WUJNS
Wuhan University Journal of Natural Sciences

Vol.11 No.5 2006 1167-1171

Article ID:1007-1202(2006)05-1167-05

A Novel Web Query Automatic Expansion Based on Rough Set

□ YI Gaoxiang[1,2], HU Heping[1†], LU Zhengding[1], LI Ruixuan[1]

1. College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China

2. China Academy of Safety Science and Technology, Beijing 100029, China

Abstract: One of important reasons caused low precision was presented, which was due to inaccurate express of the query. So a new method of automatic query expansion based on tolerance rough was put forward. In the algorithm, the uncertain connection between query terms and retrial documents was described as term tolerance class. The upper approximation set of query sentence was considered as query expansion. The new additional terms were also given weight numbers. The results of experiment on collection of Google 5 000 Web pages showed that the approach was effective on query expansion and high search precision was gained.

Key words: Web query; query expansion; rough set

CLC number: T 391

Received date: 2006-02-23

Foundation item: Supported by the National Natural Science Foundation of China(60403027)

Biography: YI Gaoxiang(1972-), male, Ph. D. candidate, Lecturer of China Academy of Safety Science and Technology, research direction: Data Mining, Web information extraction, E-mail:hunanliling_1972@163.com

† To whom correspondence should be addressed. E-mail:hphu@andin.net

0 Introduction

Searching the web is not always so successful as users expect. Most of the retrieved sets of documents in a web search meet the search criteria but do not satisfy the user's needs. One crucial reason is that users generally lack of specificity in the formulation of the queries. Some causes of this are that most of the times, the user does not know the vocabulary of the topic, or query terms do not come to user's mind at the query moment.

One possible solution to this problem is the process known as query expansion or query reformulation. After the query process is performed, new terms are added to and/or removed from the query in order to improve the results, i.e., to discard uninteresting retrieved documents or to retrieve interesting documents that were not retrieved by the query. A good review of the topic in the information retrieval can be found[1,2].

The purpose of this work is to provide a system with a query expansion based on rough set technologies. The rough set model was proposed by Pawlak in the early 1980s. It is an extension of standard set theory that supports approximations in decision making. The main goal of rough set analysis is to synthesize approximation of concepts from the acquired data[3]. It has been successfully applied in various tasks, such as feature selection/extraction, rule synthesis and classification[4]. The query expansion takes into account the degree of importance of terms in the representation of documents. And the method is good to comprehend and easy for applications.

1 Traditional Expansion Query in VSM

In this section the vector space model is xplained, which is

the basis of our work. Essential parts of this model are the representation of documents and queries, a scheme for weighting terms, and an appropriate metric for calculating the similarity between a query and a document[1,2].

1.1 Representation of Documents and Queries

The task of document retrieval is to retrieve documents that are relevant to a given query from a fixed set of documents, i.e. a document database. A common way to deal with documents, as well as queries, is to represent them using a set of index terms (simply called terms). In the following, $t_i (1 \leq i \leq m)$ and $d_j (1 \leq j \leq n)$ represent a term and a document in the collection respectively, where m is the number of terms and n is the number of documents.

$$d_j = (w_{1j}, w_{2j}, \cdots, w_{ij}, \cdots, w_{mj}) \quad (1)$$

where w_{ij} is the weight of a term t_i in a document d_j. A query is likewise represented as:

$$q_k = (w_{1k}, w_{2k}, \cdots, w_{ik}, \cdots, w_{mk}) \quad (2)$$

where w_{ik} is the weight of a term t_i in a query q_k, $(1 \leq k \leq K)$, K is the number of query.

1.2 Weighting Schemes

In our work, we used the most distributed weighting scheme: the standard normalized formula is defined as follows:

$$w_{ij} = f_{ij} \times \log(n / p_i) \quad (3)$$

where f_{ij} is the frequency of the term t_i occuring in document d_j, and p_i is number of documents in corpus in which term t_i occurs.

1.3 Similarity Measurements

The result of the retrieval is represented as a list of documents ranked according to their similarity to the query. The selection of a similarity function is a further central problem having decisive effects on the performance of an IR system. A common similarity function in text-based IR systems is the cosine metric $S(d_j, q)$.

$$S(d_j, q) = \frac{d_j \cdot q^T}{\|d_j\| \cdot \|q\|} \quad (4)$$

where T indicates the transpose, d_j is a document and q is a query vector, $\| * \|$ is the Euclidean norm of a vector.

1.4 Reformulation of the Query

Query expansion tries to reformulate the initial user query in a way that the query moves nearer to the relevant documents. This includes methods that modify weights of the query terms and add new terms. These new terms may be found from generally available online thesauri or from feedback document.

Someone suggested a method for relevance feedback that uses average vectors (centroids) for each set of relevant and irrelevant documents[2]. The new query is formed as a weighted sum of the initial query and the centroid vectors. Formally the relevance feedback is defined as follows: Let q be the initial query and n_1 be the amount of relevant and n_2 be the amount of irrelevant documents. Then the new query q is formed by[5]:

$$q' = q + \frac{1}{n_1} \sum_{\text{relevant}} \frac{D_i}{|D_i|} - \frac{1}{n_2} \sum_{\text{non-relevant}} \frac{D_i}{|D_i|} \quad (5)$$

The relevance feedback of the previous section supplies good results but it has a crucial disadvantage. It needs user feedback. However this is very hard to get in real IR-Systems because only few users are willing to do the job of assessing documents. One idea to simulate this explicit user feedback is to rely on the performance of the IR system and to postulate: The best n_1 of the ranked document list are relevant. These are used as positive feedback for the relevance feedback method. It may be possible to postulate: The last n_2 documents are irrelevant and use them as a negative feedback. Experimental results have shown that positive feedback, i.e. marking only relevant documents, is generally better than using positive and negative feedback. For simply and practically, we only consider the n top return pages for positive feedback

2 Rough Set and Tolerance Rough Set

The classical rough set theory is based on equivalence relation that divides the universe of objects into disjoint classes[6-8].

Consider a non-empty set of object U called the universe. The central point of rough set theory is the notion of set approximation: any set in U can be approximated by its lower and upper approximation. Originally, in order to define lower and upper approximation we need to introduce an indiscernibility relation: $R \subseteq U \times U$ (where R can be any equivalence relation, which is reflexive, symmetric, transitive.). For two objects $x, y \in U$, if xRy then we say that x and y are indiscernible from each other. The indiscernibility relation R induces a complete partition of universe U into equivalent classes $[x]_R, x \in U$. We define lower and upper approximation of set X, with regards to an approximation space denoted by $A = (U, R)$, respectively as:

$$L_R(X) = \{x \in U \mid \mu(x, X) = 1\} \quad (6)$$

$$U_R(X) = \{x \in U \mid \mu(x, X) > 0\} \quad (7)$$

where $\mu_X(x, X) = \frac{|[x]_R \cap X|}{|[x]_R|}$.

Intuitively, X lower approximation contains objects that certainly belong to our concept while upper approximation contains objects that may belong to our concept.

Practically, for some applications, the requirement for equivalent relation has showed to be too strict. And it must be extended. For example, let us consider a collection of scientific documents and keywords describing those documents. It is clear that each document can have several keywords and a keyword can be associated with many documents. Thus, in the universe of documents, keywords can form overlapping classes. By relaxing the relation R to a tolerance relation, where transitivity property is not required, a generalized tolerance space is introduced below[9,10].

Let $I: U \to P(U)$ to denote a tolerance relation, if and only if $x \in I(x)$ for $x \in U$ and $y \in I(x) \Leftrightarrow x \in I(y)$ for any $x, y \in U$, where $P(U)$ are set of all subsets of U. Thus the relation $xIy \Leftrightarrow y \in I(x)$ is a tolerance relation (i.e. reflexive, symmetric) and $I(x)$ is a tolerance class of x. To define the tolerance rough membership function $\mu_{I,v}$ as: $x \in U$, $X \subseteq U$,

$$\mu_{I,v}(x, X) = v(I(x), X) = \frac{|I(x) \cap X|}{|I(x)|} \quad (8)$$

The tolerance rough set for any $X \subseteq U$ are then defined as

$$L_R(X) = \{x \in U \mid v(I(x), X) = 1\} \quad (9)$$

$$U_R(X) = \{x \in U \mid v(I(x), X) > 0\} \quad (10)$$

3 Query Expansion Based TRSM

Tolerance Rough Set Model (TRSM) was developed as basis to model documents and terms in information retrieval, text mining, etc. With its ability to deal with vagueness and fuzziness, tolerance rough set seems to be a promising tool to model relations between terms and documents. In many information retrieval problems, especially in query expansion, defining the relation (i.e. similarity or distance) between document-document, term-term or term-document is essential. The application of TRSM in query expansion was proposed as a way to enrich term representation with the hope of improvement information retrieval.

3.1 Tolerance Space of Term

In TRSM, the tolerance space is defined over a universe of all index terms $U = T = \{t_1, t_2, \cdots, t_m\}$

The idea of query expansion is to capture conceptually related index terms into classes. For this purpose, the tolerance relation I is determined as the co-occurrence of index terms in all documents from set D. The choice of co-occurrence of index terms to define tolerance relation is motivated by its meaningful interpretation of the semantic relation in context of IR and its relatively simple and efficient computation.

3.2 Tolerance Class of Term

Let $f_D(t_i, t_j)$ denotes the number of documents in D in which both terms t_i and t_j occurs. The uncertainty function I with regards to threshold θ is defined as $I\theta(t_i) = \{t_j \mid f_D(t_i, t_j) \geq \theta\} \cup \{t_i\}$

Clearly, the above function satisfies conditions of being reflexive: $t_i \in I\theta(t_j)$ and symmetric: $t_j \in I\theta(t_i) \Leftrightarrow t_i \in I\theta(t_j)$ for any $t_i, t_j \in T$. Thus, $I\theta(t_i)$ is the tolerance class of index term t_i.

In context of Information Retrieval, a tolerance class represents a concept that is characterized by terms it contains. By varying the threshold θ (e.g. relatively to the size of document collection), one can control the degree of relatedness of words in tolerance classes.

The membership function μ for t_i and q, $q = \{t_1, t_2, \cdots, t_k\}$, $q \subseteq T$ is then defined as:

$$\mu(t_i, q) = v(I\theta(t_i), q) = \frac{|I\theta(t_i) \cap q|}{|I\theta(t_i)|} \quad (11)$$

Finally, the lower and upper approximations of any subset $q \subseteq T$ can be determined with the obtained tolerance relation respectively as:

$$L_R(q) = \{t_i \in T \mid v(I\theta(t_i), q) = 1\} \quad (12)$$

$$U_R(q) = \{t_i \in T \mid v(I\theta(t_i), q) > 0\} \quad (13)$$

For example, the upper tolerance of query "fuzzy and rough set" is shown if Table 1.

Table 1 Tolerance classes generated from snippets of query "fuzzy and rough set" with co-occurrence threshold θ set to 7

Term	Tolerance class
Fuzzy	Fuzzy, knowledge, applications, rough
Rough	Rough, computing, data, granular, fuzzy
Computing	Computing, data, rough
Discovery	Discovery, knowledge
Data	Data, computing, granular, rough
Knowledge	Knowledge, discovery, applications, fuzzy
Applications	Applications, knowledge, fuzzy
Granular	Granular, data, rough

3.3 Expansion the Query on Tolerance Class of Term

With TRSM, the aim is to enrich representation of query by taking into consideration not only terms actually occurring query but also other related terms with similar meanings. A "richer" representation of query can be acquired by representing query as set of tolerance classes of terms it contains. This is achieved by simply representing query with its upper approximation. Let $q = \{t_1, t_2, \cdots, t_k\}$ be a query on D and $t_1, t_2, \cdots, t_k \in T$ are index terms of D.

$$U_R(q) = \{t_i \in T \mid v(I\theta(t_i), q) > 0\} \quad (14)$$

3.4 Weighting Scheme for Query Expansion

To assign weight values for query's vector, generally the same 1 is used for query term. But the term in the query is not of the same importance. Clearly, the terms in the lower approximation of query are more important than the other. Therefore, the terms in the lower approximation of query may be assigned 1 while the terms in query's upper approximation but not in the lower approximation may be assigned the tolerance membership. The extended weighting scheme is defined as below and should be normalized.

$$w_i = \begin{cases} 1, & \text{if } t_i \in L_R(q) \\ v(I\theta(t_1), q), & \text{if } t_i \in U_R(q) - L_R(q) \\ 0, & \text{other} \end{cases} \quad (15)$$

3.5 Tolerance Class Generation Algorithm

Input F—document-term frequency matrix, θ—co-occurrence threshold

Output T—term tolerance binary matrix defining tolerance classes of term

Step 1 Calculate a binary occurrence matrix C based on document-term frequency matrix TF as follows: $C = [c_{ij}]_{N \times M}$ where

$$c_{ij} = \begin{cases} 1, & \text{if } f_{ij} > 0 \\ 0, & \text{other} \end{cases} \quad (16)$$

Each column in C is a bit vector representing term occurrence pattern in a document 1 bit is set if term occurs in a document.

Step 2 Construct term co-occurrence matrix $B = [b_{x,y}]_{M \times M}$ as follows:

$$b_{x,y} = P(C[x] \text{ AND } C[y]) \quad (17)$$

where $C[x], C[y]$ are pairs of term x, y bit vectors in the C matrix, AND is a binary AND between bit vectors and P return cardinality number of bits set —of a bit vector, $b_{x,y}$ is the co-occurrence frequency of term x and y.

Step 3 Given a co-occurrence threshold θ, a term tolerance binary matrix $T = [t_{x,y}]_{M \times M}$ can be easily constructed by filtering out cells with values smaller than threshold θ:

$$t_{x,y} = \begin{cases} 1, & \text{if } b_{x,y} > \theta \\ 0, & \text{other} \end{cases} \quad (18)$$

Each row in the resulting matrix forms a bit vector defining a tolerance class for given term, $t_{x,y}$ is set if term x and y are in tolerance relation.

4 Experiment on the Web

To evaluate the proposal algorithm, we implemented query expansion based on tolerance rough set to compare Google results. We formed randomly on various topics a set of 50 queries and then submit the queries to the Google. The top 100 results are collected as test corpus and the top 30 pages also as feedback document for query expansion. We use relevancy to evaluate the expansion effectiveness on return documents. Since on the Web, most users only review the first 10 or 20 results returned by the search engine, and the actual number of relevant pages is unknown, we only consider the relevant pages in the top 40 results in our experiment.

Let us define the evaluation standard of relevancy R.

$$R = \sum_{i=1}^{n} (n - i + 1) \times W_i \quad (19)$$

where i denotes the ith page in the result page-list, n represents the top n pages chosen from page-list, and w_i is the weight of page i, manually chosen one of 1.0, 0.5, 0.1 or 0, based on the relevancy of query topic. Page relevancy is independently judged by five persons who are generally professors or Ph. D. candidates. The experiment results are resumed in Table 2.

From Table 2, we see that top results obtained from query expansion based on tolerance rough set are more likely relevant than common query. Some query expan-

Table 2 Comparison of the relevancy values for query without expansion and expansion

Query	R (40 top pages)	
	Google	TRS query expansion
Data mining	428.5	512.8
Rough set	337.1	568.5
Clustering search results	318.2	375.7
Web rank	326.5	394.2
Fuzzy set	457.3	488.4
Fuzzy and rough set	311.6	366.3

sion gain significant improvement such as "Rough set" because that its expansion of "Rough Set Computing, data, Granular" make it look like a special conception.

5 Conclusion

In this paper, we consider term co-occurrence in documents in order to form groups of correlated terms. We express the context of the query in its upper approximation set and use it to expand the query in a context-sensitive manner. This statistical approach is useful when no knowledge about the terms is available. In the algorithm, the uncertain connection between query terms and retrial documents was describe as term tolerance class. The upper approximation set of query sentence was viewed as query expansion. The new additional terms were also given weight numbers. The results of experiment on collection of Google 5 000 Web pages showed that the approach was effective on query expansion and high search precision was gained.

References

[1] Akrivas G, Wallace M, Stamou G, *et al*. Context - Sensitive Query Expansion Based on Fuzzy Clustering of Index Terms [C] // *Proceedings of 5th International Conference Flexible Query Answering Systems*. Copenhagen, Denmark, Oct. 27-29, 2002: 1-11.

[2] Stefan K. Improving Document Transformation Techniques with Collaborative Learned Term-Based Concepts [J]. *LNCS* 2956, 2004: 281-305.

[3] Liu Qing. *Rough Set and Rough Reasoning* [M]. Beijing: Science Press, 2003.

[4] Pal S K, Talwar V, Mitra P. Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions [J]. *IEEE Transactions on Neural Networks*, 2002, **5** (13): 1163-1177.

[5] Harman D. Towards Interactive Query Expansion [C] // *Pro of the Eleventh Annual International ACMSIGIR*, Grenoble, June 24-26, 1988: 321-331.

[6] Jouni J. Approximations and Rough Sets Based on Tolerances [J]. *Rough Sets and Current Trends in Computing*, 2000, **1**: 182-189.

[7] Yao Y Y, Song K, Saxton L V. Granular Computing for the Organization and Retrieval of Scientific XML Documents [C]// *Proceedings of the Sixth International Conference on Computer Science and Informatics*, Durham, NC, USA, March 8-14, 2002: 377-381.

[8] Yao Y Y, Yao J T. Granular Computing as a Basis for Consistent Classification Problems [C]// *Proceedings of PAKDD '02 Workshop on Toward the Foundation of Data Mining*, Taibei, China, May 6-8, 2002: 101-106.

[9] Tu Baoho, Ngoc Binhnguyen. Nonhierarchical Document Clustering Based on a Tolerance Rough Set Model [J]. *International Journal of Intelligent Systems*, 2002, **17** (2): 199-212.

[10] Ngo Chilang. *A Tolerance Rough Set Approach to Clustering Web Search Results* [D]. Warsaw: Warsaw University, 2003.