Patent Search using IPC Classification Vectors

Manisha Verma
Language Technologies Research Centre
International Institute of Information Technology
Hyderabad, India
manisha.verma@research.iiit.ac.in

Vasudeva Varma
Language Technologies Research Centre
International Institute of Information Technology
Hyderabad, India
vv@iiit.ac.in

ABSTRACT

Finding similar patents is a challenging task in patent information retrieval. A patent application is often a starting point to find similar inventions. Keyword search for similar patents requires significant domain expertise and may not fetch relevant results. We propose a novel representation for patents and use a two stage approach to find similar patents. Each patent is represented as an IPC class vector. Citation network of patents is used to propagate these vectors from a node (patent) to its neighbors (cited patents). Thus, each patent is represented as a weighted combination of its IPC information as well as of its neighbors. A query patent is represented as a vector using its IPC information and similar patents can be simply found by comparing this vector with vectors of patents in the corpus. Text based search is used to re-rank this solution set to improve precision. We experiment with two similarity measures and re-ranking strategies to empirically show that our representation is effective in improving both precision and recall of queries of CLEF-2011 dataset.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Experimentation, Theory

Keywords

Patent Retrieval, Keyphrase Extraction

1. INTRODUCTION

Patents give exclusive rights to the inventor for using and protecting his intellectual property. For a patent to be granted, the invention has to be novel, non-obvious and useful. Since a lot of patents are present in digital form on the web and with the number of patents filed and granted each year increasing rapidly, patent examiners today use information retrieval tools to accomplish several search tasks.

A patent examiner routinely performs search tasks like prior art search, patentability search, novelty search and invalidity search. The examiner starts with a document and manually creates suitable queries to search patent databases. Since a lot of time is spent in constructing relevant queries, transforming the document into a query automatically would save the examiner a lot of effort. Hence, one should be able to input a document as a query instead of making several queries. Query formulation is still a manual process and automating it would require the right combination of query formation, refinement and expansion techniques.

Existing approaches form queries using the text of a patent application for prior-art retrieval or invalidity search. In such a scenario quality of search results depends heavily on the choice of words and their weight in the query. If the quality of results is poor, then any post-processing on these results would not improve precision. In this paper, we propose a different approach to find similar patents. Instead of using the patent text to query, we use its meta-data to perform initial search. These results are then re-ranked using queries constructed from patent text. Meta-data based search ensures high recall and re-ranking ensures high precision.

We propose a novel representation for patents to perform meta-data search. Each patent is manually assigned one or more International Patent Classification (IPC) codes. We use this information to represent each patent as an IPC class vector. Citation network of patents is used to propagate these vectors from a node (patent) to its neighbors (cited patents). Thus, each patent vector is a weighted combination of its neighbors IPC information and its own. A query patent is represented as a vector using its IPC information and similar patents can simply be found by comparing query vector with all the vectors in the corpus. Text based search is used to re-rank this solution set to improve precision. We show that our approach outperforms text based search both in terms of precision and recall on 300 sample queries from CLEF-IP 2011 dataset.

In Section 2, we discuss the current state of the art in patent retrieval. In Section 3, we explain our approach of vector generation and result re-ranking. The experiments, result and analysis are explained in Section 4 and Section 5. Conclusion and future work are discussed in Section 6.

PaIR'11, October 24, 2011, Glasgow, Scotland, UK.

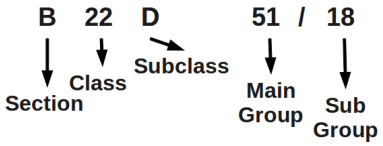

Figure 1: IPC Class code (B22D 51/18)

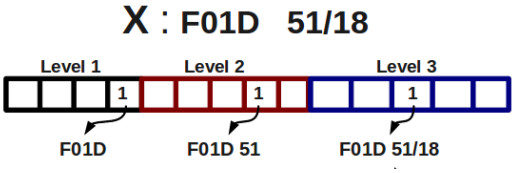

Figure 2: IPC Vector of a sample patent X

2. RELATED WORK

Prior-art retrieval and invalidity search have received considerable attention from the research community recently. Several workshops by NTCIR [1] and CLEF [2] have been conducted to evaluate and improve the state of the art in patent retrieval. Patent retrieval poses a unique challenge as the language of patents is not only ambiguous but also contains several new terms and concepts introduced by the inventor. This results in a lot of content that discusses similar aspects but uses different vocabulary which makes search for similar patents a daunting task. Patents are lengthy but well-structured documents and contain a title, abstract, description, summary of invention and claims. The claims define the scope of protection granted by the patent. Each patent has a manually-assigned patent classification codes to indicate the technical field or fields it belongs to.

Several approaches have been proposed to improve patent retrieval. Some systems use entire claim text as a query [8] or use information in the patent text to form queries and modify existing retrieval models to improve performance [2, 6]. However, little has been done to explore searching on basis of both classification code information and citations or on different representation of patents.

Kang et al. [7] construct clusters of patents containing same IPC class codes. They employ cluster based retrieval with variations in number and depth of clusters. However, their approach performs marginally better than cluster-less patent retrieval. Harris et al. [5] use USPC classification code hierarchy to find similar patents. They also calculate similarity between two patents using the USPC classification codes. They show that use of hierarchy of classification system results in higher MAP compared to primary codes. In [4], they extend the idea to patents with IPC classification codes. In CLEF-IP task, BiTeM group [3] have used IPC codes to filter patents which do not share at least one IPC code with the query patent. Chen et al. [1] propose IPC-based vector space model for patent search. They use the patent text with the classification codes to construct a document-category vector and find similar patents using several metrics. All the above approaches use the classification information directly to filter or rank documents. They do not combine this information with citations to improve retrieval.

We propose a vector based representation for patents which incorporates both their citations and hierarchical IPC Class information. We use this representation to generate a small set of similar patents. Query patent text is used to re-rank this small set to increase precision.

3. OUR APPROACH

Patents contain meta-data other than text which can be leveraged to improve retrieval accuracy. A patent has manually assigned classification code, defining broad area of the invention. It also cites other patents to discuss similar inventions in the past. Our approach is to combine both the classification and citation information to represent a patent. The two stages of the system are :

1. **Stage 1 :** Converting the query and corpus patents into vectors using IPC codes and citation network. Find similar patents using cosine similarity. We experiment with two variations of cosine similarity.

2. **Stage 2 :** Re-Ranking top K documents using text of the query patent. We use *tf-idf* to select and weigh top 20 words in the query.

We first describe the IPC Classification system and then elaborate on how a patent vector is constructed.

3.1 IPC Classification

International Patent Classification (IPC) system was established under World Intellectual Property Office (WIPO). A patent is assigned to one or more of several IPC codes that indicate the related technical field or fields the patent covers. These codes are arranged in a hierarchical, tree-like structure in four levels. The highest hierarchical level contains eight sections (A-H) corresponding to very broad technical fields. Next, sections are subdivided into classes. Classes are further subdivided into a number of subclasses. Last level, i.e. subclasses are then further divided into main groups and subgroups. Example of all levels in a sample IPC Code is shown in Figure 1.

3.2 IPC Vector Generation

There are mainly two steps in converting a patent into a vector : (1) Creating IPC class vector and (2) Propagation.

IPC class vector : Each patent is assigned one or more of several IPC Class codes. Each code represents information in five levels. The vector is generated by concatenating first three, first four and all five levels of information. First three levels (Section + Class + Subclass) are initial set of dimensions. The next set of dimensions represent information contained in first four levels (Section + Class + Subclass + Main Group). Entire classification code is captured in last set of dimensions. Since the first two levels of an IPC code represent very broad information and may retrieve several documents during similarity calculation, they are not included in the vector. A patent query outside the graph is first converted into a vector by above mentioned process. Vector representation of a sample patent is shown in Figure 2.

Propagation : Citation graph of patents is used to enrich the vector. It is a directed graph which has a link from Node A to Node B if patent A cites patent B. We use the

[1]http://research.nii.ac.jp/ntcir/publication1-en.html
[2]http://www.ir-facility.org/clef-ip

inlinks (incoming edges) of a node to add information to its vector. For a given node P_i, let $In(P_i)$ be subset of the set of nodes that point to it (predecessors) and k be the current iteration. The vector of node P_i for $k+1^{th}$ iteration is defined as follows :

$$P_i^{k+1} = P_i^k + \frac{1}{2^k} \times \frac{\sum_{j=1}^{In(P_i)} P_j^k}{|In(P_i)|} \quad (1)$$

$\frac{1}{2^k}$ is used to dampen the effect of adjacent vectors as the iterations increase. The above formula simply adds the average of vectors of all nodes that point to P_i. Propagation ensures that if Node A is retrieved then its neighbors are also present in the solution set, this improves the recall of the system.

Once a query patent is converted into a vector, similar patents can be retrieved with use of various similarity measures. This phase helps in reducing the search space to a small number of patents. We will show in the next section that similarity calculation on this representation ensures high recall. The retrieved patents are re-ranked by using the text of query patent. A query of 20 words with high *tf-idf* is formed to re-rank the documents.

4. EVALUATION

4.1 Dataset

We use the CLEF-IP 2011 collection of Prior Art Search (PAC) task that has 2.6 million patents pertaining to 1.3 million patents European Patent Office (EPO) with content in English, German and French, and extended by documents from WIPO. There are 300 sample patent applications as queries with the dataset. We use these queries to evaluate our system. The queries have been translated to English from German and French with the help of Google Translator[3]. Both English and original patents are used for making queries. The data has been indexed using Lemur[4] toolkit. All the fields of a patent have been indexed. Of 1.3 million patents, 0.8 million patents cite at least one patent in the corpus and 0.64 million patents are cited by at least one patent. Dimension of concatenated IPC class vector for this dataset is 79963, of which level 1 has 875, level 2 has 8631 and level 3 has 70457 dimensions respectively.

4.2 Similarity Measures

Once each patent has been converted to a vector, there are several ways to calculate similarity between two vectors. We experiment with cosine similarity. We explore two ways of calculating similarity - simple cosine and linear combination of cosine for different levels in the vector. Since each vector is simply concatenation of IPC information at three different levels, we can calculate similarity at each level and linearly combine them to get final score. If P_q is the query patent vector and P_i is a vector of patent in the corpus, we use following to calculate graded similarity :

$$sim(P_i, P_q) = \sum_{j=0}^{3} a_j \cdot sim_{level_j}(P_i, P_q) \quad (2)$$

where a_j represent importance of similarity score at level j and sim_{level_j} is cosine similarity between vectors of $level_j$.

[3]http://translate.google.com

[4]http://www.lemurproject.org/

Two paired t-test were conducted to calculate the statistical significance of the results.

4.3 Evaluation Method

As in the CLEF-IP Workshop, we use the mean average precision (MAP), R@100, R@200 and R@1000 as an evaluation measure. Since our system has two stages, for the first stage we compare the following methods:

Base: Simple Text Retrieval, 20 words, from the query patent, with high *tf-idf* values are used to form a weighted query. The weight of each word is its *tf-idf* score.

COS: Cosine Similarity, IPC information present in the patent is used to make the vector. Entire vector is used to calculate cosine similarity between a patent and query.

GCS: Graded Cosine Similarity, calculating similarity at each level and linearly combining them to get final score.

For the above methods, we limit the value of k (number of iterations in propagation) to 3. This is done to ensure that values of nodes at greater distances do not reach a node. Too many iterations, despite the damping factor of $\frac{1}{2^k}$ will result in several nodes having same vectors. The documents retrieved by above methods are re-ranked in the second stage. We re-rank top 1000 documents by using following methods:

COS + Base: top 1000 documents obtained from COS are re-ranked using $\lambda Base + (1-\lambda)COS$.

GCS + Base: top 1000 documents from GCS are re-ranked using $\lambda Base + (1-\lambda)GCS$.

5. RESULTS AND DISCUSSION

The results of simple and graded cosine at each iteration are shown in Table 1. After the first iteration, the vector of a node represents its own IPC information. As the iterations increase, it incorporates information of neighboring nodes as well. It is evident from Table 1 that the propagation of vectors improves recall significantly. For COS + Base and GCS + Base p value is 0.002 and 0.0002 respectively.

However, linear combination of cosine similarity at each level (Graded similarity)[5] performs better than simple cosine. Each level in the IPC vector covers certain depth of IPC hierarchy. The first level is most general and last level is very specific to an invention. If two patents are similar at first level, it cannot be assumed that they talk about same invention. Whereas, if two patents are extremely similar at the third level, there is a high probability they are talking about either same invention or inventions from the same domain. Hence, linear combination of similarity at each level is a more accurate measure of finding similar patents. We found that level 1, level 2 and level 3 similarity scores when combined with ratio of 0.1, 0.2 and 0.7 respectively give best results.

Text based retrieval is easily outperformed in terms of recall by both simple cosine and GCS even without re-ranking. R@1000 for text retrieval is only 0.443 but for Stage 1 GCS it is 0.680, this confirms that vector representation and propagation improves recall. Stage 1 GCS was not able to retrieve even a single relevant document only for 10% of 300 queries, which is a fairly small number as compared to simple text retrieval. If one looks beyond top 1000 results, this number drops to 3% for top 3000 results.

Importance of re-ranking documents is evident from stage

[5]$0.1 * cos_{l1} + 0.2 * cos_{l2} + 0.7 * cos_{l3}$

Table 1: Comparison of methods in stage 1

Method	k	MAP	R@100	R@200	R@1000
COS	1	0.036	0.200	0.281	0.530
	2	0.047	0.258	0.349	0.603
	3	**0.052**	**0.282**	**0.382**	**0.639**
GCS	1	0.051	0.260	0.370	0.638
	2	0.054	0.278	0.388	0.656
	3	**0.056**	**0.303**	**0.415**	**0.680**

Table 2: Comparison of methods in stage 2

	MAP	R@100	R@200	R@1000
Base	0.078	0.26	0.32	0.443
Base + COS	0.096	0.345	0.417	0.630
Base + GCS	**0.1021**	**0.369**	**0.46**	**0.680**

1 results, as the MAP is still low. We use top 1000 results of third iteration for both the methods in second stage. The value of λ is varied from 0.10 to 0.90 at an interval of 0.1 to combine stage 1 score with text based score. We found that $\lambda = 0.7$ gives the best results. The MAP improves considerably once the documents are re-ranked. This can be explained by contribution of text based score of the document to its final score. When text of query patent is used to query this small set of documents, it improves position of those documents which might not have had high GCS or COS scores but are talking about similar invention. MAP improves for both COS + Base and GCS + Base over the baseline, however GCS + Base performs than COS + Base. The MAP of GCS + Base is 0.10 which is 30% higher than the baseline. R@100 and R@200 have improved significantly when documents are re-ranked. Thus, re-ranking in Stage 2 does improve precision. Overall, the system improves both precision and recall.

6. CONCLUSION AND FUTURE WORK

Finding similar patents is often required in patent retrieval. Approaches proposed in the literature use either IPC information or citations of a patent to retrieve similar patents. In this work, we leverage both IPC information and citations to improve retrieval. We proposed a vector based representation which uses IPC information of patent and its neighbors. The representation proves effective in increasing the recall. We also explored re-ranking of top 1000 documents in second stage to improve precision. Patent text is used to form queries in second stage. Both IPC based representation and re-ranking on sample queries of CLEF-IP 2011 dataset perform better than the baseline i.e. text based retrieval in terms of precision and recall. An extension to this work would be to use a learning-to-rank approach to re-rank top documents. It would be interesting to observe effects of combining both vector representation with patent text to avoid re-ranking.

7. REFERENCES

[1] Y.-L. Chen and Y.-T. Chiu. An ipc-based vector space model for patent retrieval. *Inf. Process. Manage.*, 47:309–322, May 2011.

[2] A. Fujii. Enhancing patent retrieval by citation analysis. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 793–794, New York, NY, USA, 2007. ACM.

[3] J. Gobeill, E. Pasche, D. Teodoro, and P. Ruch. Simple pre and post processing strategies for patent searching in clef intellectual property track 2009. In *Proceedings of the 10th cross-language evaluation forum conference on Multilingual information access evaluation: text retrieval experiments*, CLEF'09, pages 444–451, Berlin, Heidelberg, 2009. Springer-Verlag.

[4] C. G. Harris, R. Arens, and P. Srinivasan. Comparison of ipc and uspc classification systems in patent prior art searches. In *Proceedings of the 3rd international workshop on Patent information retrieval*, PaIR '10, pages 27–32, New York, NY, USA, 2010. ACM.

[5] C. G. Harris, S. Foster, R. Arens, and P. Srinivasan. On the role of classification in patent invalidity searches. In *Proceeding of the 2nd international workshop on Patent information retrieval*, PaIR '09, pages 29–32, New York, NY, USA, 2009. ACM.

[6] I.-S. Kang, S.-H. Na, J. Kim, and J.-H. Lee. Cluster-based patent retrieval. *Inf. Process. Manage.*, 43(5):1173–1182, 2007.

[7] J. Kim, I.-S. Kang, and J.-H. Lee. Cluster-based patent retrieval using international patent classification system. In Y. Matsumoto, R. Sproat, K.-F. Wong, and M. Zhang, editors, *Computer Processing of Oriental Languages. Beyond the Orient: The Research Challenges Ahead*, volume 4285 of *Lecture Notes in Computer Science*, pages 205–212. Springer Berlin / Heidelberg, 2006.

[8] H. Mase, T. Matsubayashi, Y. Ogawa, M. Iwayama, and T. Oshio. Proposal of two-stage patent retrieval method considering the claim structure. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(2):190–206, 2005.