Taxonomy-based Adaptive Web Search Method

Said Mirza Pahlevi and Hiroyuki Kitagawa

University of Tsukuba
Tsukuba, Ibaraki 305-8573, Japan
{mirza, kitagawa}@kde.is.tsukuba.ac.jp

Abstract

Current crawler-based search engines usually return a long list of search results containing a lot of noise documents. By indexing collected documents on topic path in taxonomy, taxonomy-based search engines can improve the search result qualities. However, the searches are limited to the locally compiled databases. In this paper, we propose an adaptive web search method to improve the search result qualities enabling the users to search in many databases existing in the web space. The method has a characteristic that combines the taxonomy-based search engines and a machine learning technique. More specifically, we construct a rule-based classifier using pre-classified documents provided by a taxonomy-based search engine based on a selected context category on its taxonomy, and then use it to modify the user query. The resulting modified query will be sent to the crawler-based search engines and the returned results will be presented to the user. We evaluate the effectiveness of our method by showing that the returned results from the modified query almost contain documents that will be categorized into the selected context category.

1. Introduction

It is well-known that the crawler-based search engines have a good coverage of the web. It is because they crawl web pages automatically with only a bit of human intervention. However, since the engines store each crawled page/document as a bunch of keywords – no topic organization of the pages is made, they typically only support keyword-based search, which makes the returned results generally contain a lot of noise documents. On the other hand, taxonomy (directory)-based search engines like *Open Directory Project/ODP* (http://dmoz.org) have good precision of the search results. It is because they manage web pages by using taxonomy. Pages with similar topics are (logically) stored/grouped in the same category, which makes the user easily find the information sought. However, since the classification is done manually, the engines can only cover a small fraction of the web.

There are many attempts to classify the web content automatically into a taxonomy [2] [4]. The main goal of those systems is to deal with the exponential growth in the volume of the online text databases. They start with a small sample of corpus that is classified by hand to build a hierarchical classifier. At run time, each web page retrieved will be classified automatically by the classifier into an appropriate category. However, this approach has the following disadvantages.

- It is very hard to build a good and large hierarchical classifier that can deal with a wide variety of topics like ODP.
- Most of the classifiers cannot deal with modification of category hierarchies, for instance deletion and addition of category nodes and their associated documents, which is important in the dynamic web environment.

In this paper, we propose an adaptive web search method combining existing taxonomy-based search engines and a machine learning technique. More specifically, we modify the user query by using a rule-based classifier constructed from a document collection provided by a taxonomy-based search engine and send the modified query to the crawler-based search engines. The query is modified such that the results returned by the crawler-based search engines will almost contain documents that will be categorized into a selected category on the taxonomy of the taxonomy-based search engine. The modification process is adaptive – the classifier constructed is different depending on both the selected category and the query given by the user. The most important characteristic of our method is that it can be applied to the query-based search of any databases compiled independently from the given taxonomy as long as they support Boolean search.

The paper is organized as follows. Section 2 elaborates our motivation. Section 3 describes the proposed method. Section 4 presents the experiments and results. Section 5

reviews related work. Section 6 gives our conclusions and suggests future work.

2. Motivation

One of the main reasons why the crawler-based search engines generally return results containing a lot of noise documents is the ambiguity of terms used in the user query. This ambiguity originates from the use of very short queries, which is usual in the web environment [6]. By allowing the user to select an appropriate category in a taxonomy besides providing search terms, the taxonomy-based search engines can solve the search term ambiguity problem. It is because the searches can now be restricted to documents in the specified category. However, the searches can only be done against the manually compiled local databases, and cannot be expected to give many useful results.

The idea here is instead of providing the search results from the taxonomy-based search engine directly to the user, it is better to get some useful keywords first by "learning" the search results, then use the keywords to modify the user query and finally send the modified query to the crawler-based search engines.

3. Proposed method

Our challenge is to make the best of the existing taxonomy-based search engines to facilitate web searches. One way to do this is to extract some useful information from the taxonomy-based search engines and use the information to enrich the user query. Another challenge is to preserve the enriched user query so that it is still in a Boolean form. By doing this, we can get many useful information from many search engines available in the web space since they typically support Boolean query.

In this paper, we assume that a crawler-based search engine and a taxonomy-based search engine are available and they can process queries in a Boolean form. We further assume that the taxonomy-based search engine allows search based on all categories existing in the taxonomy and provides additional information about the category of each matched document. (Most of major taxonomy-based search engines support this.)

3.1. Query formulation and context category selection

The query formulation process is the same as the search process that is usually used in taxonomy-based search engines. To find relevant information, first the user navigates the taxonomy provided by a taxonomy-based search engine. After the user has found a category related to the topic sought, he/she then constructs a keyword-based query [1] that will be sent to the engine. We call the category selected by the user as a *context category*. The user may choose the context category after browsing some documents under the category or seeing the category description.

3.2. Separation of relevant and non-relevant documents

The system sends the given query condition to the taxonomy-based search engine without specifying a specific category. After the system receives the query results from the engine, it separates the relevant and non-relevant documents based on the context category as follows.

- Documents that are classified into the context category (and subcategories under the context category) [2] are considered to be relevant to the user query. This conforms to the method used by the taxonomy-based search engines to catch the user intent.
- Otherwise they are considered to be non-relevant to the query.

Based on this procedure, a relevant document is a document that matches the user query condition and is classified into the context category.

3.3. Query modification and execution

After the relevant and non-relevant documents have been found, next the system modifies the user query and sends it to the crawler-based search engines. In this work, we use a rule-based classifier to modify a Boolean query. First, we construct a classifier for two new categories: *relevant* and *non-relevant categories*. The relevant category is a category for the relevant documents while the non-relevant category is for the non-relevant documents. The classifier is constructed by setting the relevant and non-relevant documents as positive and negative examples, respectively. The resulting classifier is a set of rules in the form of $T \rightarrow c$ where T is a conjunction of terms and c is *Relevant* or *Non-relevant* category. Construction of such rule-based classifiers has been intensively studied in the area of machine learning [3] [1]. In our experiment explained in Section 4, we use RIPPER [3] for constructing the classifier.

Next we modify the initial user query q with the rule set for the relevant category as follows.

1. Let $R = \{r_1, \ldots, r_n\}$ be the rule set for the relevant category, where $r_i = T_i \rightarrow Relevant$. Note that T_i is a conjunction of terms.

[1] Most of search engines treat the given terms as a term conjunction, and thus we assume this is a Boolean query.

[2] In the remaining part, we refer to the context category and its descendant subcategories just as the "context category".

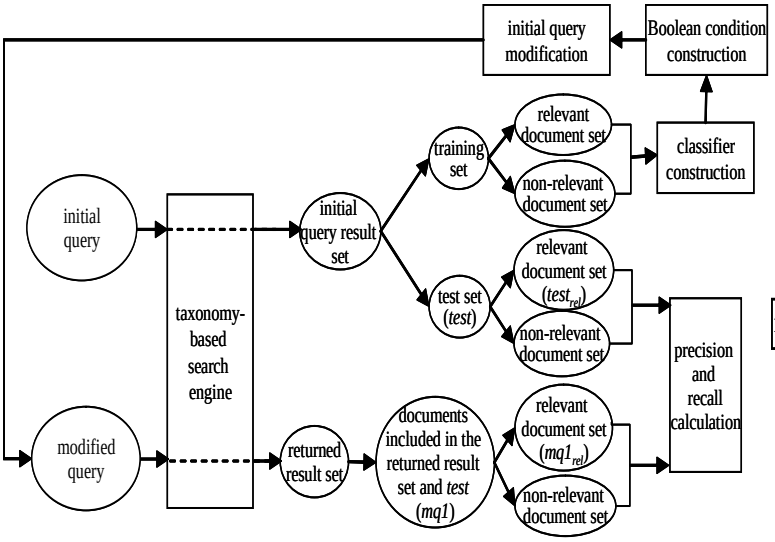

Figure 1. Experiment 1.

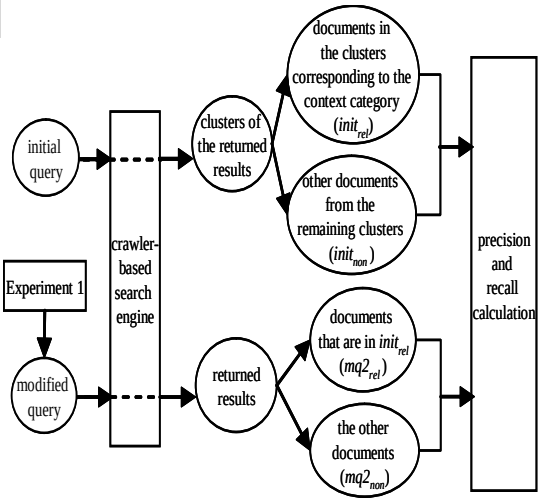

Figure 2. Experiment 2.

2. Let q' be T_1 OR . . . OR T_n.

3. Finally, we modify q by AND-ing it with q', that is, q AND $q'$is the query condition of the modified query.

It is important to note that, our method *dynamically* constructs the classifier used to modify the user query. It is because the classifier is constructed based on documents that match the query where the documents should be different from query to query. This characteristic is important because each query usually has different meaning even for the same context category.

The classifier is used to tell whether a document that matches the initial query condition will be classified to the context category. In other words, we use the classifier to eliminate the term ambiguity problem that may occur when conducting a search in the crawler-based search engines. Hence, by ”sending” the classifier with the initial query to the crawler-based search engine (i.e., transforming it to a Boolean condition and modifying the initial query), it seems that the returned results from the search engine will almost contain documents that are related to the user intent.

4. Experiments and results

To evaluate the effectiveness of the proposed method, we compare the precision and recall of the modified query with those of the initial query. However, in order to calculate the precision and recall, we have to know the ”true” answer of each query with respect to the selected context category. One way to do this is to check whether each document returned by the crawler-based search engine is relevant or not to the query. However, this approach requires too much effort since the returned result size is usually very large.

4.1. Experiment 1

To make relevance judgment easy, we simulate the crawler-based search engine with a taxonomy-based one. This can be done by having the search carried out against documents in all categories of the taxonomy-based search engine. That is, the search is not done against a particular category as usual. The ”true” answer of a query from the simulated crawler-based search engine is the subset of documents that match the query condition and that are classified into the query’s context category. (Note that the returned documents are associated with their categories.)

The detail of Experiment 1 is shown in Figure 1. The taxonomy-based search engine has two functions: it is used to catch the user intent by the proposed method and used as the simulated crawler-based search engine. The flow of the experiment is as follows. First, we define an initial query and select an appropriate context category for the query from the taxonomy. After the query is submitted to the taxonomy-based search engine without specifying a specific category, we get the initial query result set. The result set is then divided into training set and test set ($test$), which in turn are divided into relevant and non-relevant document sets based on the selected context category. The relevant and non-relevant documents in the training set are used to construct the classifier, which in turn is used to modify the initial query. The resulting modified query is then sent to the simulated crawler-based search engine (in this case the taxonomy-based search engine itself) and the precision and recall of the returned results are calculated based on $test$.

Table 1. Queries and their context categories.

Query	Broad context category		Narrow context category		Basic meaning
	Notation	Context category	Notation	Context category	
q1:ATM	c1.1	/Computers/	c1.1'	/Computers/Data_Communications/	ATM networks
	c1.2	/Business/	c1.2'	/Business/Financial_Services/Banking_Services/Automatic_Teller_Machines/	ATM banks
q2:salsa	c2.1	/Arts/	c2.1'	/Arts/Performing_Arts/Dance/Latin/Salsa/	salsa art
	c2.2	/Shopping/	c2.2'	/Shopping/Food/Condiments/	salsa sauce
q3:apple	c3.1	/Computers/	c3.1'	/Computers/Systems/Apple/	Apple computer
	c3.2	/Home/Cooking/	c3.2'	/Home/Cooking/Fruits_and_Vegetables/	apple cooking
q4:oil AND product	c4.1	/Business/Industries/	c4.1'	/Business/Industries/Energy/Oil_and_Gas/	oil product in industries
	c4.2	/Shopping/	c4.2'	/Shopping/Health/Beauty/	oil product for health

The precision and recall of the modified query is calculated as follows. Let $mq1$ be a set of documents included in the result set of the modified query and in $test$. Let $mq1_{rel}$ be the set of relevant documents in $mq1$, namely, documents that meet the initial query condition and are classified into the context category. Similarly, let $test_{rel}$ be the set of relevant documents in $test$. In this experiment, $test_{rel}$ is the "true" answer of the initial query because it is a relevant document set and it is not involved in constructing classifier of the proposed method. We calculate the precision and recall of the modified query using the following equation.

$$precision = \frac{|mq1_{rel}|}{|mq1|}, recall = \frac{|mq1_{rel}|}{|test_{rel}|}$$

Note that the recall of the initial query is always 1, while the precision is $|test_{rel}|/|test|$.

We conduct the evaluation process with 3-fold cross validation and use Open Directory Project/ODP as the taxonomy-based search engine. The search results from ODP are lists of site entries, each of which consists of a title, description, address and category name. In the experiments, each site entry is regarded as a document.

There are 4 queries with 16 different meanings used in the experiment. We select two context categories for each query such that the meaning of the query at each context category is different. For example, the meaning of query "apple" at context category "/Computers/" is completely different from the same query at different context category "/Home/ Cooking/". We also do evaluation when the context categories are shifted to narrower concepts (i.e., shifting them to subcategories). By shifting the context category of each query to a narrower one, we can shift the meaning of the query to a more specific topic. It is because the meaning of the query depends on its context category. For example, the meaning of query "apple" at context category "/Computers/" is *to find pages related to Apple computers such as companies, hardware, software etc.*, while the meaning of the query at context category "/Computers/System/Apple/" is *to find pages specially related to Apple computer system* [3].

Table 1 shows the queries and their basic meanings at the selected context categories. Figures 3 through 6 show the experiment results. Prefix I and M denote the initial query and modified query, respectively. The recall of the initial query is omitted, because it is always 1. As can be seen, at broad context categories our proposed method can significantly increase the precision of the initial query with a low decrease in recall. At narrow context categories, the increase of the precision is slightly change but the decrease of recall is somewhat larger.

The performance at the broad context categories is better than that at narrow context categories because the more specific the context categories are, the harder for the classifiers to recognize documents belonging to the categories. This is also true in a real world situation. For example, it is generally easier to decide whether a document belongs to category "/Health/" or "/Art/" rather than to decide whether a document belongs to category "/Health/Medicine/" or "/Health/Pharmacy/" assuming the document has been classified into the "/Health/" category.

[3] We can derive the meaning of the query from category description provided by the taxonomy-based search engine.

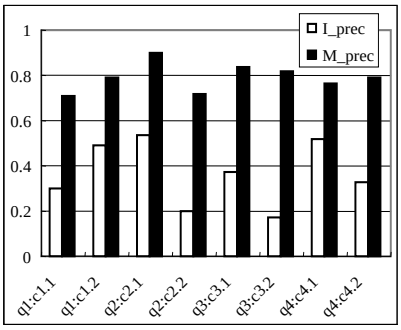

Figure 3. Precision at broad context categories.

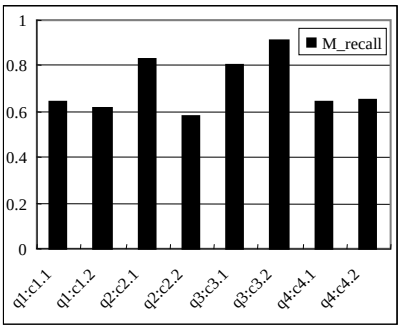

Figure 4. Recall at broad context categories.

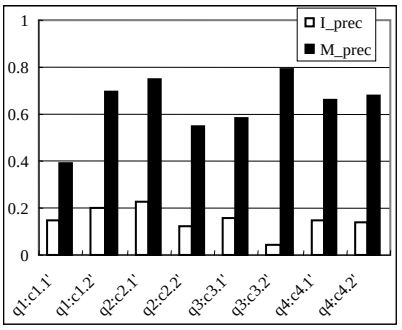

Figure 5. Precision at narrow context categories.

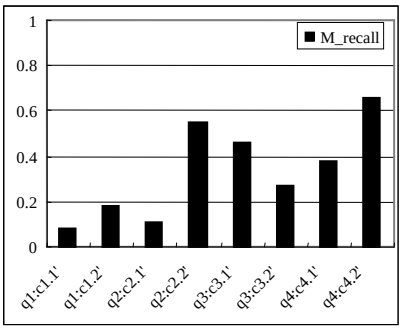

Figure 6. Recall at narrow context categories.

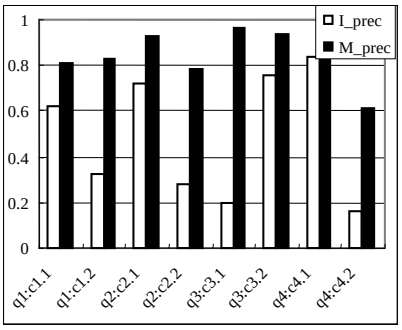

Figure 7. Precision at Northernlight.

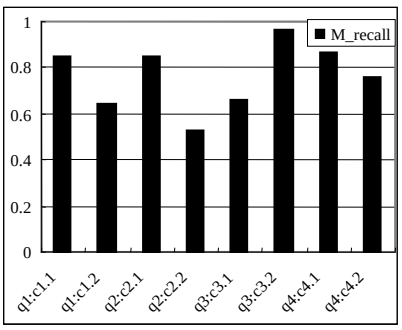

Figure 8. Recall at Northernlight.

4.2. Experiment 2

We use a real crawler-based search engine in this experiment. To make relevance judgment easy, the (initial/modified) query is sent to a crawler-based search engine that clusters its search results. Then, we identify clusters whose documents will most likely be classified into the selected context category and treat documents in the clusters as the "true" answer for the query. We denote the selected and remaining clusters as *relevant* and *non-relevant clusters*, respectively. The relevant clusters are picked by inspecting the cluster names and several documents existing in the clusters. However, since it is difficult to match the clusters with a context category having a very specific topic, we only take broad categories as the context categories in the experiment. (Note that queries used in the experiment are the same as those of Experiment 1.)

Figure 2 shows the flow of Experiment 2. As shown in the figure, we reuse the modified queries derived from Experiment 1. Let $init_{rel}(init_{non})$ be a set of documents from relevant clusters (non-relevant clusters) and let $mq2_{rel}(mq2_{non})$ be a set of documents from the modified query results that are in $init_{rel}(init_{non})$. Note that, since we modified the initial query by AND-ing it with a Boolean condition, the returned results of the modified query should be a subset of those of the initial query. The recall and precision of the modified query is shown below.

$$precision = \frac{|mq2_{rel}|}{|mq2_{rel}| + |mq2_{non}|}, recall = \frac{|mq2_{rel}|}{|init_{rel}|},$$

Similar to Experiment 1, the recall of the initial query is always 1 and the precision is shown below.

$$precision = \frac{|init_{rel}|}{|init_{rel}| + |init_{non}|},$$

We use Northern Light (http://www.northernlight.com/) as the crawler-based search engine since it clusters its search results and supports Boolean query. Note that it is only for evaluation purpose, that is, our method can be applied to any crawler-based search engines supporting Boolean search.

As can be seen, the precision and recall of the modified queries are slightly larger than those of Experiment 1 (Figures 3 and 4) but in general they show the same trend.

4.3. Summary

From the experiments, it is clear that our method can retrieve documents based on a selected context category with high precision regardless of the context category position in the taxonomy, while the recall change depending on the position. This indicates that our method is suitable for the searches in the web space that require high precision.

5. Related work

The most closely related to our work is the Inquirus 2 [5]. They proposed an automated method for learning query modifications to locate pages within specified categories using web search engines. Our work is different from theirs in that we use the existing taxonomy to catch the user intent. By using the existing taxonomy, we can make best of it as a useful information source. On the other hand, they use flat categories that they have to construct and provide to users. Another difference is that their query modification is static, while ours changes dynamically depending on the query provided by the user.

Another related work is WebSifter II, a semantic taxonomy-based personalizable meta-search engine agent system [7]. The system captures the user intent by having users create personalized taxonomies expressing their queries via the proposed Weighted Semantic-Taxonomy Tree. The taxonomies are then transformed into Boolean queries processed by existing search engines. Although the system uses taxonomy, it does not employ classifiers. In addition, the system needs a new taxonomy for each query intent.

6. Conclusions and future work

We have proposed an adaptive web search method combining existing taxonomy-based search engines and a machine learning technique. The method is able to adaptively modify the user query based on a selected context category from taxonomy provided by the taxonomy-based search engines. The modification is done so that the results returned by the crawler-based search engines may contain many documents that will be categorized into the selected context category. This enables the user to freely shift the broadness of his/her intent topics just by selecting an appropriate category from the taxonomy.

Our method is adaptive in that the classifier constructed to modify the query is different depending on both the selected category and the query given by the user. In other words, our method dynamically constructs a small classifier corresponding to a small part of the taxonomy that is related to the current user query. Moreover, since the modified query is still in a Boolean form, our method can be applied to any databases supporting Boolean search. This characteristic is very important because we can get more relevant information not only from the crawler-based search engines but also from many legacy databases or hidden web (i.e. the web that cannot be indexed by crawlers) that actually occupy a large portion of the web.

We are going to do detailed evaluation of the performance (response time) of the proposed method. The response time is the time between issuing the initial query and modifying the query. In the experiment, most of the time is occupied by the time to collect the positive and negative examples from the taxonomy-based search engine. (Note that the time needed to build the classifier in only a few seconds.) It is because the taxonomy-based search engine does not present the resulting documents in one page at a time, so that we have to fetch them by sending several HTTP requests. We can decrease the time by sending the requests concurrently using multi-thread process. Furthermore, since the documents are only used to build the classifier (not presented to the user), we can stop sending the requests after we get enough examples to build the classifier.

7. Acknowledgement

This research has been supported in part by the Grant-in-Aid for Scientific Research from Japan Agency for the Promotion of Science.

References

[1] C. Apte et al. Towards language independent automated learning of text categorization models. *In SIGIR Conference on Research and Development in Information Retrieval*, pages 23–30, 1994.

[2] S. Chakrabarti et al. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. *The VLDB journal*, 7(3):163–178, 1998.

[3] W. W. Cohen. Fast effective rule induction. *International Conference on Machine Learning*, pages 115–123, 1995.

[4] S. Dumais and H. Chen. Hierarchical classification of web content. *In SIGIR Conference on Research and Development in Information Retrieval*, pages 256–263, 2000.

[5] E. J. Glover et al. Improving category specific web search by learning query modifications. *Symposium on Applications and the Internet (SAINT)*, pages 23–31, 2001.

[6] B. J. Jansen et al. Real life information retrieval: A study of user queries on the web. *SIGIR Forum*, 32(1):5–18, 1998.

[7] L. Kerschberg et al. A semantic taxonomy-based personalizable meta-search agent. *Proceedings of the second International Conference on Web Information Systems Engineering (WISE)*, pages 53–62, 2001.