Patent Document Retrieval and Classification at KAIST

Jae-Ho Kim, Jin-Xia Huang, Ha-Yong Jung, Key-Sun Choi
Korea Advanced Institute of Science and Technology (KAIST) /
National Language Resource Research Center (BOLA)
373-1 Guseong-dong Yuseong-gu, Daejeon, 305-701, Republic of Korea
{jjaeh, hgh, hymanse, kschoi}@world.kaist.ac.kr

Abstract

In this paper, we propose a method to retrieve similar patent documents for a given patent and classify a given patent. We focus on the one of patents' characteristics: "patents are structuralized by claims, purposes, effects, embodiments of the invention and so on." In order to retrieve similar documents from target document set, some specific components to denote the so-called 'semantic elements' such as "claim", "purpose" and "application field" are compared instead of the whole texts.

Keyword: *Patent Retrieval, Patent Classification, Structural Information, kNN, MEM, Hierarchical Classification*

1 Introduction

Existing several statistical methods and machine learning techniques can be applied to the patent retrieval and classification. However, as patent documents are a kind of structural documents with their own characteristics distinguished from general documents, these ones should be considered in the patent retrieval and classification.

Japanese patent documents are structuralized into a sequence of normative sections (or large narrative text fields, [1]) for <Bibliography> (or <Front page>), <Abstract>, <Claims>, <Description>, <Explanation of Drawings>, and <Drawings> as Table 1.

Some sections such as <Abstract> and <Description> consist of more detailed components (or elements) which names like [prior art], [application field], [means of solving problems], [effects of invention], [examples of embodiment] and so on. Such detailed elements are used to improve the readability, but their tags are named by the patent applicant and have some variations even though they must have one meaning. In this context, we will call "applicant element" for a detailed applicant-given component and "applicant tag" for their naming.

Table 1. Structure of Japanese patent document.

	<DOCNO>PATENT-JA-UPA-1995-000001</DOCNO>
<Bibliography> [publication date] [title of invention]	<SDO BIJ> (43)【公開日】平成７年 １月６日 (54)【発明の名称】スラリ散布を行う土壌作業機 ......
<Abstract> [purpose] [composition]	<SDO ABJ> 【目的】 スラリの処理と ...... 【構成】 トラクタとスラリを ......
<Claims> [claim1] [claim2]	<SDO CLJ> 【請求項１】 バキウムカーを ...... 【請求項２】 トラクタに対し......
<Description> [industrial application field] [problem to be solved] [means of solving problems] [operation] [embodiment examples] [effects of invention]	<SDO DEJ> 【産業上の利用分野】本発明はスラリ散布を行う土壌作業機に ...... 【発明が解決しようとする課題】このようなスラリを圃場に...... 【課題を解決するための手段】上述のような目的を達成する...... 【作用】本発明のスラリ散布を ...... 【実施例】以下、本発明を採用した土壌作業機について...... 【発明の効果】以上の説明から明らかなように、......
<Explanation of Drawings> [figure1]	<SDO EDJ> 【図１】本発明のスラリ散布を......
<Drawings> [figure1]	<SDO DRJ> 【図１】

Components with applicant tags like [prior art] and [application field] can be more helpful to classify patent documents than ones of the other components because they include more information related to technical background and technical field. Representing the whole patent document and so often used in the <Abstract> section, the contents of [purpose] of invention and [means of solving problems] are as important as <Claims>. It can be said that the two documents

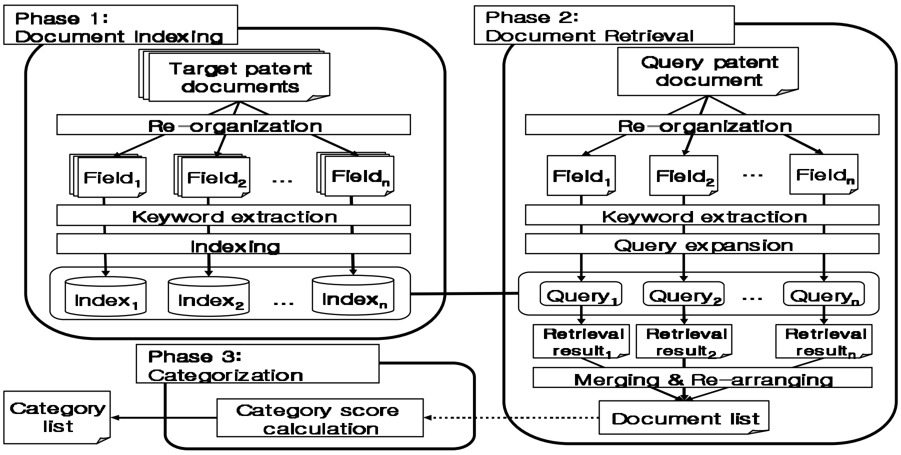

Figure 1. The overall system architecture for patent retrieval and classification

are similar if they are in the same technical classes and have the same (or similar) problem and solution (method). Therefore, if the detailed applicant elements are considered as major features for patent retrieval and categorization, we can achieve good performance. It is the basic claim of this paper.

2 Patent Retrieval & kNN-based Patent Classification

In this section, we describe a method for patent retrieval. In addition, a classification method based on kNN approach using retrieval results is also introduced. By kNN approach, a given patent is classified into the categories of *k* documents similar to it

2.1 Overall System Description

Figure 1 shows the overall architecture of patent retrieval and classification system. The system is composed of indexing phase, retrieval phase and categorization phase.

In the indexing phase (Phase 1 in Figure 1), patent documents in the target document set are indexed in order to retrieve similar documents for a given query document. If you want to classify a query document, training documents that classification codes are assigned to are used as target set. Before indexing, we re-organize and divide each document in the target set by already defined semantic tags. Index files are respectively built by keywords extracted from each divided semantic fields. In Figure 1, they are represented by Field_i. Lemur toolkit (versions 3.1) [2] is used for document indexing and retrieval in this paper.

In the retrieval phase (Phase 2 in Figure 1), we retrieve similar documents for a given query document by using indexing files built in the previous phase. Like the indexing phase, we also re-organize and divide a query document by the already defined semantic tags. Then keywords are extracted from each divided field and then respectively corresponding queries are made for the retrieval. We retrieve similar documents for each query by using indexing files for each semantic field. These retrieval results are merged and then generate a list of similar M documents. This list is re-arranged by comparison of noun-verb pairs to increase precision. It become to be a final result for patent retrieval.

The categorization phase (Phase 3 in Figure 1) is executed only when we try to classify given documents. We assign classification codes to the query documents by using classification codes of similar documents retrieved in the previous phase. When we calculate the score of classification code, the similarity score and the rank of retrieved documents are considered.

2.2 Document Indexing

In the indexing phase, we index patent documents in the target document set in order to retrieve similar documents for a given query document. Before we index, it is necessary to observe the structure of a patent document because a patent document has its own characteristics. As mentioned in the introduction, Japanese patent documents are structuralized into a sequence of normative sections for <Bibliography>, <Abstract>, <Claims>,

<Description>, <Explanation of Drawings>, and <Drawings>. <Abstract> and <Description> sections consist of the more detailed elements which names like [prior art], [application field], [means of solving the problems], [effects of the invention], [examples of embodiment] and so on. While the titles of sections are fixed ones, the names of the detailed elements are applicant-defined ones. Because applicants decide the names of the detailed components with important words that represent the contents of the components, it can be said that names of the applicant elements have applicant-defined meanings. So we call them "applicant tags."

Although applicants write the same content, they can label different tags. Actually, 3,516 applicant-given tags are used from <Abstract> and <Description> sections among 347,227 Japanese patent documents issued in 1993. Table 2 shows the examples of applicant tag with high frequency. Most of them have a low frequency while several tags used a lot of times in the patent documents.

Table 2. Top 10 applicant tags extracted from Japanese patent documents issued in 1993.

Frequency	Applicant tag (Japanese)	Applicant tag (English)
346,157	実施例	Embodiment example
335,300	構成	Composition
330,757	産業上の利用分野	Industrial application field
311,015	従来の技術	Prior art
310,276	課題を解決するための手段	Means of solving problems
309,026	目的	Purpose
307,602	発明の効果	Effects of invention
306,350	発明が解決しようとする課題	Problem to be solved
243,012	作用	Operation
176,676	表	Table

From Table 2, we can infer that patent applicants have a tendency to describe their inventions under the same tag naming for the important components. But, in order to utilize these applicant tags for our purpose, we should classify them into several fixed classes of 'semantic tags'. Firstly, we extract head nouns from applicant tags by using heuristic rule, the last simple noun of tag is a head noun (e.g. NのN_{head} (N_{head} of N), ~るN_{head} (N_{head} which ~)). And then we rank head nouns according to their frequency in applicant tags.

1,475 head nouns are extracted from all applicant tags. Note that 100 most frequent head nouns among 1,475 ones are found in 1,940 applicant tags among 3,516 in total. But those 1,940 applicant tags including only 100 high frequent head nouns cover 99.85% of the total cumulative occurrences of applicant tags. It shows why top-frequent head nouns of applicant tags are the crucial feature of tag classification.

We manually classify 1,940 applicant tags into six semantic tags by their top 100 head. Some useless applicant tags such as 式 (equation), 表 (table) and 図 (picture) are not classified and removed. Table 3 shows the examples of classified applicant tags into semantic tags.

It is possible to classify one applicant tag into the multiple number of semantic tags if it has a coordinate conjunction or a pause such as "課題を解決するための手段及び作用 (the means of solving the problem and the operation)".

Although we can classify content without any applicant tag by using machine learning technology, by using the description patterns or keywords in each applicant element, this paper does nothing but classification based on head nouns of the applicant tags. So we ignore other applicant tags unclassified.

Table 3. Examples of classified applicant tags into semantic tags.

Semantic tag	Examples of Applicant tag
Technological field	産業上の利用分野 (Industrial application field) 従来の技術 (prior art) 発明の背景 (background of the invention)
Purpose	発明の名称 (title of the invention) 発明の目的 (purpose of the invention) 発明が解決しようとする課題 (problem to be solved by the invention)
Method	問題点を解決するための手段 (the means of solving the problem) 課題を解決するための手段及び作用 (the means of solving the problem and the operation)
Claim	All titles in the <Claim> part
Explanation	構成 (Composition) 発明の効果 (the effect of the invention) 課題を解決するための手段及び作用 (the means of solving the problem and the operation) 発明の具体的説明 (The concrete explanation of composition)
Example	実施例 (embodiment example) 参考例 (referential example) 実験例 (experimental example)

In summary, Figure 2 shows the re-organization of patent documents into six semantic fields.

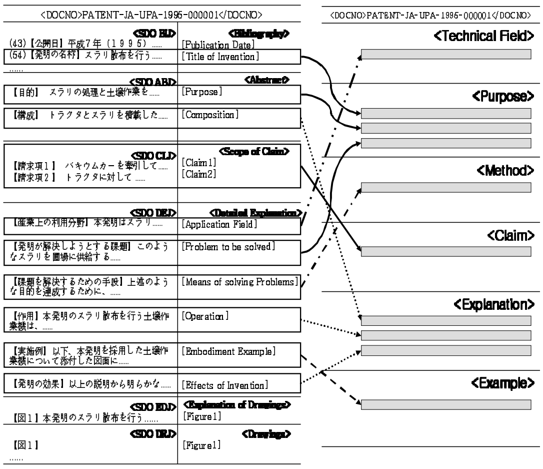
<DOCNO>PATENT-JA-UPA-1995-000001</DOCNO>
<SDO BID>
<Bibliography>
(43)【公開日】平成7年（1995）……
[Publication Date]
(54)【発明の名称】スラリ散布を行う……
[Title of Invention]
<SDO ABD>
<Abstract>
【目的】 スラリの処理と土壌作業を……
[Purpose]
【構成】 トラクタとスラリを積載した……
[Composition]
<SDO CLD>
<Scope of Claim>
【請求項1】 バキウムカーを牽引して……
【請求項2】 トラクタに対して ……
[Claim1]
[Claim2]
<SDO DED>
<Detailed Explanation>
【産業上の利用分野】本発明はスラリ……
[Application Field]
【発明が解決しようとする課題】 このようなスラリを圃場に供給する……
[Problem to be solved]
【課題を解決するための手段】上述のような目的を達成するために、……
[Means of solving Problems]
【作用】本発明のスラリ散布を行う土壌作業機は、……
[Operation]
【実施例】 以下、本発明を採用した土壌作業機について添付した図面に……
[Embodiment Example]
【発明の効果】 以上の説明から明らかな……
[Effects of Invention]
<SDO EDD>
<Explanation of Drawings>
【図1】本発明のスラリ散布を行う……
[Figure1]
<SDO DRD>
<Drawings>
【図1】
[Figure1]
<Technical Field>
<Purpose>
<Method>
<Claim>
<Explanation>
<Example>

Figure 2. The re-organization of a patent document by six semantic tags.

Some applicant elements may be deleted due to the ignored applicant tags, and some elements can be assigned to more than one semantic fields due to the multiple classifications of applicant tags. Keywords are extracted from each element and built index files respectively for the retrieval. We restrict keywords to single nouns.

2.3 Document Retrieval

In the retrieval phase, we retrieve similar documents for a given query document. Like the indexing phase, the query document is re-organized into six fields with the already defined six semantic tags. That means six queries are generated for the retrieval, while the weights of the keywords are assigned by the term frequency of the query document.

The unimportant terms are deleted from the keywords of the query. 67 stopwords are collected by hand from 500 nouns with high document frequency (e.g. "こと(thing)", "発明 (invention)", "目的 (object)", "問題 (problem)", "課題 (problem)", "請求 (claim)", "記載 (mention)")

When retrieving the similar ones for the given query patent, each field of the meaningful pairs of semantic tags are compared instead of the whole texts. It can be said that the two documents are similar if they are in the same technical classes and have the same (or similar) problem and solution (method). The simplest way of similarity computation is to retrieve the target documents whose semantic fields are respectively similar pair-wise to ones of the query document.

However if we compare exclusively between semantic fields with the same tag in a pair-wise manner, the retrieval performance can be worse because of the following reasons.

1. To enlarge the scope of invention, vague or general terms are often used in claims. If we compare the claim of the query with that of the target document, the recall goes down.
2. We cannot fully trust user-defined applicant tags because the described content can be different from the one whose applicant tag represents. For example, some writers describe problem and method together even under the tag name of "the problem of the invention."
3. We cannot fully trust semantic tags because they are semi-automatically classified based on the head nouns. The semi-automatic process can cause an error. For example, although "課題の説明 (i.e., explanation of the problem)" should be classified into 'Purpose', it is classified into 'Explanation' by the classification method based on the head nouns described in this paper.

Therefore we allow cross comparison like Figure 3. 36 retrieval results are produced by cross comparison and merged by equation (1).

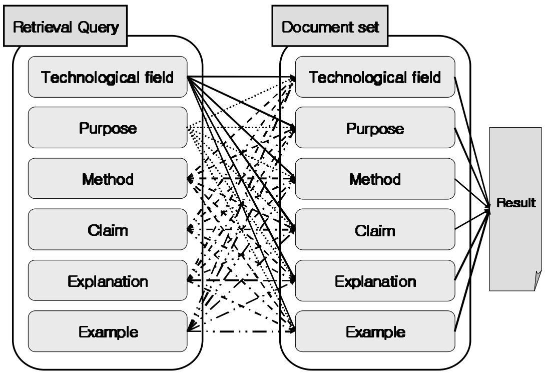

Figure 3. Cross comparison to retrieve similar documents (N-to-N mapping).

$$R(Q,T) = \sum_{i\in\{t,p,m,c,\exp,exa\}} \left\{ w_i \cdot \sum_{j\in\{t,p,m,c,\exp,exa\}} w_{ij} \cdot R(Q_i,T_j) \right\} \qquad (1)$$

We let $R(O_i,T_j)$ to be a retrieval result retrieved from the index file T_j for the query Q_i. For example, $R(O_t,T_m)$ is a retrieval result retrieved from the "Method" indexing file for the "Technical Field" query. The retrieval result has similarity scores for N retrieved documents. w_i is the weight value for the query O_i and w_{ij} is the weight value for the index file target T_j when query is O_i. All weights are given to be equal in the paper. Firstly, six results retrieved from six index files for one query are merged. This procedure is repeated for six queries. Six merged results are merged again and then a list of similar M documents for a query document is generated. Cross comparison brings an escape from the error of the previous classification of semantic titles.

In order to increase the precision, post processing is executed. We re-retrieved N relevant documents from M documents with a new query built by Noun-Verb pairs. A noun and a verb are extracted when they are located under a syntactic relation in a sentence. Final similarity scores for relevant documents are calculated by the equation (2).

$$score = score_{original} + \beta \cdot score_{re-retrieval} \tag{2}$$

M relevant documents are the final result of patent retrieval. In case of the patent classification, this list can be used as an input in the next phase.

2.4 Document Categorization

When we classify patent documents, the categorization phase is executed.

In categorization phase, classification codes are proposed for the given query document by using the similar documents (D) retrieved in the previous phase. When we calculate the score of classification code, the similarity score and the rank of retrieved documents are considered like equation (3).

$$CategorySc\ ore(c, D) \\ = \sum_{\{d \mid d \in D, category\ of\ doc\ d\ is\ c\}} \{DocScore\ (d) \times weight\ (d)\} \tag{3}$$

$$weight\ (d) \\ = \begin{cases} 1 & rank\ (d) \le k \\ \alpha & k < rank\ (d) \le N\ \ (0 \le \alpha < 1) \end{cases} \tag{4}$$

CategoryScore (*c,D*) is calculated for each category and then the given query document is assigned to the category *c* that has the highest value in all of *CategoryScore* (*c,D*). *DocScore*(*d*) is the similarity score of retrieved document *d* and *weigh*(*d*) is the weight of document *d* where they are included in the category *c*. *Weight*(*d*) is changed into 1 or *α* according to the *rank*(*d*), retrieved order of *d*.

3 MEM-based Patent Classification

There are two main issues in text classification. The first one is feature selection, and the second one is about classification algorithm and approach.

3.1 Feature Selection

One of the characteristics of supervised patent classification is its high dimensionality of the feature space, especially if we want to utilized all existing patents as training data. To reduce the feature space, we used several weighting functions to evaluate term-goodness, and removed non-informative terms from the feature space.

We compared three weighting functions in experiments, including Term Frequency (TF), Term Frequency/Inverted Document Frequency (TF/IDF), and Term Frequency/Inverted Category Frequency (TF/ICF), and adopted the best one in the formal run.

TF and TF/IDF are two simple weighting functions used to be adopted in the information retrieval and text classification area. Similar to TF/IDF [3], TF/ICF assumed a word *w* is an important indexing term for a document *d* if *w* occurred frequently in *d* (with high TF) and *w* which occurred in many categories is rated less important due to its low inverse category frequency (see equation (5) and (6)).

$$TFICF(w_i^{(d)}) = TF(w_i, d) \cdot ICF(w_i) \tag{5}$$

$$ICF(w_i) = \log(\frac{|C|}{CF(w_i)}) \tag{6}$$

We used both a fixed threshold (t∈[0, n], where t is threshold, n is an integer from empirical experiments) and a floating threshold according to average value of term weights (t∈[avg($TFICF(w^{(d)})$)-m, avg($TFICF(w^{(d)})$)+m], where m is an integer from empirical experiments) to eliminate non-informative terms in our experiments, and used the one with better result in the formal run.

3.2 MEM-based Classification

MEM-based approach has been used in varies natural language processing areas, and demonstrated reasonable results in text classification [4].

In MEM-based classification, we treated each patent as one event (which means one example in training data), and built a training model with all events from training data. We applied the same approach on both theme and F-term classification in dry run, and submitted only theme classification in formal run. We used Zhang's

MEM toolkit in our model training and MEM-based classification experiments [5].

3.3 Hierarchical-based Classification

Hierarchical classification has been used in many big category classification systems [6,7].

There were more than 2,000 classes in theme classification, much more than most of the general domain classification. We assumed if we separated theme classes to two level, that was, let the first level include 40 classes from 2B to 5L, and the second level include full theme classes, the final classification performance could be improved with much less classes in each level.

4 Experiments and Discussions

4.1 Experiments on Patent Retrieval

We submitted six runs to the Document Retrieval Subtask of NTCIR-5 Patent Retrieval Task. The collection was a publication of unexamined patent in 1993-2002. Firstly, we retrieved 6,000 documents in the document retrieval phase and then re-retrieved 1,000 relevant documents from 6,000 documents with a new query built by Noun-Verb pairs. In this experiment, we didn't execute cross comparison due to lack of time.

Table 4 shows the evaluation results about submitted runs.

Table 4. The submitted results in the Document Retrieval Subtask (MAP).

RunId	Condition in eq. (2)	Topics			
		a.ntc4	b.ntc4	a.ntc5	b.ntc5
baseline	β=0.00	0.1362	0.1286	0.1419	0.1181
d0010	β=0.00	0.1576	0.1488	0.1642	0.1366
d0011	β=0.10	0.1620	0.1473	0.1675	0.1396
d0012	β=0.15	0.1655	0.1489	0.1647	0.1368
d0013	β=0.20	0.1621	0.1453	0.1608	0.1334
d0014	β=0.25	0.1608	0.1397	0.1591	0.1306
d0015	β=0.30	0.1594	0.1381	0.1573	0.1286

Baseline means that the performance achieved by indexing files and queries constructed by keywords extracted from full documents. The performance of our proposed system was better than the one of our baseline system. β is the ratio to reflect the result of the post-processing, re-retrieval by Noun-Verb pairs. We can observe that the performance is improved a little thanks to the post-processing. The performance of the retrieval result was low because high-level text processing like a query expansion was not executed in this experiment.

4.2 Experiments on kNN-based Classification

We submitted seven runs to the Theme Categorization Subtask of the Classification Subtask. Among them, five runs were kNN-based results and two runs are MEM-based results.

We used only two years patent documents issued in 1993 and 1997 as training data and didn't execute cross comparison due to lack of time. The retrieval results of three pairs, (technological field, technical field), (purpose, purpose), (method, method), for query and indexing file are merged with equal weight values.

Table 5. The submitted results in the Theme Categorization Subtask.

RunID	Condition	MAP
ft001	k=10	0.6872
ft002	k=20	0.6842
ft003	k=30	0.6819
ft004	k=50	0.6744
ft005	k=100	0.6666

Five results are differently produced according to the value of k in the equation (4). Table 5 shows the evaluation for submitted results. The performance was the best when we classify query documents by using 10 similar documents retrieved in the training set.

In the small-scaled development set built to evaluate our system, the performance of our system using some detailed applicant elements were better than the one of the system using full text as index file. And cross comparison of specific semantic fields brought better performance rather than a straight pair-wise comparison between semantic fields.

4.3 Experiments on MEM-based Classification

In MEM-based classification experiments, there were several criteria we had to figure out which one was the best. The first one was a weighting function, we wanted to figure out the best weighting function for feature selection among TF, TF/IDF and TF/ICF. The second one was threshold for non-informative term elimination, we needed to find out if fixed cutting threshold to all events, or a floating threshold for each event according to event's average term weight would be better. We considered both MAP and the feature space size in our evaluation.

In our experiments, TF/ICF showed slightly better MAP than TF/IDF with the same feature size, in another word, TF/ICF required smaller feature size than TF/IDF to reach the same MAP. Both TF/ICF and TF/IDF showed better MAP than TF criteria, and floating threshold according to average term weight for each patent was better than fixed threshold. Eliminating non-informative terms according to weighting function threshold was benefit to MAP enhancement but not only feature space reduction.

Table 6. MEM-based classification for Theme Categorization Subtask in formal run

RunID	Condition	Training data size (GB)	MAP
ft006	TFICF, avg-2	1.12	0.3776
ft007	TFICF, avg	0.27	0.3709

Two runs in our formal run submission were from MEM-based classification. One was ft007, it adopted TF/ICF as weighting function and a average TF/ICF term weight as threshold for each document. The fp006 run adopted a more tolerant threshold, which was average minus 2 got from empirical experiments.

Compared Table 6 to Table 5, we can observe that there were sharp contrasts between the results from MEM-based and kNN-based classification. To our understanding, this kind of contrasts came from how we built the training data. As we have mentioned, we threw all existing patents into training data in MEM-based classification, while we utilized only similar patents for given topic in kNN-based classification. It seems that, compared to use all existing patents as training data and just to eliminate non-informative terms from the training data, kNN method could filter noisy data more efficiently by eliminating patents which were not that similar to the topic from the training data at the first place.

We didn't have time to apply the same approach on F-term classification in formal run before submission. But in our self-complement experiments, the MAP on F-term formal run data was 0.4001 (ffx01 in Table 7), in which we extracted feature with TF/ICF weighting function and the threshold was "avg-2", wich condition was exactly the same with the one we used in dry one except we used a little. According our expriment results in both dry-run and formal-run, including our self experiment results on formal run data (Table 7), MEM-based classification for F-term seems relatively better than the one for Theme classification. It might because, there were only about 10 classes in F-term classification, when there were more than 2,000 classes in Theme one, and MEM-based classification approach with non-informative term elimination noisy filtering is more suitable for small category classification task, when kNN-based noisy filtering approach is better for big category classification task. But this is only our assumption which needs more experiments to verify.

Table 7. MEM-based classification for Theme and F-Term Categorization: comparison

Run	ID	Condition	MAP	Top MAP from other teams/runs
Theme	dt001	TFICF, avg-1	0.3776	0.6928 (dry run) 0.6872 (formal run)
	dt002	TFIDF, avg	0.3709	
F-term	df001	TFICF, avg-1	0.4819	0.4819 (dry run) 0.4998 (formal run)
	ffx01	TFICF, avg-2	0.4001	

4.4 Experiments on Hierarchical-based MEM Classification

We randomly chose 877 patent documents from year 1997 as our test data, and 7338 documents form year 1993 to 1996 as our training data. We classified patents into 40 top level classes from 2B to 5L, and then classified them again to full theme classes according to first level classification results.

From Table 8, we can observe that, different from our expectation and other previous results in general domain hierarchical classification [6,7], the MAP in our experiments was lower than the one of non-hierarchical MEM-based classification. This result was not submitted.

Table 8. Hierarchical classification results

	Hierarchical	Non-hierarchical
MAP	0.2907	0.3229

5 Conclusions

This paper showed that the semantics of patent document structure is one of important features for the categorization purpose. However we could not verify the semantics of patent document structure is also valuable in the patent retrieval due to low performance. Further examinations of our methods are needed in the future.

In the aspects of classification techniques, we experimented the kNN, MEM and SVM on the semantic re-organized structure. SVM was good for small-scale of prototype experimentation but we could not catch up with the timing limitation of this task. But kNN approach always outperformed MEM in our experimentation, especially for the theme categorization subtask. The current conclusion is that the transparent semantic handling was possible in kNN but not in the other methods.

References

[1] L. S. Larkey. A Patent Search and Classification System. Proc. DL-99, 4th ACM Conference on Digital Libraries, 1999.

[2] The Lemur toolkit for language modeling in information retrieval. http://www.lemurproject.org/

[3] Thorsten Joachims. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In the Proceedings of the Fourteenth International Conference on Machine Learning, 1997

[4] Kamal Nigam, John Lafferty, Andrew McCallum. Using Maximum Entropy for Text Classification. In IJCAI-99 Workshop on Machine Learning for Information Filtering, 1999.

[5] Le Zhang, Maximum Entropy Modeling Toolkit for Python and C++. http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

[6] S. Dumais and H. Chen. Hierarchical Classification of Web Content. Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval (SIGIR'2000), 256-263, 2000.

[7] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. Proceedings of the 14th International Conference on Machine Learning (ICML'97), 170-178, 1997.