Learning to Map Query Terms to Document Categories i Adaptive Information Retrieval

Rey-Long Liu
Dept. of Information Management
Chung-Hua University
HsinChu, Taiwan, R.O.C.
Email: rlliu@mi.chu.edu.tw

John Chien
Institute Information Engineering
Chung-Hua University
HsinChu, Taiwan, R.O.C

ABSTRACT

Information retrieval systems (IRS) often emplo inverted files to map query terms to those documents that contain the query terms. An inverted file consists of a set of terms and serves as an index to specific documents. However, *selecting* the terms and then *mapping* the terms to relevant documents are major bottlenecks. Manually selecting and map ping the terms often suffer from the problems of high cost and incomplete inverted files, since almost all terms (except for the small amount of stop words such as 'an' in English) may be meaningful to individual users. Furthermore, a document containing a term does not necessarily be relevant to the term. In this paper, we argue that there should be an incrementall extensible inverted file to map query terms to their suitable document categories in which relevant documents are more likely to be found for the query We propose a machine learning technique to acquire this kind of inverted files. The technique works on hierarchically structured text databases and acquires the way of mapping unknown terms to their suitable document categories. Thus the IRS ma adapt its search strategy to both the text database and the individual users' queries. This kind of *adaptive information retrieval* may promote both the quality and the efficiency of IRS, since full-text searching is conducted in suitable and smaller search spaces. The technique is theoretically evaluated. Its performance is empirically investigated using a real -world text database on the World Wide Web.

Keywords: Term-to-Category Mapping, Machine Learning, Adaptive Information Retrieval

1. INTRODUCTION

The quality and the efficiency of retrieving relevant information are two major concerns of information retrieval systems (IRS). Among the various techniques implemented in current IRS, inverted files are the ones commonly employed to promote the efficienc of information retrieval (IR). By selecting and storing meaningful terms in an inverted file, the IRS may efficiently locate those documents that contain the terms without exhaustive search on the whole text database. An inverted file may thus serve as an index to the text database [21].

However, almost all terms (except for the small amount of stop words such as 'an' in English) may be potentially meaningful to individual users [18]. Thus the *selection* of the terms to be stored in the inverted file becomes a difficult task. Incorporating all potential terms into the inverted file is impractical and computationally expensive. On the other hand, incorporating only a small amount of the potential terms will deteriorate the performance of the IRS, since exhau stive search will be needed when users query terms are not included in the inverted file. Therefore, a better solution should lie between the two ends. Even manual selection is employed, the *completeness* and *extensibility* of the inverted file deserve more exploration in order to promote the efficiency of the IRS.

Furthermore, from the viewpoint of the quality of IR, a document containing a query term does not necessarily be relevant to the term. A query term coming from different users may call for differ ent search spaces. For example, a user may simply input the queries such as "Give me all I need to know about personnel recruiting," "Show me the information about the competitors," and "Describe the monetar policy of our country." For these queries, sear ching should be conducted in different search spaces (or document categories) for those users of different departments, companies, and countries respectively.

Therefore, an inverted file should be able to map query terms to their suitable *document categories* rather than specific documents only. A document category contains a set of documents and thus forms a search space. For example, the documents on the Internet search engines (e.g. Yahoo, Whatsite, and Kimo) are often categorized and structured as a tree. In the document categories corresponding to the quer terms, relevant documents for the query should be more likely to be found. Therefore, given a query term from the user, full -text searching may be conducted in the search spaces (or document categories) corresponding to the query term. Both precision and recall of IRS may thus be improved.

In this paper, we propose a framework for incrementally acquiring the mapping between query terms and document categories. The mappings acquired are then assimilated into an inverted file to promote both the efficiency and the quality of the IRS. The IRS actually performs *adaptive information*

retrieval (Adaptive IR) in the sense that it adapts its search strategy to the users' queries and the text database.

Adaptive IR is a research branch of intelligent IR, which aims at providing solutions to smart document retrieval [2]. Many techniques have been proposed in intelligent IR including case-based reasoning [10], query expansion [1, 12], genetic algorithm [8], artificial neural network [13], vector space model [14], and apprenticeship learning [3]. An adaptive IR system differs from general intelligent IR systems in that it attempts to capture (and then adapts its search strategy to) individual users' preferences. T herefore, adaptive IR systems often incorporate a learning component. Most previous studies focused on the acquisition of the *weights* of the keywords meaningful to individual users [1, 3, 13, 14]. In our work reported in this paper, we are concerned with t he identification of the *search spaces* in which relevant documents are more likely to be found for the keywords meaningful to the individual users.

This paper is organized as follows. In the next section, we propose of the framework. Experiments on a real-world Internet text database are reported in section 3. The framework is then evaluated in section 4. We finally conclude that, by incrementall acquiring the mapping between query terms and document categories, both the efficiency and the quality of IR may be improved.

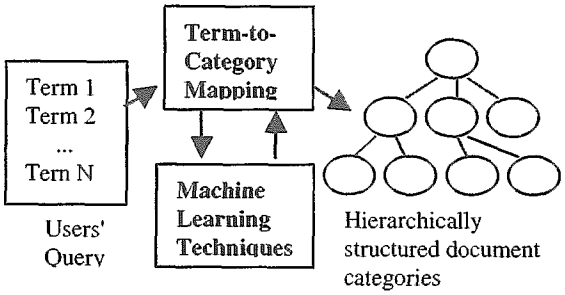

Figure 1. Learning to identify document categories for a query

2. IDENTIFICATION OF DOCUMENT CATEGORIES FOR A QUERY

Figure 1 illustrates the idea of the framework. We assume that the documents are hierarchicall organized as a tree, which is common for most search engines in libraries and the Internet. After accepting a query consisting of a set of query terms, the system tries to identify the suitable document categories in which full-text searching should be conducted. query term may be mapped to several document categories. The document categories corresponding to the query terms are integrated to identify the document categories of the query. Thus it is a kind of *concept retrieval* [17] in the sense that information is retrieved based on the interactions among the query terms rather than treating each query term independently. In section 2.1, we introduce the way of mapping a query to its document categories. In section 2.2, we introduce the technique to acquire the way of mapping a query term to its document categories.

2.1. Mapping a query to document categories

As described above, the documents are hierarchically categorized. The target categories of an input quer are identified based on the hierarchical relationships among the document categories. Given a query term that is mapped to category A, the target document categories of the query are identified according to the following two *heuristic semantic patterns* (HSP)

HSP1: If there are any known terms mapped to A's brother categories, the target category of the query is A's father category.

HSP2: If no other known terms are mapped to A's father category or A's brother category, the target category of the query is category A.

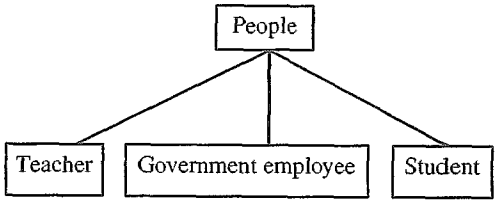

Figure 2. A sample part of a hierarchical text database

As an example, consider the query 教師與公務人員調薪的幅度" (the range of salary adjustment of teachers and government employees). As illustrated in figure 2, suppose in the hierarchical text database, there are thee categories named "teacher," "student," and "government employee." They are brother categories and have the same father category named "people." Suppose the query terms " 教師" and "公務人員" are known terms. They are mapped to the "teacher" category and the "government employee" category respectively. Thus according to HSP1, the target document category of the query is their father category "people," meaning that full-text searching should be conducted on the documents of the "people" category. Being the father category of "teacher" and "government employee," the "people" category should contain more documents concerning both teachers and government employees. Therefore, relevant documents for the query are more likely to be found in the "people" category.

As an example for HSP2, consider the above quer again. Suppose the query term "公務人員" is the only known term (i.e. "教師" is an unknown term), HSP2 will be applied and the target category of the quer will become the "government employee" category. In practice, HSP1 and HSP2 work together in the sense that the document categories of an input query are the union of the document categories hypothesized b HSP1 and HSP2.

2.2. Learning to map query terms to document

categories
HSP1 and HSP2 may be used to acquire the ways of mapping query terms and document categories as well. In this case, the learning technique is based on Explanation-Based Learning (EBL) [19]. The hierarchical relationships among document categories serve as the minimum domain theory in EBL [15, 16]. For example, consider the above query "教師與公務人員調薪的幅度" again. Suppose the target categor is known to be the "people" category "公務人員" is mapped to the "government employee" category, but "教師" is an unknown term. Thus by HSP1, the system will learns to map "教師" to "teacher" and "student" categories.

The identification of the target category of the input query is essential for learning. In practice, it may be achieved in two ways. The first way is to directly ask the user for the target category. This way is effective and has been used in many com mercialized search engines on the Internet (e.g. www.yahoo.com.tw). The user may select a category for restricting the search space. The second way is to observe the user's behaviors in reading the documents retrieved. This technique is based on the releva nce feedback technique in IR [18]. When the user enters a query and then reads the documents retrieved, the categories of the documents being read are likely to be the target category of the query. No extra information from the user is needed.

It should also be noted that, each query term might be mapped to multiple document categories with different strengths. When using HSP1 and HSP2 (in both mapping and learning), the strengths of the results are normalized (into a scale of 1 to 100). A threshold is pr edefined for determining the final categories of the input query (in mapping) and the query terms (in learning).

3. EXPERIMENT

The experiment was designed for investigating the performance of the framework. In particular, we focused on the improvements in the quality and the efficiency of IR.

3.1. Environments of the experiment

We introduce the ways of setting up the document database, the minimum domain theory for learning, and the input queries for the experiment.

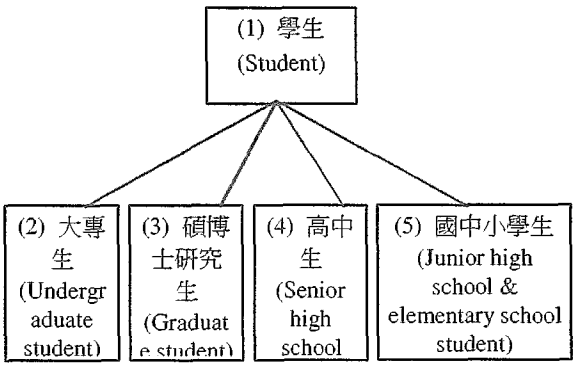

Figure 3. A part of the hierarchical document database in WhatSite

3.1.1. The document database and its structure

The document database was from WhatSite (http://www.whatsite.com.tw). WhatSite was a web site containing a search engine and a hierarchicall structured text database. The real-world database was organized as a tree by WhatSite's professionals. An example subtree was shown in Figure 3. Each node (e.g. node 3 in figure 5) in the tree contained a set of documents of the same category (e.g. graduate students). We used a tree rooted by the "education" category as our underlying database for the experiment. It contained documents about education. There were totally 102 nodes in the tree. For those nodes with very few documents, other Internet search engines were invoked to expand their document sets. There were totally 4885 documents in the tree. That is, on average a node recorded a search space of 48 (4885/102) documents. The depth of the tree was 5.

3.1.2. The predefined domain theory

As noted above, the hierarchi cal relationships among the document categories serve as a minimum domain theory. The relationships were naturally extracted from the tree. Another kind of predefined knowledge was a set of query terms and their preferred document categories. The set of te rms was mainly used for segmenting words in Chinese queries. Word segmentation is a fundamental step of processing Chinese queries, and there have been many methods to segment word successfully. Therefore, to focus on the acquisition of new knowledge, with out loss of generality, we assumed a minimum set of terms as a kind of predefined knowledge. For each document category, a term is inserted into the initial inverted file.

For example, the term "大專生" (Undergraduate students) mapped to category 2 in Figur e 3 was inserted. Since there were 102 categories, there were 102 terms in the initial inverted file. From the viewpoint of machine learning, these terms served as the bootstrapping knowledge for learning new knowledge. The system then tried to locate and acquire missing knowledge (i.e. the mapping of unknown terms to categories).

3.1.3. The queries for training and testing

A set of 120 common queries was constructed for the experiment. Each query consisted of several terms that might be known or unknown for the system. As noted above, the semantics of a query was based on the interactions among its query terms. Each quer was manually tagged with a target document categor and a set of relevant documents. The tagging process was fully independent to the l earning process. Relevant documents did not necessarily come from the same category. The set of relevant documents could be null, if no documents in the database could be relevant to the query.

To objectively conduct the experiments, we avoided the possib le biases in the tagging process b employing a person to read the queries, determine their meanings, select their document categories, and then identify their relevant documents. The tagging task was solely based on his personal judgement and preferences, which are the knowledge the system attempted to acquire. Each query could be tagged with a document category. However, For some of the queries (33 out of 120), no relevant documents could be found in the database of 4885 documents. Most other queries had about 50 relevant documents that could be from different categories. Example queries are listed below for illustrative purposes

(1) 教師與公務人員調薪的幅度
(the range of salary adjustment of teachers and government employees)
Target document category: 人物 (people)

(2) 有關中文打字的補習班
(about the supplementary schools of Chinese typewriting)
Target document category: 課程 (course)

(3) 附有管理學系的院校
(the college with the department of management)
Target document category: 商學院 (business college)

(4) 大專生與研究生的就業狀況
(the status of getting employme nt of undergraduate and graduate students)
Target document category: 學生 (student)

The subtrees (in the case tree) related to the four queries were:

(1) 課程 (course) had two children categories: 補習班 (supplementary school) an 職業訓練 (vocational training);
(2) 獨立院校 (college) had six children categories 商學院 (business college), 理工學院 (college of science and technology), 師範學院 (normal college), 人文及社會學院 (college of humanity and sociology), 醫學院 (college of medicine), an 技術學院 (technical college);
(3) 學生 (student) had four children categories 大專生 (undergraduate student), 研究生 (graduate student), 高中生 (senior high school student), an 國中小學生 (Junior high school & elementary school students).
(4) 人物 (people) had three children categories 軍公教 (soldier and government employee), 學生 (student), and 教師 (teacher).

The queries were used for either training or testing. In training, the associated target document categories were used to acquire the ways of mapping unknown terms; while in testing, the target document categories were used to determine whether the system had correctly identified the target document categories. Although there were on -line tagged queries and databases available (e.g. the TREC corpus), they were not designed for evaluating the adaptability of IR systems. Therefore, to investi gate the system's performance in a real scenario, we used a real-world database on the Internet (i.e. the WhatSite) and constructed the set of common queries. To objectively conduct the experiment, the queries were tagged without considering the documents to be searched.

3.2. The training method

After setting up the databases and the queries, the system was ready for being trained. In the training phase, 100 queries were randomly selected for training, and the remaining 20 queries served as the queries for testing. The same process repeated 50 times to estimate the average performance of the system.

3.3. The testing method

After training, 20 queries (other than the 100 queries for training) were input to the system. The output document categories for each query were compared with the target document category of the query in order to determine whether the system can properly identify the target document category. The performance of the system was compared with that of a traditional IR system without the ab ility of extending its inverted file. The traditional non-extensible IR system was constructed by disabling the learning capability of the IR system.

The reasons of using the non-extensible IR system as the baseline for performance comparison were

(1) Except for the learning component, the non-extensible IR system shared all routines, databases, and queries with AIR.

(2) The no -extensible IR system could represent most previous IR systems whose search strategies were *predefined*. Many previous IR systems with learning capabilities (e.g. [3, 5, 6, 14]) did not adapt their search strategies to the users and the databases. They focused on the adaptive way of determining the relevance of a document to a particular user. On the other hand, our system *acquired* and *adapted* its search strategy. In fact, the two IR systems shared a "small" file of keywords (i.e. the 102 predefined query terms). We investigated the system's performance in expanding the file and adapting the file to the environments (i:e. the user's preferences and the databases).

3.4. Evaluating both the quality and the efficiency of IR

The two systems were evaluated from two aspects: *quality* and *efficiency* of IR. Thus precision and recall in the identification of document categories and relevant documents were defined as follows

(1) *Category precision rate* = 100% * C / A where A is the total number of output categories. Among these categories, C is the total number of categories that are target categories for the test queries.
(2) *Category recall rate* = 100% * C / B, where B is the total number of target categories for the test queries.
(3) *Document precision rate* = 100% * D / E,

where E is the total number of output documents. Among these documents, D is the total number of documents that are relevant to the test queries.

(4) *Document recall rate* = 100% * D / F, where F is the total number of relevant documents for the test queries.

It is obvious that, if the documents in the database were suitably categorized, the improvements on category precision and recall will lead t o the improvements on document precision and recall.

3.5. Results and analysis

The reports and discussions of the experimental results are separated into two parts: (1) category precision and recall, and (2) document precision and recall.

Table 1. Category precision rates, non-extensible vs. extensible (threshold=80)

Random selection	Precision rate	Standard deviation	Improvement
5%	7.80 vs. 9.66	0.58 vs. 0.61	24%
10%	5.10 vs. 8.61	0.48 vs. 0.60	69%
20%	4.07 vs. 7.31	0.40 vs. 0.64	80%

Table 2. Category recall rates, non-extensible vs. extensible (threshold=80)

Random selection	Recall rate	Standard deviation	Improvement
5%	37.40 vs. 45.50	2.56 vs. 2.88	22%
10%	37.20 vs. 46.60	2.90 vs. 2.70	25%
20%	48.70 vs. 50.10	1.70 vs. 1.76	3%

Table 3. Category precision rates, non-extensible vs. extensible (threshold=50)

Random selection	Precision rate	Standard deviation	Improvement
5%	7.78 vs. 10.87	0.76 vs. 0.91	40%
10%	5.31 vs. 9.36	0.70 vs. 0.94	76%
20%	3.48 vs. 8.40	0.40 vs. 1.09	141%

Table 4. Category recall rates, non-extensible vs. extensible (threshold=50)

Random selection	Recall rate	Standard deviation	Improvement
5%	30.90 vs. 34.60	2.71 vs. 2.77	12%
10%	32.80 vs. 36.40	3.28 vs. 3.10	11%
20%	37.50 vs. 38.90	3.07 vs. 3.03	4%

Table 5. Document precision rates, non-extensible vs. extensible (threshold=80)

Random selection	Precision rate	Standard deviation	Improvement
5%	0.45 vs. 0.55	0.09 vs. 0.09	22%
10%	0.49 vs. 0.54	0.10 vs. 0.07	10%
20%	0.40 vs. 0.51	0.07 vs. 0.08	28%

Table 6. Document recall rates, non-extensible vs. extensible (threshold=80)

Random selection	Recall rate	Standard deviation	Improvement
5%	10.29 vs. 18.30	1.86 vs. 2.71	78%
10%	15.93 vs. 20.56	3.26 vs. 3.19	29%
20%	21.75 vs. 21.70	3.87 vs. 3.45	0%

Table 7. Document precision rates, non-extensible vs. extensible (threshold=50)

Random selection	Precision rate	Standard deviation	Improvement
5%	0.62 vs. 0.59	0.18 vs. 0.11	-5%
10%	0.42 vs. 0.54	0.08 vs. 0.09	29%
20%	0.41 vs. 0.49	0.08 vs. 0.08	20%

Table 8. Document recall rates, non-extensible vs. extensible (threshold=50)

Random selection	Recall rate	Standard deviation	Improvement
5%	11.96 vs. 14.97	2.86 vs. 2.90	25%
10%	11.86 vs. 13.50	2.03 vs. 2.05	14%
20%	19.42 vs. 14.06	3.61 vs. 2.53	-28%

3.5.1. Category precision and recall

The experimental results of category precision and recall are summarized in table 1 to table 4. In these tables, two parameters deserve more descriptions. One parameter is the threshold for determining th e output categories for each test query. Since all the

possible categories may have different strengths (ref. Section 2.2), they are sorted in descending order. The system outputs the categories in descending order. The output process terminates when the total strength of the output categories exceeds the threshold.

Obviously, the higher the threshold is, the more categories the system outputs. Therefore, in general, the precision rate for a threshold of 80 is lower than the precision rate for a threshold of 50. On the other hand, the recall rate for a threshold of 80 is higher than the recall rate for a threshold of 50. The improvement on precision is more significant when the threshold is set to 50. This means that the extensible system has been able to effectively acquire the mapping between query terms and document categories.

Another parameter is the percentage of randomly selecting categories (among the 102 categories) when all the query terms are unknown terms for the systems. We set up three values (5%, 10%, and 20%) for the parameter. Obviously, the higher the parameter is, the lower (higher) the precision rate (recall rate) will be. The result also shows that the higher the parameter is, the more significant the improvements on precision will be. This indicates that the acquisition of query terms has successfull located and reduced the number of categories (and hence the size of the search spaces) in IR.

The overall improvements on both precision and recall were significant in the experiment. The differences in standard deviations of the two systems were not significant. The performance levels did not significantly oscillate among the 50 sessions of training and testing.

3.5.2. Document precision and recall

Based on the significant improvements on the identification of document categories of input queries, we further investigated whether relevant documents may be actually found in the document categories identified. Therefore, all the documents belonging to the categories identified were extracted to check relevancy. The results are summarized in table 5 to table 8.

The overall improvements on document precision and recall were not as significant as that on category precision and recall. A detailed analysis showed that this was mainly due to the misclassifications of documents in the real-world text database. Some documents were classified into those categories that they should not belong to. In general, the extensible system improved both document precision and recall. The standard deviations did not have significant effects on the results.

The result also showed that an effective way of mapping query terms to document categories could help the IR system to locate suitable search spaces, and hence promote the efficiency of IR. Furthermore, if an IR system could not locate suitable search spaces, irrelevant information would be retrieved from the inappropriate search spaces. For example, a document that happened to have the term "government employee" does not necessarily be relevant to the query "find the information about government employees." Therefore, the term "government employees" should not only be a keyword in searching, but also be an indicator of suitable search spaces.

The way of segmenting Chinese words in the queries deserves more discussions. Since the system acquires the mapping between query terms and document categories, the idea could be easily ported to different languages. The only difference was that word segmentation (i.e. term segmentation) was needed for some languages such as Chinese. To investigate the "basic" performance of learning, the current experiment did not presume that a "powerful" word segmentation process was available. Given a Chinese sentence, the terms already defined (i.e. known terms, ref. 3.1.2) were filtered out, resulting in several islands of Chinese characters. An island was then treated as a word (i.e. a query term). Obviously, a more effective word segmentation process, which has been successfully developed in many previous studies, may be helpful to reduce the errors in word segmentation.

4. EVALUATION

We have proposed a technique to acquire an inverted file in which the ways of mapping query terms and their suitable document categories are stored. This kind of inverted file may be used to extend traditional inverted files in which only the ways of mapping query terms to specific documents are stored. Since it's often impractical to predefine an inverted file containing all possible term-to-document mappings, the learning technique may be helpful in locating suitable search spaces (categories) for those terms that are not included in the initial inverted file. The major contributions of the framework are the improvements on (1) quality and efficiency and (2) adaptability of IR systems.

4.1. Quality and efficiency of IR

Recall and *precision* often serve as the fundamental criteria for evaluating IR systems. An IR system with good quality should retrieve as many relevant information pieces as possible (i.e. high recall rate), and at the same time, as few irrelevant information pieces as possible (i.e. high precision rate). Simultaneously achieving high precision and recall is a major challenge of IR. Therefore, previous IR systems proposed many ways of reformulating the documents (e.g. [23]) and the queries (e.g. [10, 12]) so that relevant information may be more likely to be found. Although the quality of IR was improved, the efficiency of IR still deserved more improvements, since much effort might be wasted for searching for information in improper spaces.

We improve both the quality and the efficiency of IR by incrementally acquiring the mappings between

query terms and document categories. Therefore, the IR system learns to locate search spaces. Therefore both the quality and the efficiency of IR m ay be improved significantly. More relevant information may be found in smaller search spaces. The technique may be integrated with the previous techniques of reformulating documents and queries. The former indicates suitable search spaces in which the latter may perform complete searching.

4.2. Adaptability of IR systems

It is commonly believed that individual users preferences should be considered by IR systems. An adaptive IR system differs with general intelligent IR systems in that it has a learning component to capture individual users' preferences. Currently most popular search engines on the Internet do not consider individual users' preferences when searching for information. Thus spotlight cues were employed to help the user to focus on those documents that might be of interest for the user [22]. For dealing with different users' preferences, many previous studies employed intelligent agents to retrieve or filter information based on the users' profiles [8]. To automatically acquire the users' profiles, several techniques focused on identification of the hyper-links [3], subject features [6], and document categories [14] that were intere sting to the users. To acquire the users' way of associating specific keywords and documents, connectionist techniques were developed [5]. Other previous studies focused on the acquisition of the weights of the keywords meaningful to individual users [1, 1 3]. These previous techniques focused on retrieving information that is more likely to be of interest to the users.

In our work reported in this paper, we are concerned with the problem how an IR system may adapts its search strategies to both the users' preferences and text database:

(1) Adaptability to the user's preferences Different users may have different information needs and preferences, which may grow and change over time. Therefore, the semantics of the terms in users' queries may grow and change as well. It's impractical to predefine a complete set of query terms for a particular set of users. Therefore, our IR system captures users' preferences by acquiring the semantic preferences of the terms (i.e. term-to-category mapping) in their queries.

(2) Adaptability to the underlying database: The structure of a document database may be organized in different ways in different IRS, although most documents will finally be well fitted into the structure [23]. The structure of the document database may even change over time. Therefore, relevant documents for a query might be found in different categories in different IRS at different time. The learning technique may dynamicall link the user's query terms with their suitable document categories. Thus when the structure of the underlying database changes, the technique may help the IRS to adapt to the current structure by adjusting the term-to-category mappings. No matter what the database structure is, the system may search for information in a space corresponding to the information needs of the users.

The learning technique is based on Explanation-Based Learning (EBL, [19]). Given an initial domain theory, the system may efficientl acquire missing knowledge [15, 16]. We have shown that, for those databases that are hierar chicall organized, the domain theory may be easil constructed based on the structure of the databases. The novel application of EBL to IR may speed up the convergence process, which is often a major challenge in previous EBL systems (e.g. [19]).

4.3. Future research directions

We are extending the framework from the following perspectives

(1) Filtering out inappropriate mappings acquired: The system might acquire inappropriate mappings for unknown terms. These mappings may be filtered out by observing thei r performance in problem solving [16]. As most rule pruning techniques (e.g. [9]), the filtering process is for reducing the error rates of the mappings acquired.

(2) Implementing the modules for relevance feedback: To focus on the investigation of the system's performance, the experiment assumed that each query was tagged with a target categor and relevant documents. As mentioned above, the tagging process may be replaced with the modules for relevance feedback (ref. Section 2.2). The modules may determine the target document categories and the relevant documents b interacting with the user or observing the user' behaviors in reading documents.

(3) Applying the framework to the design of Executive Information System (EIS): Textual information is an essential kind of information for the executives in decision -making. Providing personalized information services is a critical issue of developing EIS as well.

5. CONCLUSION

In this paper, we propose a technique to develop an IR system that can adapt itself to the user's queries and the document database. Through incremental learning, the way of mapping query terms to suitable document categories may be acquired so that more relevant information may be found in smaller search spaces. Thus both the quality and the efficiency of IR may be improved significantly. As users require more personalized information services, the framework may serve as a basis for developing adaptive information systems for the users.

ACKNOWLEDGMENT

This research was supported by the National Science Council of the Republic of China under the grants NSC 87-2213-E-216-009 and NSC 88-2213-E-216-003.

REFERENCE

[1] Allan J., 1996, "Incremental Relevance Feedback for Information Filtering," Proc. of *19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[2] Appleton E. L., 1992, "Smart Document Retrieval," *Datamation*, Vol. 38, No. 2.

[3] Armstrong R., Freitag D., Joachims T., and Mitchell T., 1995, "WebWatcher: A Learning Apprentice for the World Wide Web," Proc. o f *AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environment*, AAAI Press.

[4] Ba S., Lang K. R., Whinston A. B., 1997, "Enterprise Decision Support Using Intranet Technology," *Decision Support System*, 20, 99-134.

[5] Belew R. K., 1989, "Adaptive Information Retrieval: Using a Connectionist Representation to Retrieve and Learn about Documents," Proc. of *ACM SIGIR*.

[6] Bloedorn E. Mani I., and MacMillan T. R., 1996, "Machine Learning of User Profiles Representational Issues," Proc. of AAAI.

[7] Burke R. D., Hammond K. J., and Kulyukin V. J., 1997, "Question Answering from Frequentl -Asked Question Files: Experiences with the FAQ Finder System," technical report TR-97-05, Department of Computer Science, University of Chicago.

[8] Chen H., Chung Y. -M., Ramsey M., and Yang C. C., 1998, "An Intelligent Personal Spider (Agent) for Dynamic Internet/Intranet Searching," *Decision Support Systems*, 23, 41-58.

[9] Cohen W. W., 1995, "Fast Effective Rule Induction," Proc. of *the Twelfth International Machine Learning Conference (ML95)*.

[10] Daniels J. J. and Rissland E. L., 1995, " Case-Based Approach to Intelligent Information Retrieval," Proc. of the *18th ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, Washington, USA.

[11] Fedorowicz J., 1993, "A Technolog Infrastructure for Document-Based Decision Support Systems", in *Decision Support System: Putting Theory into Practice*, Sprague R. H. and Watson H. J. (eds.), Prentice-Hall Inc.

[12] Fitzpatrick L. and Dent M., 1997, "Automatic Feedback Using Past Queries: Social Searching?", Proc. of the 20th *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[13] Khan I. And Card H. C., 1998, "Adaptive Information Agents Using Competitive Learning," *Journal of Network and Computer Applications*, pp. 69-90.

[14] Lam W., Mukhopadhyay S., and Palakal M., 1996, "Detection of Shifts in User Interests for Personalized Information Filtering", Proc. of the *19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[15] Liu R.-L. and Soo V.-W., 1992, "Augmenting and Efficiently Utilizing Domain Theory in Explanation-Based Natural Language Acquisition," Proc. of *the 9th Intern ational Machine Learning Conference (ML92)*, Aberdeen, Scotland.

[16] Liu R.-L. and Soo V.-W, 1994, "A Corpus-Based Learning Technique for Building Self-Extensible Parser," Proc. of the *15th International Conference on Computational Linguistics*, Kyoto, Japan.

[17] McCune B. P., Tong R. M., Dean J. S., and Shapiro D. G., 1985, "RUBRIC: A System for Rule-Based Information Retrieval," *IEEE Transactions on Software Engineering*, Vol. SE-11, No. 9.

[18] Meadow C. T., 1992, "Text Information Retrieval Systems," Academic Press Inc.

[19] Mitchell T. M., Keller R. M., and Kedar-Cabelli S. T., 1986, "Explana tion-Based Generalization: A Unifying View," *Machine Learning*, 1:47-80.

[20] Rissland E. L. and Skalak D. B., 1991, "CABARET: Rule Interpretation in a Hybrid Architecture," *International Journal of Man-Machine Studies*, 34, 839-887, Academic Press.

[21] Tseng C. -M., Lang C.-Y., and Chien L. -F., 1994, "An Approach to Extending Record-Based Database Management Systems for Chinese Full-text Indexing and Searching," Proc. of *the 5th International Conference on Information Retrieval*.

[22] Tu H.-C., 1998, "Interactive Web IR Focalization Model, Effectiveness Measures, and Experiments, *Ph.D. dissertation*, National Taiwan University.

[23] Weiss S. A., 1996, "Towards an Adaptive Framework for Information Retrieval," Proc. of *AAAI Spring Symposium on Information Retrieval*.