Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

A dynamic knowledge base based search engine

WANG Hui-jin (王会进)[†1], HU Hua (胡 华)[†2], LI Qing (李 清)[1]

([1]*Department of Computer Science, Jinan University, Guangzhou 510632, China*)

([2]*Department of Computer Engineering, Zhejiang Gongshang University, Hangzhou 310035, China*)

[†]E-mail: twang21cn@21cn.com; huhua@mail.hzic.edu.cn

Received Mar. 25, 2005; revision accepted May 20, 2005

Abstract: Search engines have greatly helped us to find the desired information from the Internet. Most search engines use keywords matching technique. This paper discusses a Dynamic Knowledge Base based Search Engine (DKBSE), which can expand the user's query using the keywords' concept or meaning. To do this, the DKBSE needs to construct and maintain the knowledge base dynamically via the system's searching results and the user's feedback information. The DKBSE expands the user's initial query using the knowledge base, and returns the searched information after the expanded query.

Key words: Dynamic knowledge base, Query expansion, Information retrieval, Search engine
doi:10.1631/jzus.2005.A0683 **Document code:** A **CLC number:** TP391

INTRODUCTION

Search engines are widely used on the Internet to help users to mine the Web efficiently. Most search engines adopt the whole length retrieval (WLR) technique to collect and index Web pages in a database before they accept a query of keywords from the user. The database data will be used to retrieve Web pages that match the query. WLR technique has two disadvantages: firstly, it just compares each character of keywords, not the internal concept of the keywords; secondly, the retrieval results just rely on the keywords from the user and the system has no further interactive action with the user (Li *et al*., 2000). An Information Retrieval (IR) system is suggested for overcoming the disadvantages by expanding the user's query based on some kinds of knowledge base (van Rijsbergen, 1979); there are two methods to implement the query expansion:

1. Expanding the query with a Static Knowledge Base

In this method, an IR system first constructs a static knowledge base in advance and then expands the query via the static knowledge base. Li *et al*.(2000)'s Chinese Intelligence Information Retrieval System (CIIRS) constructs a concept network represented by graph to store all information of the searched fields. CIIRS implements the query expansion using the concept network.

2. Expanding the query with real-time results

In this method, an IR system searches for the documents first with the initial query, then extracts characteristic concepts from the documents to expand the initial query. INQUERY (Broglio *et al*., 1994) of Massachusetts University presents strategies as: Local Feedback, Relevance Feedback, Local Context Analysis, etc. for the typical implementation of this method.

While static knowledge base constructed by experts always contains the concepts and the relations of the concepts, its construction relies on the specified expert's knowledge and the query expansion based on such knowledge base is reasonable and coincides with human perception. But on the other hand, the organizing and management of the knowledge base will need entail much labor. Experts should update the knowledge base from time to time in case the extended query misses some updated information. Because the update could require reconstruction of the knowledge base, this method is not suitable for the

field that changes rapidly such as the computer field.

The latter method overcomes the former method's disadvantage in requiring manual update of the knowledge base. But, the expansion based on the searched documents may not coincide with human perception, which may cause misunderstanding. In addition, the seemingly inappropriate workload here would adversely affect the response time of the IR system.

In this paper, we present an IR system called Dynamic Knowledge Base based Search Engine (DKBSE) which are the syntheses of two previous methods. Given below are some of its characteristics:

(1) Construction of the knowledge base automatically. Different from the static knowledge base, the dynamic knowledge base is constructed automatically by the search results.

(2) Updating the knowledge base dynamically. The expert does not need to update the knowledge base periodically. The system will update knowledge base dynamically using the search result and the user's feedback.

(3) Improving the search result of the query. The knowledge base has close relation with the content of the documents, as the concepts and the relations among the concepts in the knowledge base are extracted from the documents. This close relation will greatly benefit the search accuracy of the IR system.

(4) Much should be more. The system updates the knowledge base using the users' feedback information, to some extent, can help accumulate human perception.

CONSTITUTION OF THE DYNAMIC KNOWLEDGE BASE

In DKBSE, the knowledge base describes the concepts and the relations of the concepts. The relations of the concepts include semantic resemblance relation, semantic generality relation and semantic specificity relation. Semantic resemblance relation describes the relation of the different terms which express the same concept, such as "computer" and "PC". Semantic specificity relation describes the specificity relation of the concepts, such as "operation system" and "Microsoft windows". Semantic generality relation describes the general relation of the concepts, which is the reverse relation of the semantic specificity relation.

The knowledge base is the kernel of the system, its rationality affects the retrieval accuracy greatly. In DKBSE, we construct the knowledge base using the retrieval results and the feedback information, which includes the following:

(1) Extracting concepts;

(2) Judging the relations of the concepts;

(3) Optimizing knowledge base.

Extracting concept

Concept is an abstract entity expressed by concept describing elements. Extracting concept from normal information is equivalent to extracting concept describing elements from normal information. In DKBSE, we extract describing elements from the search results. If we choose all of the search results to construct the knowledge base, the capability of the knowledge base will increase rapidly and will greatly and positively affect the system performance. So to construct the knowledge base we only select some characteristic terms representing document content from the search results.

The steps to extract characteristic terms are:

Step 1: Selecting relative terms from the search result set with the weight of the terms in documents. To select the relative terms, we define the weight v_i of the term d_i in the document *doc* as follows:

$$v_i = \frac{TF(i)}{\sum_{d_j \in doc} TF(j)}$$

here, $TF(i)$ is the frequency of the term d_i in the document doc;

Step 2: By doing Step 1 for every document in the search result set, we can obtain the relative term set A.

Step 3: Counting the associate degree of the term c_i with the query Q. The associate degree refers to the degree of the term c_i associating with all terms of the query Q in a document (Xu, 1997). If c refers to a term, the query Q is made up of the keywords $w_1, w_2, \ldots, w_n$. The following equation measures the associate degree of c with w_i:

$$co\_degree(c, w_i)$$

$$=\lg(co(c,w_i)+1)\min(1.0,\lg(N/N_c)/5.0)$$

here, $co(c, w_i)$ is the number of the documents in which the terms c and w_i exist simultaneity in the searched result set. N is the number of the documents of the searched result set. N_c is the number of the document in which the term c exists in the searched result set.

The associate degree of the term c with the query Q is:

$$g(c,Q)=\sum_{w_i \text{ in } Q} co\_degree(c,w_i)/k$$

Sept 4: Selecting the terms which have larger associate degree as the characteristic terms.

Judging the relations of the concepts

In the above paragraph, we stated that the relations of the concepts including semantic generality relation, semantic specificity relation and semantic resemblance relation. The first two are the embodiers of the subsumption relation. The following subsection will introduce the judging method of the relations.

1. Finding subsumption relation

According to the special syntax regulation or the context comparability of the term (Sanderso and Croft, 1999), there are two approaches to finding the subsumption relation. In DKBSE, we adopt the term subsumption method, which is based on the DF (Document Frequency) hypothesis (Sanderso and Croft, 1999). In DF method, the generality and specificity of the terms are determined by their document frequency, the more documents a term occurred in, the more general it is assumed to be. For two terms, x and y, Dx is the document set in which x exists; Dy is the documents set in which y exists and Dxy is the documents set in which x and y both exist. There are some possible situations and the relevant rules according to the relation among the sets:

(1) If $P(x|y)<1$ and $P(y|x)<1$, then no relation exists between x and y;

(2) If $P(x|y)=1$ and $P(y|x)<1$, then the set Dx subsumes the set Dy, x subsumes y;

(3) If $P(x|y)<1$ and $P(y|x)=1$, then the set Dy subsumes the set Dx, y subsumes x.

In there, $P(x|y)=||Dxy||/||Dy||$, $P(y|x)=||Dxy||/||Dx||$, $||Dx||$ refers to the count of the term x elements in the set Dx (Wu, 1999).

In actual use, we should make some modifications of the above rules in order to find more subsumption relation. The revised rules for finding subsumption relation are:

(1) $P(x|y)<\alpha$ and $P(y|x)<\alpha$, then no relation exists between x and y;

(2) $P(x|y)>\beta$ and $P(y|x)<\alpha$, then the set Dx subsumes the set Dy, x subsumes y;

(3) $P(x|y)<\alpha$ and $P(y|x)>\beta$, then the set Dy subsumes the set Dx, y subsumes x.

Here, α is the upper limit, β is the lower limit. These two constants value may be adjusted during the system's implementation.

According to the above rules, the DKBSE system selects the characteristic terms from the search result set, finds the subsumption relations among them and then stores the terms and the relevant relations into the knowledge base.

2. Finding semantic resemblance relation

In DKBSE, if the generality words set of a term x resembles the one of another term y at the same time, the specificity words set of the term x resembles the one of the term y, then we suppose the term x resembles the term y.

Optimizing knowledge base

When a user makes a search, he will probably get the extended query from the system. After that, he can choose some terms from the extended query according to his knowledge, and feedback the chosen information to the system. From this situation, the system can optimize the knowledge base according to the feedback information. The optimization includes increasing the weight of the relations related to the chosen terms and decreasing the weight of the relations that have not been used for a long time. When the weight of a relation is less than a threshold, it will be deleted from the knowledge base. After having been used by some people for some time, the structure of the knowledge base will be more reasonable and coincide with human perception.

IMPLEMENTATION OF THE SYSTEM

Structure of DKBSE

DKBSE has a structure of three tiers (Fig.1). The

three tiers are User service, Application service and Data service (Luo and Huang, 2001). The User service mainly implements the interaction between the user and the IR system; Application service includes three modules named query expander, searcher and result processor. The query expander is used to expand the initial query, the searcher is used to search documents related to the query in the documents set, and the result processor extracts new knowledge to update the knowledge base from the search result set. Data service includes two parts: knowledge base and documents set.

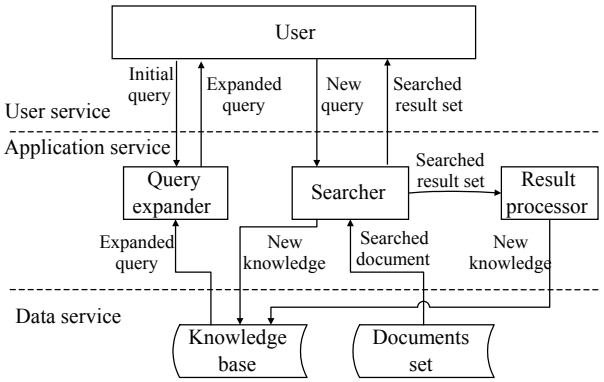

Fig.1 Structure of DKBSE

Building indexes for the content of the documents

In DKBSE, to build the indexes of the document content, we should do:

(1) Analyze the document contents. A document is composed of words; we should extract words from the document first. For English documents, this work is very easy; for Far East character set document, it will be relatively difficult.

(2) To get the etymon of the word. A word has many variations. If all variations of a word are stored in database, the capability of the database will increase rapidly, which also will affect the retrieval system's response time. In DKBSE, we get every word's etymon by Stemmer method to reduce the demand on the database's capability.

(3) To wipe off the stop words. To a certain extent, the frequency of a word in a document shows the significance of the word in the document. But some general words are exceptional, which have high frequency, but have no means. In DKBSE, we design a "stopper word" table, which includes most general words. The system omits all "stopper word" from the documents, and just reserves the words related to the content of the document.

(4) To build indexes. After the system deals with the documents by the above three steps, it can establish an inverse index table of words and documents, which stores the information such as the relation of word and document, the frequency of word in document, etc.

Query expansion

In DKBSE, we expand the query via the concepts and the concepts' relations of the knowledge base; there are two tables named subsumerelation and connectrelation in the knowledge base to store the information of subsumption relation and resemblance relation among concepts separately.

In DKBSE, the work flow to process the initial query by Query expander is:

(1) When Query expander has received the query (usually including one or more keywords) submitted by Customer, it deals with the keyword w_i in the query separately. Firstly, the Query expander gets all records related to the keyword w_i in the subsuming relation table and connecting relation table. Secondly, the Query expander will find out generality words (semantic generality expansion), specificity words (semantic specificity expansion), resemblance words (semantic resemblance expansion) from the records and stores them in three sets: W_1, W_2, W_3, and then feeds back the sets to the User.

(2) User selects the words set relevant to the initial query from W_i and submits the chosen words set S_i, which is the subset of W_i (i=1,2,3), to the system.

(3) Before the extended query is submitted to the Searcher, the words in the extended query will be set to different selected weight according to the subset that the words come from (in DKBSE, we suppose that the words in the same chosen set have the same weight). Here, we define the SW (select weight) of the chosen subset of every set is:

$$SW(S_i)=\frac{\|S_i\|}{\sum_{j=1}^{3}\|S_i\|}C_i \quad (i=1,2,3)$$

in which, S_i is the subset chosen from the set W_i; $\|S_i\|$

refers to the counts of the elements in the subset S_i; W_1, W_2, W_3 refer to the set of generality words, the set of specificity words, the set of resemble words; C_i is a constant, and refers to the weight of the set W_i.

(4) In DKBSE, we suppose the words in the same chosen subset have the same weight. This assumption means that if w_k is in S_i, then the *TW* (term weight) is equal to the *SW* value, that is $TW(w_k)=SW(S_i)$.

Execute query

In DKBSE, the Searcher is used to search for relevant documents. It scans the documents set, finds the documents related to the query. The flowing is the following working steps:

Step 1: The Searcher receives the new extended query from the User, at the same time, every keyword w_i in the query has a corresponding term weight $TW(w_i)$;

Step 2: Searches for the documents in which w_i occurs from the inverse index table. Here, we define the *DW* (document weight) of the searched document D_{ij} is:

$$DW(D_{ij})=\frac{F_j(w_i)}{\sum F_j}TW(w_i)$$

here, D_{ij} is the *j*th document in which the key word w_i occurs; $F_j(w_i)$ is the frequency of the key word w_i in the document D_{ij}; $\sum F_j$ is the summation of the frequency of all words in the document D_{ij};

Step 3: Adds the searched for binary group (D_{ij}, $DW(D_j)$) into the search result set. If the document D_{ij} exists in the search result set, we update the weight of the document D_{ij} to: $DW_{new}=DW_{old}+DW(D_j)$;

Step 4: Does Steps 2 and 3 repeatedly for the document in which the keyword w_i exists;

Step 5: Does Steps 2, 3 and 4 repeatedly for every keyword in the new extended query from the User;

Step 6: Rearranges the documents of the search result set sorted according to the weight of the documents.

After the above six steps, user can retrieve a document set sorted by the document weight. The system will also update the knowledge dynamically.

EXPERIMENTAL RESULTS

The DKBSE is developed on the Oracle 8i Database programming in Java language. We choose 50 English documents of IR as the experimental test document set. After every query, DKBSE updates the knowledge base with searched results and user's feedback information. We may call one user's research as one time's training. Table 1 shows the variance of the accuracy according to the variance of the number of the training.

Table 1 The relation of accuracy of retrieval and number of training

Number of training	Accuracy of retrieval
0	33.3%
10	33.3%
20	46.7%
30	60.0%
40	66.7%

Table 1 shows that the larger the number of the training is, the higher is the accuracy of the retrieval. The knowledge base of the DKBSE is constructed using the searched results step by step, the larger the number of the training is, the more reasonable is the constitution of the knowledge base, and then the system can expand query more reasonably and accurately.

The experimental result showed that when DKBSE constructs the knowledge base gradually, our IR system can help the user to expand his query reasonably. This improves the interaction between system and user and finally improves the accuracy of the system in some extent.

This early stage of the experiment only includes a small document set. In the next experiment stage, we will choose the TREC5 as the experimental document set to evaluate the system's retrieval accuracy.

CONCLUSION

DKBSE extracts characteristic terms from the searching result set. It then finds out the relations among the terms by DF hypothesis and constructs the dynamic knowledge base with terms and relations.

DKBSE can improve the accuracy of the IR system by expanding the initial query using the dynamic knowledge base. Further work will include: improving the approach of extracting characteristic terms from documents in a more efficient way, improving the searching approach to have higher quality search result.

References

Broglio, J., Callan, J.P., Croft, W.B., 1994. INQUERY System Overview. http://ciir.cs.umass.edu/.

Li, L., Wang, N., Zhang, J., 2000a. A probe on Chinese concept_based retrieval. *Computer Engineering and Application*, **6**:1-3.

Luo, S.D., Huang, Y., 2001. A search engine based on a concept_network which can teach itself automatically. *Computer Engineering*, **9**:89-92.

Sanderso, M., Croft, W.B., 1999. Deriving Concept Hierarchies from Text. http://ciir.cs.umass.edu/.

Van Rijsbergen, C.J., 1979. Information Retrieval (Second Edition). http://ciir.cs.umass.edu/.

Wu, Y.F., 1999. Automatic Concept Hierarchies Development: A Revised Subsumption Approach. State University of New York, Albany.

Xu, J.X., 1997. Solving the Word Mismatch Problem Through Automatic Text Analysis. http://ciir.cs.umass.edu/.