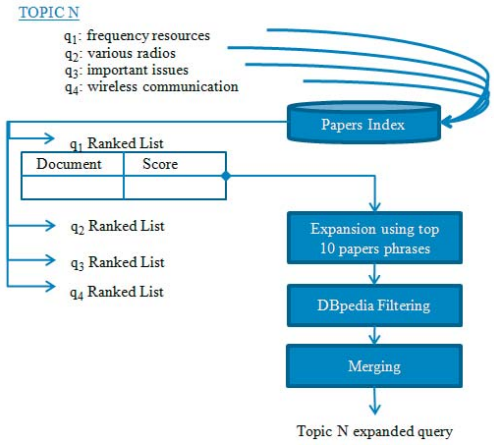

Figure 2: Query Expansion schema

Let BOP = $\{q_1,q_2..q_n\}$ be the set of keyphrases representing sub-queries for topic N, we search the research articles index using each single sub-query, and retrieve its ranked list of scored documents with similarity values. Further, we select the top 10 articles for keyword/keyphrase extraction. Then, we search DBpedia for SKOS related to each phrase, and intersect the retrieved ontologies returned by DBpedia. The remaining phrases after the intersection process are ranked based on Ontology Frequency, and the top 5 phrases were selected. WordNet and DBpedia expansions are explained in the following subsections.

4.2.1. WordNet

WordNet is a large database of English language, which contains nouns, verbs and adjectives grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are connected by means of conceptual-semantic and lexical relations, forming a network.

As for WordNet, we expanded by adding synonymies to BOW. An example is shown in figure 3. Q represents the query keywords, and QE represents the expanded query using WordNet.

Q: Railway, automatic, train, control, system, safety, technologies, effectiveness,..

QE: railway, railroad, railroad_line, railway_line, railway_system, railroad_train, control, system, safety, technologies, effectiveness, effectivity, effectualness, effectuality,…

Figure 3: Topic 10100 keywords and their expansions

4.2.2.DBpedia

DBpedia is a semantic web project maintained by University of Leipzig and Free University of Berlin. The aim of the project is to extract structured information from information available in Wikipedia, allowing users subsequent queries of relationships and properties associated with Wikipedia resources. DBpedia is also linked to the other big project as a part of Linked Data project[1].

As a partner of Wikipedia project, DBpedia has almost instant access to any changes available in Wikipedia. And, among the downloadable resources of DBpedia there is a dataset of Wikipedia articles abstracts. Abstract are the short representations of the topic and are much more suited for our final purpose – query expansion.

Utilizing DBpedia abstracts for query expansion proved to be ineffective, due to large amounts of information that had to be scanned multiple times. Instead, we have located and used an interface to lookup web-service, which is an application of IR system for DBpedia. Lookup service turned to be a good solution, as it would retrieve the required information.

We have also analyzed the URI pages of DBpedia articles and found that it is possible to use semantic web hierarchy, namely SKOS, to further expand query. SKOS (Simple Knowledge Organization System) is a W3C recommended common data model for knowledge organization systems such as thesauri, classification schemes, subject heading systems and taxonomies. Application of SKOS allows defining links between semantic concepts, creating a network, which can be navigated easily. We have utilized skos:subject links of DBpedia resources for the sake of keyphrase expansion. The papers index was searched using each keyphrase, and SKOS were used to define candidate phrases for expansion. For several keyphrases in a single topic, the lists of candidate phrases were intersected, and from the remaining phrases we selected the top 5 occurring phrases or all of them if they are less than 5.

5. INDEXING

In our experiments, we used Lemur IR Toolkit (www.lemurproject.com). Titles and Abstracts of both PAJ and research articles were indexed in two separate indices. Furthermore, in some experiments we used titles of PAJ files to build the PAJ index.

5.1. Lemur Project

Lemur is an open-source project made in collaboration between the Computer Science Department at the University of Massachusetts and the School of Computer Science at Carnegie Mellon University. It supports indexing of large-scale text databases, construction of simple language models for documents, queries, or sub collections, and the implementation of retrieval systems based on language models, as well as a variety of other retrieval models.

In work [4] performance of Lemur and two other famous information retrieval systems Terrier (University of Glasgow) and Lucene (Apache Software Foundation) was tested for the purpose of Geographic Information Retrieval system. It was reported that Lemur achieved the best MAP (*0.2619*) compared to Lucene (*0.2207*) and Terrier (*0.2570*) on English queries.

6. EXPERIMENTS

As explained before, each topic represents a research article. In this work, queries were generated from the given topics using titles, abstracts and keywords assigned manually by the authors of the research article.

In this work, 12 different experiments, in terms of expansion method, search method, PAJ fields covered by search (title only or both title and abstract), and the number of IPCs assigned to each topic for evaluation purposes, were conducted. Details about our experiments are shown in table 1.

Table 1: Experiments description

	WordNet	DBpedia	PAJ Titles Index	PAJ Titles and Abstracts Index	10 IPCs per Topic	100 IPCs per Topic	Research Papers Index Expansion	DBpedia Filtering	Simple Query	Structured Query
1			X			x			x	
2	x		X			x			x	
3	x		X		x				x	
4	x			x		x			x	
5		x		x		x			x	
6	x	x		x		x			x	
7	x	x		x	x			x	x	
8	x	x		x		x		x	x	
9	x	x	X		x		x	x		x
10	x	x	X			x	x	x		x
11	x	x		x	x		x	x		x
12	x	x		x		x	x	x		x

As a baseline, we used simple query, consisting only of keywords (verbs, nouns and adjectives), to search the PAJ titles index. No expansions were done. This experiment was annotated as (1). MAP is shown in table 2. In experiment (2), we tested the effect of WordNet expansion on experiment (1). Then we repeated experiment (2) with less IPCs assigned for each topic (10 rather than 100) and we annotated as experiment (3). Further, in experiment (4) we repeated experiment (2) on titles and abstracts index rather than titles index. As we found that experiment (4) show better Mean-Average Precision (MAP) compared to experiment (2), we believe that repeating

[1] DBpedia homepage - http://dbpedia.org/About

experiment (3) with 100 IPCs per topic is an unneeded repetition.

In experiment (5) keyphrases were expanded using DBpedia. Following, in experiment (6), the keywords expanded using WordNet were merged with the keyphrases expanded using DBpedia into one query to understand their effect. For the previous 6 experiments, experiment (4) gave the best results over all 3 classification levels.

Further, filtering out the phrases returned by DBpedia appeared to be important, thus, all phrases that didn't appear more than once after expansion were removed, then phrases were split using spaces, and used the remaining words as keywords for the query in experiment (7). As shown in table 1, experiment (7) gave better MAP on the general level, but worse on the more specific ones. This can be explained by the nature of the expansion as adding words from context sensitive phrases to a query means adding insignificant terms to the query. In figure 3, adding the word "line", or "system" to the query will affect the search results as they both have high term frequencies and spread over a wide range of documents.

Further, the number of assigned IPCs per topic was increased from 10 to 100 to study the possibility that a correct IPC might appear in the first 100 among the ranked IPCs per topic (experiment 8). As shown, experiment (8) gave better MAP than all former experiments on all 3 levels. Moreover, experiment (8) was the best among our all 12 experiments in terms of number of correctly retrieved and ranked IPCs on the sub-class level. On the other 2 levels of classification, it gave better MAP than other former experiments, and almost the same number of correctly retrieved and ranked IPCs per topic compared to experiment (2).

In the following experiments, we decided to use a structured query to search keywords as keywords, and keyphrases as phrases. For that purpose we utilized Indri query language in Lemur [6].

In experiments 9 through 12, we applied our proposed query expansion model (Figure 2), and conducted the same experiment with different search fields (title for experiments 9 and 10, and title and abstract for experiments 11 and 12). And for experiments 9 and 11 we assigned only 10 IPCs per topic compared to 100 IPCs in experiments 10 and 12.

Table 2: MAP for Dry Run experiments

	Sub-Class	*Main Group*	*Sub-Group*
1	.399	.188	.109
2	.514	.294	.161
3	.445	.244	.139
4	.583	.345	.233
5	.390	.190	.095
6	.477	.257	.145
7	.550	.286	.157
8	.559	.307	.178
9	.491	.287	.143
10	.503	.311	.157
11	**.622**	**.402**	**.248**
12	**.628**	**.418**	**.266**

After every search, IPCs of the top 1000 returned ranked documents were collected, and then re-ranked using Listweak scoring as shown below.

$$Score(c;q) = \sum_{i=1}^{k} occur(c, d_i).Sim(q, d_i) * 0.95 \dots\dots\dots (1)$$

Where *occur* is a function returning *1* if the document belongs to the IPC class, and *0* otherwise. *Sim* function returns the score of the document for the given query q, and the *0.95* (as in [1]) is the punishing factor for the least similar documents. This formula returns the score of each IPC among the ranked list of retrieved patents for a specific query q. After collecting all scores for all possible IPCs, we re-rank IPCs again, and assign them to the query topic.

It is worth mentioning that some PAJ documents were assigned *2* IPCs instead of one. Before re-ranking step, and after retrieving that ranked list of documents, and selecting the top *1000*, we expanded all documents by means of IPCs. Meaning that, if a document has two IPCs, it will be added to the list twice with a different single IPC for each one, but with an equivalent score/rank.

As we found that our last two experiments (11 & 12) were the best in terms of MAP, we decided to submit their results to the Formal Run evaluation. Results are listed in table 3.

Table 3: MAP for Formal Run systems

	Sub-Class	*Main Group*	*Sub-Group*
11	0.6089	0.4221	0.2450
12	0.6162	0.4388	0.2648

Apparently, both systems almost maintained their MAP values with slight increment in the Main Group level, and slight decrement in the sub-class level.

7. CONCLUSIONS

In this work, we presented a way of query expansion using research articles for the purpose of document classification. We found that WordNet can be a very good tool for query expansion for the document classification task. Even though the technique is quite simple, it achieved good retrieval results on Sub-Class, and Main Group levels. DBpedia along with WN achieved better results than WN if we emerged keyphrases. Adding keyphrases to the query enhanced the retrieval and thus classification results on all levels, along with their expansions, they contributed to this enhancement on all levels, negating what have been concluded earlier in [7] that phrases, especially bi-grams, have potentials for document representation but not for text categorization.

ACKNOWLEDGEMENTS

"This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2010-C1090-1011-0008)

8. REFERENCES

[1] Tong Xiao, et. al., **KNN and Re-ranking Models for English Patent Mining at NTCIR-7**, 2008

[2] Hisao Mase & Makoto Iwayama, **NTCIR-7 Patent Mining Experiments at Hitachi**, NTCIR-7, 2008

[3] Youngho Kim, et al., **Automatic discovery of technology trends from patent text,** SAC '09, pp. 1480-1487.

[4] Josre M. Perea-Ortega, et al., **Comparing Several Textual Information Retrieval Systems for the Geographical Information Retrieval Task**, Proceedings of the 13th international conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems, 2008, pp. 142-147.

[5] Yuen-Hsien Tseng, et al., **Text mining techniques for patent analysis**, Information Processing & Management, Vol. 43, No. 5. (September 2007), pp. 1216-1247.

[6] Lemur IR Toolkit (www.lemurproject.com)

[7] Bekkerman R. & Allan J., **Using Bigrams in Text Categorization**, CIIR ,Technical Report IR-408, 2004

[8] Hidetsugu Nanba et. al., **Overview of the Patent Mining Task at the NTCIR-8 Workshop**, NTCIR-8, 2010.