NTCIR-8 Research Paper Classification Experiments at Hitachi

Hisao Mase
Hitachi, Ltd.
292 Yoshida-cho, Totsuka-ku,
Yokohama-shi, Kanagawa, 244-0817, Japan
hisao.mase.qw@hitachi.com

Makoto Iwayama
Hitachi, Ltd.
1-280 Higashi-Koigakubo,
Kokubunji-shi, Tokyo, 185-8601, Japan
makoto.iwayama.nw@hitachi.com

ABSTRACT

We report the results of our experiments on the automatic assignment of patent classification to research paper abstracts in NTCIR-8. In mandatory runs, we applied an augmentation of the K-nearest neighbors methods and "Learning to Rank" to improve the classification accuracy. The results show that these methods slightly improve the classification accuracy. We also compared the accuracy by technical fields and the results show that the accuracy differs.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – linguistic processing.

General Terms

Documentation

Keywords

classification of research papers, patent retrieval, k-nearest neighbors method, learning to rank.

1. HITACHI'S APPROACH IN NTCIR-8

One of the subtasks in the NTCIR-8 Patent Mining Task[1] is to assign appropriate International Patent Classifications (IPCs) to research paper abstracts in a fully automatic manner. We used the K-nearest neighbors (KNN) method as a baseline (HTC01) of automatic classification. Our system identifies IPCs through the following steps:

(1) Terms (all nouns, verbs and adjectives) are extracted from an input abstract text using the Chasen[2] morphological analysis tool.

(2) A weight for each term is calculated using a general term frequency-inverted document frequency (TF-IDF) method.

(3) The top K similar patent documents are retrieved from a patent document database using a similar document retrieval engine, GETA[1][3].

(4) The IPCs assigned to each of the K patent documents are identified.

(5) For each of the identified IPCs, the retrieval scores of the patent documents with the IPC are summed.

(6) The IPC scores are sorted in descending order. The top X IPCs are assigned to the input abstract text.

First in this task, we applied the IPC score calculation methods in the KNN method considering (a) the retrieval rank, (b) co-occurrence of IPCs, and (c) the selection of K similar patent documents.

Then, we applied "Learning to Rank" using the IPC scores as features. We used both research papers and patent documents as training data.

2. IPC SCORE CALCULATION IN KNN METHOD

2.1 Score calculation using retrieval rank

In our baseline described in Section 1, the score of a category is calculated using the following formula:

$$CS_j = \sum_{i=1}^{K} (W_{ij} \times DS_i),$$

where CS_j is the score of category-j, W_{ij} is a flag (if category-j is assigned to document-i, W_{ij}=1 or otherwise W_{ij}=0), DS_i is the retrieval score of document-i, and K is the number of similar documents used in the KNN method. In this formula, the retrieval rank is not fully reflected in IPC scores when there is little difference between the retrieval score of a top-ranked document and that of a Kth document.

Thus we used the following formula in this method while considering the retrieval rank of similar documents (HTC02):

$$CS_j = \sum_{i=1}^{K} (W_{ij} \times \frac{DS_i}{\log(1+rank_i)}),$$

where $rank_i$ is the retrieval rank of document-i.

2.2 Considering category co-occurrence

In most research papers, two or more IPCs are assigned. Thus, we focused on the co-occurrence of categories in one document. If category-A is assigned to document-X and if category-B is also assigned to document-X, then category-A and category-B are assigned co-occurrently.

We calculated the rate of co-occurrence of two IPCs using 10-year training patent documents. We used the pairs of IPCs whose rate of co-occurrence is more than a given threshold to tune the IPC score. The score of IPC-A is added by 30% if IPC-A is co-

[1] GETA is a research effort in the "Innovative Information Technology Incubation Project" promoted by the Information-technology Promotion Agency, Japan (IPA).

occurrent to the IPC-B whose rank is Nth or higher (HTC03). Note that the score of IPCs of Nth or higher is also increased by 30%.

2.3 Selection of K similar patent documents

In the conventional KNN method, top K similar documents are used to identify categories. In this case, because the number of categories output is limited, some categories to be assigned cannot be output. If the value of K increases to improve recall, precision decreases.

Thus, in this method, we dynamically changed K documents to improve recall while maintaining precision (HTC04). We used the following algorithm:

(1) 10000 similar patent documents to a query paper abstract are retrieved.

(2) Top K documents are selected from 10000 documents.

(3) N categories are identified using the KNN method.

(4) N categories are deleted from the category list of 10000 similar patent documents. If no category remains in a document, the document is deleted.

(5) Step(2) through step(4) are iterated until the number of categories identified in step(3) reaches 1000.

3. LEARNING TO RANK BASED ON KNN METHOD

3.1 Training data

We applied the “Learning to Rank” approach to research paper classification. Sufficient training data is necessary for this approach. However, we could only use 1071 research paper abstracts (NTCIR-7 dry run data, NTCIR-7 formal run data and NTCIR-8 dry run data).

We applied 97 research paper abstracts used in the NTCIR-7 dry run as training data of learning to rank, and 974 abstracts were used for the evaluation.

We also used patent documents for queries in the training. We selected 1000 patent documents as training queries for each main class. We generated model data for each main class.

3.2 Features

The features used in learning to rank were limited in the mandatory run. The bibliographic data on, for example, authors, publication dates and sources could not be used. Thus, we used IPC scores obtained by the KNN method with the combination of the following four kinds of parameters:

(1) The scope of query term extraction
  (a) Only the title
  (b) Only the body of the abstract
  (c) Both the title and body of the abstract

(2) Retrieval target
  (a) Full text of patent documents
  (b) Abstract of patent documents

(3) IDF calculation
  (a) Full text of patent documents
  (b) The title and body of abstracts

(4) Query data for training
  (a) The title and body of abstracts
  (b) The abstract of patent documents

We used 6 kinds of feature sets (HTC05 through HTC10) shown in Table 1. Feature 1 is the IPC score obtained by the KNN method in which the three methods described in Section 2.1, 2.2 and 2.3 were applied to the baseline method described in Section 1. The method described in Section 2.1 was also applied to features 2 through 7.

In HTC10, we first assigned the main class to a query paper abstract. Then, the model data corresponding to the top-ranked main class was used in learning to rank to decide the final rank.

Table 1. Feature set used in learning to rank

#	Query data for training	Features*						
		1	2	3	4	5	6	7
HTC05	Papers (97 docs)	○	○					
HTC06		○		○				
HTC07		○	○	○				
HTC08		○	○	○	○	○		
HTC09	Papers (1071 docs)	○	○	○	○	○		
HTC10	Patent abstracts (1000 docs)						○	○
HTC11	-	Score merging of HTC08 and HTC10						

*IPC score obtained by KNN method with the following parameters:
1 qte=title&abstract, rt=full text of patents, IDF=full text of patents
2 qte=title, rt=full text of patents, IDF=full text of patents
3 qte=title, rt=abstract of patents, IDF=full text of patents
4 qte=title&abstract, rt=abstract of patents, IDF=full text of patents
5 qte=abstract, rt=abstract of patents, IDF=full text of patents
6 qte=abstract, rt=full text of patents, IDF=paper abstract
7 qte=abstract, rt=abstract of patents, IDF=paper abstract
NOTE: qte=scope of query term extraction, rt=retrieval target

3.3 Tools

We used free software, SVM^{rank}[4], or the Support Vector Machine for Ranking, developed by Cornell University.

4. EVALUATION

We evaluated the effect of the methods using 879 NTCIR-7 formal run queries, 95 NTCIR-8 dry run queries, and 549 NTCIR-8 formal run queries. The results are shown in Table 2, 3 and 4. IPC score calculation methods (HTC02, HTC03 and HTC04) described in Section 2 were slightly effective in a comparison with the baseline (HTC01) in all query datasets.

However, the effect of learning to rank varies. It is slightly effective in the sub-group and main group but not effective in the sub-class. In the sub-group it is only somewhat effective in NTCIR-7 formal run queries and NTCIR-8 formal run queries, while it is not effective in NTCIR-8 dry run queries. Furthermore, using patent documents as queries for training is more effective than using paper abstracts.

Table 5 shows the MAPs according to IPC Section (query datasets of NTCIR-7 formal run, NTCIR-8 dry run and NTCIR-8 formal

Table 2. Evaluation results (sub-group)

ID	MAP		
	7 Formal 879 queries	8 Dry 95 queries	8 Formal 549 queries
HTC01	0.4334	0.4329	0.4427
HTC02	0.4425	0.4437	0.4419
HTC03	0.4434	0.4453	0.4425
HTC04	0.4495	0.4533	0.4512
HTC05	0.4512	**0.4585**	0.4503
HTC06	0.4528	0.4472	0.4487
HTC07	0.4536	0.4474	0.4484
HTC08	0.4536	0.4503	0.4492
HTC09	-	-	0.4506
HTC10	0.4569	0.4347	**0.4539**
HTC11	**0.4575**	0.4381	0.4525

Table 3. Evaluation results (main group)

ID	MAP		
	7 Formal 879 queries	8 Dry 95 queries	8 Formal 549 queries
HTC01	0.5851	0.5801	0.6263
HTC02	0.5991	0.5837	0.6286
HTC03	0.6000	0.5893	0.6290
HTC04	0.6067	0.5986	0.6388
HTC05	0.6070	0.5989	0.6373
HTC06	0.6062	0.5951	0.6418
HTC07	0.6061	0.5922	0.6409
HTC08	0.6046	0.5960	0.6397
HTC09	-	-	0.6387
HTC10	0.6066	**0.5994**	**0.6429**
HTC11	**0.6085**	0.5979	0.6410

Table 4. Evaluation results (sub-class)

ID	MAP		
	7 Formal 879queries	8 Dry 95 queries	8 Formal 549queries
HTC01	0.7499	0.7841	0.7830
HTC02	0.7571	0.7827	0.7894
HTC03	0.7577	0.7849	0.7892
HTC04	0.7661	**0.7919**	**0.7981**
HTC05	**0.7675**	0.7895	0.7930
HTC06	0.7663	0.7891	0.7932
HTC07	0.7661	0.7908	0.7941
HTC08	0.7656	0.7858	0.7913
HTC09	-	-	0.7848
HTC10	0.7661	0.7747	0.7918
HTC11	0.7652	0.7744	0.7937

Table 5. Evaluation results by Section (sub-group)

ID	MAP			
	A,B,D,E,F 470 IPCs	C 544 IPCs	G 1243 IPCs	H 1352 IPCs
HTC01	0.3685	0.3023	0.4551	0.3980
HTC02	0.3677	0.3127	0.4583	0.4029
HTC03	0.3687	0.3126	0.4590	0.4038
HTC04	0.3742	0.3168	0.4658	0.4104
HTC05	0.3786	0.3179	**0.4664**	0.4102
HTC06	0.3798	0.3221	0.4628	0.4098
HTC07	0.3825	0.3226	0.4630	0.4091
HTC08	0.3811	0.3228	0.4634	0.4115
HTC09	-	-	-	-
HTC10	0.3823	**0.3245**	0.4657	**0.4146**
HTC11	**0.3858**	0.3237	0.4656	0.4141

run were used for this analysis). IPC score calculation methods are effective in all Sections. Learning to rank in Section-G is not effective but it is slightly effective in other sections. In the NTCIR-8 formal run query dataset, 35.4% of IPCs assigned to the queries belong to Section-G and 58.4% belong to Section-G, which affects the overall accuracy.

5. CONCLUSIONS

We applied IPC score calculation methods in the KNN method considering the retrieval rank, co-occurrence of IPCs and the selection of K similar patent documents. These methods contributed to slightly improving the accuracy. We also applied the learning to rank approach to the KNN method. This approach is slightly effective in some technical fields but not in others. We should take the characteristics of technical fields into account.

Using the whole text of a research paper to assign IPCs would make for interesting future work. Using bibliographic data such as author names, author affiliations and publication dates and sources to improve classification accuracy in learning to rank could also yield some interesting results.

6. REFERENCES

[1] Hidetsugu Nanba, Atsushi Fujii, Makoto Iwayama, Taiichi Hashimoto: Overview of the Patent Mining Task at the NTCIR-8 Workshop, Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access, 2010.

[2] Chasen: http://chasen.naist.jp/hiki/ChaSen/

[3] GETA: http://geta.ex.nii.ac.jp/e/index.html

[4] SVMrank: http://www.cs.cornell.edu/People/tj/svm_light/svm_rank.html