NTCIR-7 Patent Mining Experiments at RALI

Guihong Cao, Jian-Yun Nie and Lixin Shi
Department of Computer Science and Operations Research
University of Montreal, QC, Canada
{caogui, nie, shilixin}@iro.umontreal.ca

Abstract

We participated in the patent mining task at NTCIR7 workshop. Particularly, our experiments focus on English corpus. Based on the Indri search engine, we implemented a patent classification system, which is able to assign a research paper into the IPC system according to the annotated patents in the database. As the task is a cross-genre classification task, we tried several methods to bridge the gap between the research papers and patents. Unfortunately, most the methods do not produce consistent improvements.

1. Introduction

We participated in the patent mining task at the NTCIR-7 workshop. In this task, we are given the abstracts of a set of research papers, and a large set of patents. Each patent has been labeled with an IPC (International Patent Classification) code. The IPC system can be considered as a classification specification, with each code corresponding to one class. Participants are required to assign an IPC code to each research paper. Therefore, the task is actually a text classification task, in which the abstracts of the research papers are test data and the labeled patents are training data.

However, the task is different from the traditional text classification in the following aspects.

Firstly, the terms used the research paper are different from those used in the patent. To widen the coverage of a claim, the author of a patent usually prefers to use more general and abstract terms to describe one concept. On the other hand, the research paper aims to describe a concept in a precise and concise way. It results in different styles of the terms used in patents and research papers. For example, a research paper and a patent may use “Apple ipod” and “music player” to describe the same thing. Therefore, there is a gap between the research papers and patents. As we mentioned, in this task, the research paper and patents are test and training data respectively. However, most traditional text classification approaches make one implicit assumption that the term of texts belonging to the same class should have similar distributions; no matter they are training or test data. Obviously, this assumption does not hold in the task. In fact, the text classification task we are investigating here is called cross-genre text classification (Nanba et al., 2008), i.e., the training and test data are heterogeneous.

Secondly, the classification system (IPC) of this task is much more complicated than the traditional text classification. The IPC system is a global standard hierarchical patent classification system. The sixth edition of IPC has more than 50,000 classes at the most detailed level. In this task, each patent in the training data has been assigned to one or more classes. Every research paper is required to be classified to an appropriate class according its connection with the patents in the training data. Considering the traditional text classification tasks usually have less than 100 classes (Joachims, 1998), the current task is much more difficult. If we use some parametric classifiers, such as SVM, Naïve Bayesian classifier, there would be too many parameters to be computational tractable.

Thirdly, the number of training examples (i.e., patents) in the classes is very unbalanced. We show the statistics of the IPC code in table 1. In this table, column one is the range of the number of the patents of an IPC code, and the second column is the number of this kind of IPC codes. Therefore, there are 25944 IPC codes has less than 11 patents. From this table, we can observe that the IPC codes are very unbalanced, i.e., more than half IPC codes have very few patents. For such a biased training data, it is very challenging to train a precise and robust classifier.

Table 1. Statistics of IPC Codes

#Patent	#IPC	#Patent	#IPC
1~10	25944	2001~3000	5
11~100	10911	3001~4000	3
101~500	1430	4001~5000	0
501~1000	129	>5000	23
1001~2000	46		

In the experiments, the basic classifier we used is K-Nearest Neighbor classifier (*K-NN*) (Duda et al., 2001; Athitsos et al., 2005). When a query instance and a set of labeled instances are given for the classifier, the K-NN classifier finds K nearest neighbors of the query instance, and use majority vote to determine its class label.

This classifier has several advantages. 1). It is a non-parametric classification method and there is no model parameters associated with the classifier. Therefore, it is particularly adapted to the task with many categories. 2). It is intuitive and ready to be implemented, and it also produced comparable performance with the most sophisticated machine learning techniques in previous studies (Song et al., 2007).

The performance of *K-NN* classifier primarily depends on the value of *K* and the method to measure the distance. In our experiments, we represent a patent (the labeled instance) as a probabilistic model with multinomial distribution and the research paper (the query instance) as a sequence of terms. So we measure the distance with the likelihood to generate the research paper with the probabilistic model. However, we will optimize the value of *K*.

We tried to mine the structure of the patent. As a structured document, we expect the different fields in the patent would have different impact on retrieval. In addition, we also investigated the usefulness of term distillation (Itoh et al., 2002) and query expansion.

The remainder of the report is organized as follows: in section 2, we will describe the experimental system we implemented for the task and the basic retrieval models. Section 3 describes the ideas we tried for experiments. In section 4, we present the experimental results, followed by some discussion. Section 5 summaries the report.

2. System Description

We implemented the experimental system based on Indri search engine (Strohman et al, 2005), which is an open source search engine developed in the University of Massachusetts. The retrieval model implemented in the Indri system combines language modeling (Ponte and Croft, 1998) with an inference network (Turtle and Croft, 1991). Similar to the language modeling principle, Indri ranks the relevant documents according to the likelihood of the query generated by the documents. Suppose there is a query *q* consisting of $q_1q_2,...,q_n$, and a document *d*. We consider the occurrence of each query term q_i to be independent from others. The likelihood of the query can be simply calculated with the following equation:

$$P(q \mid d) = \prod_{q_i} P(q_i \mid d)$$

We chose Indri for experiments is because it supports building separate index for individual fields in one document, which makes it convenient to mine the structure of the patents. Readers interested in Indri system can refer to (Strohman et al., 2005) for more details.

With the *K-NN* algorithm, the classification problem is actually to rank the class label. After retrieving top *K* patents with the highest likelihood to generate the query, we extract the IPC code for each patent, and ranking the IPC code with K-NN algorithm. More formally, let *d* denote a patent, *q* for a query, and c for an IPC code. We then rank the IPC code with the following equation:

$$score(c,q) = \sum_{i=1}^{K} \delta(ipc(d_i) = c) P(q \mid d_i) \quad (1)$$

where $\delta(x)$ is an indicator function, which equals to one when *x* is true and zero otherwise, *ipc(d)* is the IPC code of *d*. With equation 1, the IPC ranking problem is simply reduced to the calculation of the query likelihood, i.e., $P(q|d_i)$. In the next section, we will introduce some ideas aiming to improve the retrieval effectiveness.

3. Our Approaches

3.1 Term Distillation

As we know, the patents have different style with the research papers. For example, such terms like “paper”, “propose”, and “study” may occur frequently in the research papers, but rarely in the patents. These terms are useless for retrieval. They sometimes even hurt retrieval effectiveness. For example, if a patent is about some novel method to manufacture paper, it may be wrongly retrieved for many research papers about other topics because they contain the term “paper”. Therefore, it is better to filter out these terms.

We use a simple method to filter these terms. First of all, we calculate the document frequency (*df*) of each term in the research paper. The *df* is calculated based on all 976 topics used in the experiment, including dry run topics and formal run topics. We then select top 30 terms with the highest *df*. These terms are treated as stop words. Therefore, all these terms are filtered in indexing time and query time.

3.2 Mining Patent Structure

A patent is a structured document, which has up to 4 fields, i.e., title, abstract, specification, and claim. Different fields have different styles to describe even the same concept. For example, the claim field tend to use more general and abstract terms to widen the coverage of the claims, while the terms used in specification field is closer to those in the research paper. Moreover, we find that in the specification field, there are also some sub-fields started with some upper case words. According to our observation, the field usually contains four sub-fields: background, description, summary and drawing. Background is a sub-field to describe the background knowledge of the patent, which reviews some previous studies by citing research papers presenting the work. Therefore, this sub-field is close to the research paper, and is the most fertile field to find the terms used in scientific papers. On the other hand, the drawing sub-field only describes some figures which illustrate the patent, so this sub-field is generally useless for the given classification task.

However, there are some problems to divide the specification field into four sub-fields. The first problem is that some sub-fields are missed. We then simply disregard it. The second problem is that some sub-fields are mixed. For example, the background is mixed with description. We then duplicate the sub-field, one for background and another for description.

We then represent each patent with up to 8 fields, in which we duplicate the specification field and divide it into four sub-fields. The remaining problem is how to combine the text in all the fields. Here we used a very simple method. We call the fields we are planning to use active fields, which are denoted as *A*. Let f be an arbitrary field. Then we have:

$$P(q_i \mid d) = \frac{\sum_{f \in A} tf(q_i, f) + \mu P(q_i \mid C)}{\sum_{f \in A} | f | + \mu} \qquad (2)$$

where $tf(q_i, f)$ is the frequency of qi occurring in *f*, |*f*| is the length of *f*, *P(q_i|C)* is the probability of q_i in the whole collection, and μ is the Dirichlet prior, which is set to be 2000 empirically.

3.4 Query Expansion

Query expansion (Xu and Croft, 1996) is a technique to expand a query with some additional terms related to the query. It has been shown to be effective when the query is short. It seems not suitable in our case because the query is a research paper with title and abstract section, which is quite long. However, our purpose in using query expansion here is not to add additional terms, but to emphasize some important query terms. Because the query is long, many query terms may be only noise. With query expansion, we expect to identify some important terms in the query so stress them.

There are many query expansion techniques, among which pseudo-relevance feedback has shown to be effective across retrieval models (Zhai and Lafferty, 2001). In pseudo-relevance feedback, some top ranked documents in the initial retrieval are considered to be relevant to the query. Then some terms occurring in these documents frequently are extracted to expand the original query. In our experiments, we used the mixture model for pseudo-relevance feedback (Zhai and Lafferty, 2001) because of its state-of-the-art performance. In this model, it assumes the feedback documents are generated with two sources, i.e., a topic model expressing user's information need and a noisy model which is approximated by the whole collection. Both models have multinomial distribution. EM algorithm is used to estimate the topic model by maximizing the likelihood of the feedback documents. We selected top 80 terms with the highest probabilities in the topic model, and add them to the original query.

4. Experiments

Table 2. The Effectiveness of Query Expansion

#Exp. Terms	P@30	P@100	MAP
0	0.0271	0.0047	0.1488
20	0.0274	0.0029	0.1470
40	0.0274	0.0030	0.1451
60	0.0277	0.0029	0.1447
80	0.0277	0.0030	0.1439
100	0.0276	0.0030	0.1456

In the section, we present the experimental results for investigating the techniques we described in section 3. We focused on the English patent mining task; therefore the data we used is USPTO patent data. The main evaluation metrics is Mean Average Precision (MAP). In addition, we also used the precision at top 30 and 100 documents.

4.1 The Effectiveness of Query Expansion

In this experiment, we investigated the effectiveness of query expansion. We indexed each query with all the 8 fields. We compared the results with various number of query expansion terms. The model with no expansion terms is the baseline. Table 2 shows the results.

From the table, we observe that query expansion does not produce any positive effect. Although the change is marginal, every expanded query results in worse result. The reason may be because the query we used here are very long. The average length of query is more than three hundred terms. With such a long query, the user's information need has been described completely, and so the query expansion is dispensable.

4.2 The Optimal Value of K

In this experiment, we examine the impact of the value of K on the retrieval effectiveness. We plot the MAP of different K values in figure 1. From this figure, we can easily observe that the value of K is very critical to the classification performance, which is consistent with (Song et al., 2007). The best results occur when K is around 200.

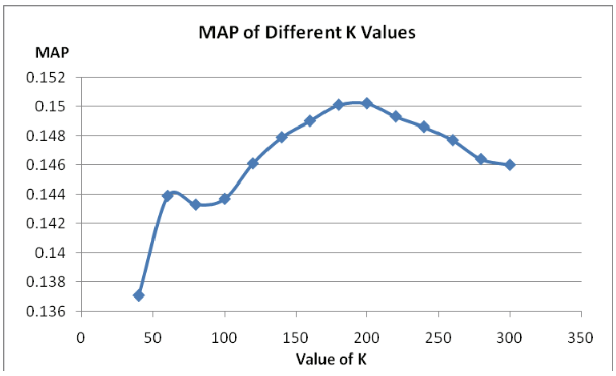

Figure 1. MAP of Different K Values

4.3 The Effectiveness of Term Distillation

This experiment examines the usefulness of term distillation. We argue that some common terms in the research paper does not contribute to retrieve relevant patents, and they sometimes even hurt the retrieval effectiveness. We extracted some words with the highest *idf* in all the topics. Considering the English terms have some morphological transformation, we included all terms' morphological variants. The terms are listed in Appendix 1. All these terms all treated as stop words in query time.

We compared two models in table 3, i.e., with or without term distillation. From this table, we found that the term distillation does not improve retrieval effectiveness. One possible reason is that these terms occur in the patents rarely, and so omitting these terms does not change the performance much.

Table 3. The Effectiveness of Term Distillation

Model	P@30	P@100	MAP
Original	0.0277	0.0047	0.1502
Term Distillation	0.0282	0.0046	0.1491

4.4 Mining Patent Structure

As a structured document, each patent has several fields, and each field aims to realize some specific purpose. We thus expect that individual field has different impact on retrieval effectiveness. As mentioned in section 3.2, we divided each patent into 8 fields. In this experiment, we investigate the effectiveness of various combinations of the fields. For simplicity, we use the following abbreviations to represent the fields:

T: title; **A**: abstract; **S**: specification; **C**: claim;

B: background; **D**: description; **M**: summary; **R**: drawing

We compare the combinations of the fields and the results are shown in table 4. From this table, we observe that alternating the field combination does not result in significant change on the performance.

Table 4. Results of Different Field Combinations

Fields	**P@30**	**P@100**	**MAP**
T+A+S+C	0.0277	0.0047	0.1502
T+A+B	0.0270	0.0041	0.1470
T+A+B+D	0.0281	0.0049	0.1489
T+A+B+D+M	0.0276	0.0047	0.1495

4.5 Formal Results

We submitted two formal results, i.e., rali_baseline and rali_short_doc. The first one used four fields, i.e., title, abstract, specification and claim. The latter used three fields: title abstract and description. The description is only one sub-field of specification; we thus call it “short doc”. For the two results, we set K=200. The result is a bit different from “T+A+B” listed in table 4. It is because we also tune the parameters used for indri to produce the results in table 4. The formal results are:

RUN ID	**P@30**	**P@100**	**MAP**
rali_baseline	0.0234	0.0050	0.1423
rali_short_doc	0.0241	0.0048	0.1437

5. Conclusion

We participated in the Patent Mining task in NTCIR7 workshop. Particularly, our experiments focus on English corpus. Based on the Indri search engine, we implemented a system to classify a research paper into the IPC system according to the annotated patents in a database.

We used K-NN classifier for classification. The distance between the research paper and a patent is measured by the generation probability of the research paper according the patent, wherein the patent is represented by a multinomial probabilistic model. In the experiments, we proposed and evaluated several approaches to improve the classification performance. Most of the approaches aim at more accurate generation probability of the research paper. Unfortunately, none of the approach produced consistent improvement, except the tuning of number of nearest neighbors.

References

Nanba, H., Fujii, A., Iwayama, M. and Hashimoto, T. Overview of the Patent Mining Task at the NTCIR-7 Workshop. In the Proceedings of NTCIR-7 Workshop, 2008.

Duda, R., Hart, P. and Stork, D. Pattern Classification (2^{nd} Edition), John Wiley & Sons, Inc., 2001

Athitsos, V., Alon, J., Sclaroff, S.: Efficient nearest neighbor classification using a cascade of approximate similarity measures. In: CVPR ’05, pp. 486–493. IEEE Computer Society, Washington, DC, USA (2005)

Song, Y., Huang, J., Zhou, D., Zha, H., and Giles, L. IKNN: Informative K-Nearest Neighbor Pattern Classification. In the Proceedings of PKDD, pp.248-264, 2007

Strohman, T., Metzler, D., Turtle, H., and Croft, B. Indri: A language-model based search engine for complex queries. In the online Proceedings of the International Conference on Intelligence Analysis, McLean, VA, May 2-6, 2005.

Itoh, H., Mano, H., Ogawa, Y. Term distillation for Cross-db retrieval, In the Proceedings of Working Notes of the 3^{rd} NTCIR Workshop Meeting, Part 3: Patent Retrieval Task.

Ponte, J. and Croft, B. A language modeling approach to information retrieval. In the Proceedings of SIGIR, pp.275-281, 1998

Turtle, H. and Croft, B. Evaluation of an inference network based retrieval model. ACM Transaction on Information System, 9(3):187-222, 1991

Joachims, T. Text categorization with Support Vector Machines: Learning with many relevant features. In the Proceedings of ECML, Springer, 1998

Xu, J. and Croft, B. Query expansion using local and global document analysis. In the Proceedings of SIGIR’2006, pp.4-11, 1996.

Zhai, C. and Lafferty, J. Model-based feedback in the KL-divergence retrieval model. In CIKM, pp.403-410, 2001.

Appendix 1: The Common Terms in the Topics

lt
gt
paper
papers
method
methods
study
studies
studying
studied
propose
proposed
based
obtain
obtains
obtained
find
found
result
results
prepare
prepares
preparing
prepared
carry
carries
carrying
carried
show
showed
shows
showing
shown
report
reported