Workshop on Artificial Intelligence

An Intelligent Approach to Mining the Desired and Related Websites

Yo-Ping Huang* and Yen-Chun Lee
Dept. of Computer Science and Engineering
Tatung University
Taipei, Taiwan 10451 R.O.C.
e-mail: yphuang@cse.ttu.edu.tw
Tel: (02)25925252 ext 2473
Fax: (02) 25941371
* corresponding author*

Abstract-In the paper, we propose an intelligent meta-search engine that bases on user's query keyword(s) to find the extended keywords and more related websites. Instead of listing a lot of candidate websites waiting for users to search, the extended related keywords and more related documents appeared in the explorer window are derived from the proposed method. In case a user is not satisfied with the search results, he/she has the options of picking the advised related keywords generated by our system without knowing any background knowledge of the submitted query. User's interaction of the search results gives a positive feedback to our system for further improvement of the designed model. Most of current search portals do not analyze the relationship between the retrieved websites, but our system can create the websites' groups easily. We will present the designed model and illustrate how the proposed MetaSearcher system works in finding the desired websites.

Keywords: Grey relational method, fuzzy inference model, search engine

1. Introduction

According to a local survey made by a .com company that the use of search engine to find the interesting information is still the most important activity in the Internet. The survey outcome shows that 29.5% of Internet activities is related to information search. Most portals aim to provide a powerful search engine in response to users' queries. The British survey shows that more than 28% users are not satisfactory with the search results of current search portals. The most popular search engine, Yahoo, powered by Google internally, indexed more than 2 billion web pages [17]. With the huge amount of web pages, it is very important to develop the fastest and smart searching way for users.

In the Internet, most search engines use variations of the Boolean model [12] to do fast ranking. Although they can retrieve many possible websites or documents in a short time, users may face a difficulty in choosing which website to visit first. For example, when a user queries the word "neural" in Google, it returns 1,690,000 search results with unknown order. Every user gets the same order of search results when they submit the same query. Some search engines further exploit vector model for their full-text searching. The vector model makes the partial matching between a document and a query possible. So index terms must be determined in advance. But it always comes up with the problem that most users seldom input more than two terms for their queries. The situation make the vector model dedicate few help in searching process. Some search mechanisms, such as Mondou [3], find the related terms by using association rules [1-2]. It usually generates a satisfactory result, but the mining process might take us much time as the number of terms increasing. To overcome the weakness, we propose the MetaSearcher system with fast computing mechanisms that could refine more related index terms efficiently and rank closely related websites in a desired order.

Our intelligent model first exploits the possible websites for further analysis. Based on the metadata or full text from the possible websites we can count the term frequencies of interesting keywords. By applying the fuzzy inference model, we get the extended keywords of query and derive the weight of each keyword to support our grey relational model. The selected keywords provide users other choices to search the interesting documents. Each keyword has a relationship to any other candidate keywords. We define the value of relationship by the algorithm in our model. Those values will be tuned by the user to proceed a more precise search work. The analysis about how to obtain the extended keywords and more related websites for users is shown in the following sections.

2. Related Work

Many information retrieval models have been proposed since the Internet comes to the world. Most of them work well in certain purposes. We illustrate the popular models in section 2.1. Besides, we introduce different models built in our MetaSearcher system in section 2.2 and section 2.3.

2.1 Traditional models

Most search engines use variations of the Boolean or vector model to do ranking. In the Boolean model, the term frequencies of index terms are all binary, i.e., $t_{n,j} \in \{0,1\}$. When using Boolean model, the user often meets huge amount of retrieved information. In the vector model, the non-binary weights can be assigned for index terms of both queries and documents [13]. But there exists a weakness in the vector model. The rank of websites related to certain query has been decided before the query is submitted. Although the pre-calculated rank order accelerates the speed of retrieval, it hurts the flexibility of search mechanism. The following section will illustrate how they work.

Suppose we have m documents on the pool. Let k be the number of index terms in the system and k_i be the ith index term, $i \in [1,n]$. The set of all index terms is given by $K = \{k_1,...,k_n\}$. Define $t_{i,j}$ as a term frequency to represent index term k_i in a document d_j, $j \in [1,m]$. If an index term does not appear in the document, $t_{i,j}$ is set to zero. Thus, the term vector $\vec{d}_j = (t_{1,j}, t_{2,j},..., t_{n,j})$ for a document can be formed. The user's query can also be formed into term vector: $\vec{q} = (q_1, q_2,..., q_n)$.

To compute the degree of similarity between document d_j and user's query q can be obtained from cosine function as follows [12-13]:

$$sim(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{\| \vec{d}_j \| \times \| \vec{q} \|} = \frac{\sum_{i=1}^{n} t_{i,j} \times q_i}{\sqrt{\sum_{i=1}^{n} t_{i,j}^2} \times \sqrt{\sum_{i=1}^{n} q_i^2}}. \tag{1}$$

In general, index terms are mainly nouns. The term frequency may be modified as follows [12]:

$$f_{i,j} = \frac{t_{i,j}}{\max_l t_{l,j}}. \tag{2}$$

In Eq.(2) $\max_l t_{l,j}$ denotes the maximum term frequency among index terms. If an index term k_i does not appear in the document d_j, then $f_{i,j}$ is set to zero. Normally those terms that appear in many documents do not have much help in distinguishing a relevant document to a non-relevant one. Let N be the total number of documents in the system and n_i be the number of documents containing the index term, said *document frequency.* Further, let idf_i, *inverse document frequency* for k_i, be given by:

$$idf_i = \log \frac{N}{n_i}. \tag{3}$$

As a result, an index term may have a weight associated with it by:

$$w_{i,j} = f_{i,j} \times \log \frac{N}{n_i}, \tag{4}$$

or by a variation of this formula. Such term-weighting strategies are called tf-idf schemes. Several variations of the above expression for the weight $w_{i,j}$ are described in an interesting paper written by Salton and Buckley [9]. However, in general, the above expression should provide a good weighting scheme for many collections. For the query term weights, Salton and Buckley suggested:

$$w_{i,j} = (0.5 + \frac{0.5 f_{i,j}}{\max_l f_{1,j}}) \times \log \frac{N}{n_i}. \tag{5}$$

2.2 Grey relational model

Assuming S is a set of data sequence. $x_0 \in S$ is the reference sequence and $x_j \in S, j = 1,...,m$ are m sequences waiting to be compared with the reference one. $x_j(i)$ represents the ith term in the jth sequence. Let $\gamma(x_0, x_j)$ denote the grey relational degree between sequences x_0 and x_j. Then, the grey relational degree for the ith term is formulated as follows:

$$\gamma(x_o(i), x_j(i)) = \frac{\min_{j=1}^{m}\min_{p=1}^{n}\left|x_0(p) - x_j(p)\right| + \xi \max_{j=1}^{m}\max_{p=1}^{n}\left|x_0(p) - x_j(p)\right|}{\left|x_0(i) - x_j(i)\right| + \xi \max_{j=1}^{m}\max_{p=1}^{n}\left|x_0(p) - x_j(p)\right|}. \tag{6}$$

In Eq.(6), ξ is called the distinguishing coefficient and is normally set to 0.5. Note that $\min_{j}\min_{p}\left|x_0(p) - x_j(p)\right|$ defined in Eq.(6) is used to find the shortest distance among all possible terms and sequences. As a result, the grey relational degree between two sequences is defined as follows:

$$\gamma(x_0, x_j) = \frac{1}{n}\sum_{i=1}^{n}\gamma(x_0(i), x_j(i)). \tag{7}$$

2.3 Fuzzy inference model

Since the publication of Lotfi A. Zadeh's seminal work, "Fuzzy Sets," in 1965 [20], the variety and number of applications of fuzzy logic have been developed. Fuzzy logic deals with problems that have vagueness, uncertainty, or imprecision, and uses membership functions with values in [0,1]. The performance of fuzzy logic control and decision systems depends on the input and output membership functions, the fuzzy rules, and the fuzzy inference mechanism.

We introduce the concept of term frequency (TF) and document frequency (DF) in last section. They are suitable for the inputs in the fuzzy inference model. The term weight (W) is defined as the inferred output in the fuzzy rules. We will illustrate the process of fuzzification, inference of rules, and defuzzification in section 3.1.

3. Improving the Search Mechanism

Based on the query, our meta-search engine searches through four famous portals to collect the possible websites and put them in the ranking pool. In this paper we use the grey relational method to mining the related terms for users' queries and rank the candidate websites queued in the ranking pool.

The purpose to introduce the grey relational method is to help us quickly locate the related documents to the query from the portals. After finding the candidate documents, our next step is to find the related keywords to the query. For illustration purpose, we use our system to retrieve 50 links relative to the term “neural” from search engines, Google [7], AltaVista [8], AllTheWeb [20] and AskJeeves [21]. We found that all meta of each websites in any search engines can be divided into three parts: header, abstract and hyberlink. Fig. 1, Fig. 2, Fig. 3, and Fig. 4 show the first search results related to query “neural” from different portals.

Fig. 1. The retrieval result from Google.

Fig. 2. The retrieval result from AltaVista.

Fig. 3. The retrieval result from AllTheWeb.

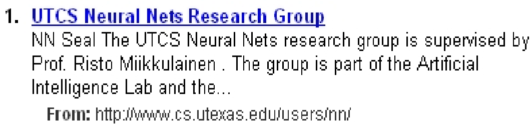

Fig. 4. The retrieval result from AskJeeves.

3.1 Generating the extended keywords and relative weights

The top 50 websites listed in each portal are selected into the ranking pool. We will illustrate how the grey relational method works in finding the related keywords. When we do information retrieval, the user’s query initializes the process first. Then our MetaSearcher system collects the related websites of this query from four famous portals and put them in the pool. Then we apply the fuzzy set theory to expanding the extended terms for users’ queries and defining the weight for each term. As a result, our system automatically arranges the related keywords in descending order for users to select. Users can rely on the recommended terms to search any interesting documents related to the original query term.

For meta-data, we pick the term frequency (TF) and document frequency (DF) as our input variables and term weight (W) as the to-be-inferred output in the fuzzy rules. The concept of TF and DF has been frequently proposed to do information retrieval before. They usually claim if the term frequency is higher or the document frequency is lower then the importance of the term grows. The concept works well in most indexing cases. Here, we emphasize a special point on the concept of DF. In most large indexing cases, the documents have few relations between each other. By this reason, the fewer the document frequency of one term, the higher distinction it will show. But in the case of indexing documents in any meta-search engines, the higher document frequency, the common features it will be found out. We can use the common features to extend the user's query. The proposed fuzzy inference rules are defined as follows:

Rule 1: If TF is H and DF is H, then W is H.
Rule 2: If TF is H and DF is M, then W is H.
Rule 3: If TF is H and DF is L, then W is L.
Rule 4: If TF is M and DF is H, then W is H.
Rule 5: If TF is M and DF is M, then W is M.
Rule 6: If TF is M and DF is L, then W is L.
Rule 7: If TF is L and DF is H, then W is M.
Rule 8: If TF is L and DF is M, then W is M.
Rule 9: If TF is L and DF is L, then W is L.

The membership functions for TF, DF, and TW are plotted in Fig. 7, Fig. 8, and Fig. 9, respectively. We use the Tsukamoto's defuzzification method for our strategy of defuzzification. In the case of singleton value, the monotonic membership functions are used in Eq.(8):

$$w_i = \sum_{i=1}^{n} x_i \cdot u(x_i) \Big/ \sum_{i=1}^{n} u(x_i). \tag{8}$$

3.2 Ranking the websites

The term frequencies for index terms in each document form a data sequence. For comparison, we need to find out the highest term frequency in all the website for each keyword. Those term frequencies form one data sequence. Based on the data sequence, the relative importance of the data frequency over the others can be calculated. By applying the Eq.(6) and Eq.(7), we can get the grey degree for each website.

In last section, we applied the fuzzy inference model to getting the extended keywords. We also tried to find out the extended keywords by grey relational method. Then the first thing we should do is to decide the suitable data sequence for comparison. The highest term frequency in each document is picked to form the data sequence. Some documents might have very high term frequencies for some terms, and it will make the data sequence of comparison having deviation. For this reason, it is not recommendable to generate the extended keywords by conventional grey relational model.

3.3 Adjusting the ranking order

Most search engines return fixed order of websites. It is impossible that the first returned search result is satisfactory to every user. So the user needs to review each of the website one by one until he/she finds the one he/she really wants. We can conclude here that different users might be interested in different kind of websites. Users can adjust the weight of each keyword shown in Fig. 5, and our system can respond to the user's feedback to combine the weights of index terms in Eq.(9) to get the answer closer to his/her desire. The system applies the grey relational method to ranking websites again. The adjusted weights of keywords will play an important role in the calculation of ranking. By integrating the calculated weights into the modified version of Eq.(8), we can decide the relative importance of each website as follows:

$$\gamma(x_o(i),x_j(i))=\frac{\min_{j=1}^{m}\min_{p=1}^{n}\left|x_0(p)-x_j(p)\right|+\xi\max_{j=1}^{m}\max_{p=1}^{n}\left|x_0(p)-x_j(p)\right|}{\left|x_0(i)-x_j(i)\times w_i\right|+\xi\max_{j=1}^{m}\max_{p=1}^{n}\left|x_0(p)-x_j(p)\right|}. \tag{9}$$

Note that in Eq.(9), a new weight w_i is attached to the original index term for re-calculating the grey relational degree.

3.4 Finding the relationship between websites or keywords

There exits a further application of the grey relational model. If we set the different comparison data sequence (for different website) in Eq.(6), we will get grey relational degrees between the chosen sequence and any other data sequences. The grey relational values (0~1) can be calculated in a short time. If there exists a strong relationship between two websites, the grey relational value will be close to 1. In another aspect, the candidate of data sequence can be replaced by the term frequency of certain keyword that shows in each website. Then the grey relational degree between two keywords can be calculated.

Some benefits can be revealed when we know the relation between two websites or two keywords. Situation 1: If the user thinks the website close to his/her desire, he/she can find closer websites in the high degree ones. Situation 2: If the user thinks the website is not relative, he/she can save time without reviewing the ones not relative to the website.

4. Experimental Results

The homepage of our MetaSearcher system is shown in Fig. 10. When the user submits the query "neural" to our system, our server will return the search results. The search results include the extended keywords and close websites relative to the query. All the information is shown in Fig. 11. According the theory mentioned in section 3.1, the extended keywords are inferred by fuzzy inference model. Based on the query keyword "neural," related keywords are also appeared for user to choose. The related keywords are arranged in descending order according to their respective grey relational degree to the submitted query. The generated grey relational degrees are meaningful to the users if they want to proceed with the advanced search. The order of keywords relative to query "neural" is "network", "system", "software" and "artificial." The fuzzy inference results of metadata of websites are shown in Table 1 and the inference results of full text of websites are shown in Table

2. Both tables contain the term frequency summation, document frequency, inferred degree, symbolic IDs, and original order for each keyword. The original order means the order of term frequency summation. We found that term “network” is more related to “neural” than the others. The inference results of full text will be more reasonable than metadata. But it costs more time to do indexing.

We implement the term weight adjustment function in our system and present it in web version. The web version of the extended keywords table is shown in Fig. 5, which is a part of search results shown in Fig. 11.

Fig. 5. The extended keywords and relative weights from our model.

In section 3.2, we introduced the concept of how to rank the websites. The descending order of degrees of websites is shown in Table 3. Of course, we also calculate the degrees of websites in full text as given in Table 4. We only show the records of the top 10 keywords’ here. The website with top rank usually has many keywords with high frequency.

The web search function is implemented in our system. We show the template of each website in our system in Fig. 6. The first part is the title text of the website. The second part represents the original order in different search engines. The third part is the metadata for this website. The fourth part shows the calculated values of the grey relational model and vector model. The last part shows hyperlinks of the homepage of the website, other links from this website and the related pages. We can find the related pages by our grey relational model mentioned in section 3.3.

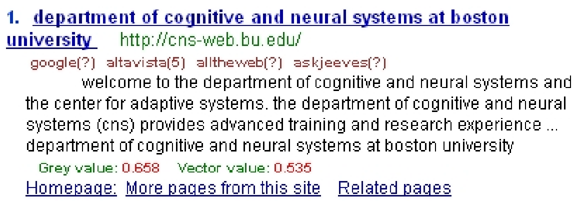

Fig. 6. The template of the website in the search result.

No matter whether the users satisfy our search result or not, the interactive functions in our system can feedback the users’ response for further adjusting the terms’ weights. As a result, our system will improve the search result. By applying the concept referred in section 3.3, we get the analyzed results in Table 5 and Table 6. Table 5 shows the grey relational matrix of the top 8 recommended websites related to the query “neural”. Table 6 shows the grey relational matrix of the top 7 extended keywords. Both tables present asymmetric matrices because of different deviations between different comparisons of data sequences with others. Each row in the table shows the relationship (strong or weak) between two websites or two keywords. With the grey relational degrees, we look out the relationship between websites or keywords easily.

5. Conclusions and Future Work

We presented an intelligent meta-search engine, MetaSearcher system, to find the closely related websites and extended keywords based on user's query. Our intelligent system first exploited the fuzzy inference model and grey relational method to locate the possible websites for further analysis. Based on the metadata from the possible websites we can count the term frequencies of interesting keywords. By applying the fuzzy inference model, our system generated the candidate keywords for extended functions. The selected keywords provided users advanced choices to search the interesting documents. Our system can find out the relationship between the retrieved keywords or websites by the grey relational method. Detailed analysis about how to obtain the more related websites for users was given. All above-mentioned functions have been implemented on the web version in our system.

In the future, many interesting functions can be further extended. First, there are many relationships existing in different types of objects. For example, the relationship between hyberlinks in the same website or in different websites, the relationship between a keyword and a website, the relationship between a website and a group of websites, and the relationship between graphic objects. In advanced analysis, not only the term frequency and document frequency, but new attributes, such as frequency of hyberlink, will be added to support our system's search mechanism.

Acknowledgment

This work is supported in part by the National Science Council, Taiwan under contract NSC90-2213-E-036-013 and in part by Tatung University under contract B90-1600-01.

References

[1] R. Srikant and R. Agrawal, "Mining generalized association rules," *Proc. of the 21st VLDB Conference*, Zurich, Switzerland, 1995.

[2] J. Han and Y. Fu, "Mining multiple-level association rules in large databases," *IEEE Trans. Knowledge and Data Engineering*, vol. 11, no. 5, Sept. 1999.

[3] H. Kawano and T. Hasegawa, "Mondou: interface with text data mining for web search engine," *Proc. IEEE 31st Annual Hawaii Int. Conf. on System Sciences*, 1998.

[4] C. L. Giles and K.D. Bollacker, "CiteSeer: an automatic citation indexing system," *Third ACM Conf. on Digital Libraries,* pp.89-98, 1998.

[5] J.-G. Kim and E.-S. Lee, "Intelligent information recommend system on the Internet," *Proc. IEEE Int. Workshops on Parallel Processing,* 1999.

[6] Z. Chen, X. Meng, R.H. Fowler, and B. Zhu, "FEATURES: real-time adaptive feature learning and document learning for web search," *Technical Report CS-00-23*, May 16, 2000.

[7] Yahoo: http://www.yahoo.com, 2002.

[8] AltaVista: http://www.altavista.com, 2002.
[9] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp.513-523, 1988.
[10] NetSizer: http://www.netsizer.com, 2002.
[11] NetCraft: http:// www.netcraft.com, 2002
[12] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*, Addison-Wesley, Taipei, Taiwan, 1999.
[13] G. Salton and M.E. Lesk, "Computer evaluation of indexing and text processing," *Journal of ACM*, vol. 15, no. 1, pp.8-36, Jan. 1968.
[14] W. Cai, "The extension set and incompatible problem," *Journal of Scientific Exploration*, vol. 1, pp.81-93, 1983.
[15] Y.-P. Huang, H.-J. Chen, and C.-P. Ouyang, "The integration of extension theory to design a new fuzzy inference model," *Int. Journal on Artificial Intelligence Tools*, vol. 9, no. 4, pp.473-492, Dec. 2000.
[16] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp.513-523, 1988.
[17] Google: http://www.google.com, 2002.
[18] G. Salton, *The SMART retrieval system – experiments in automatic document processing*, Prentice Hall Inc., Englewood Cliffs, NJ, 1971.
[19] J. L. Deng, "Introduction to grey system theory," *J. Grey System,* vol. 1, no. 1, pp.1-24, 1989.
[20] AllTheWeb: http://www.alltheweb.com, 2002.
[21] AskJeeves: http://www.ask.com, 2002.

Table 1. The statistics of keywords inferred from meta data of websites.

Keyword	ID	term frequency	document frequency	inferred weight	original order
neural	k1	97	43	1.0	1
network	k2	56	23	0.930	2
system	k3	12	7	0.781	3
software	k4	11	6	0.771	4
artificial	k5	8	8	0.771	5
research	k6	7	5	0.746	6
journal	k7	6	5	0.741	9
application	k8	7	4	0.739	8
society	k9	7	4	0.739	7
university	k10	4	4	0.723	12

Table 2. The statistics of keywords inferred from full text of websites.

Keyword	ID	term frequency	document frequency	inferred weight	original order
neural	K1	227	21	1.0	1
network	K2	153	12	0.939	2
research	K3	128	8	0.920	3
system	K4	84	8	0.884	4
application	K5	43	10	0.874	9
learning	K6	52	6	0.839	6
software	K7	28	8	0.835	19
center	K8	36	7	0.832	11
science	K9	43	6	0.829	8
model	k10	32	7	0.827	14

Table 3. Part of the ranked websites indexed from meta data.

	Degree	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10
L1	0.650	3	0	0	0	4	1	0	0	0	0
L2	0.645	6	6	0	0	0	0	0	3	1	0
L3	0.638	3	2	5	0	0	0	0	0	0	0
L4	0.613	3	3	0	0	0	0	0	0	0	0
L5	0.612	2	0	0	0	0	0	3	0	0	0
L6	0.611	1	0	0	1	2	0	0	0	0	0
L7	0.611	6	0	0	0	0	0	0	0	0	0
L8	0.605	3	1	2	0	1	0	2	0	0	0
L9	0.604	0	0	0	0	0	0	0	1	2	0
L10	0.604	1	0	0	1	1	1	0	0	0	0

Table 4. Part of the ranked websites indexed from full text.

	Degree	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
#1	0.981	123	87	40	57	83	15	39	34	9	30
#2	0.804	19	8	11	19	29	11	8	1	15	6
#3	0.735	13	13	0	0	5	0	2	1	1	0
#4	0.734	2	0	2	2	3	2	1	0	1	1
#5	0.731	10	7	0	2	0	1	1	1	0	0
#6	0.730	10	10	0	0	2	1	0	0	4	2
#7	0.728	3	1	0	1	1	0	0	0	1	2

#8	0.727	5	5	0	0	1	0	0	0	0	2
#9	0.727	9	7	0	0	4	0	0	0	0	0
#10	0.726	0	0	0	1	0	0	0	0	1	0

Table 5. The grey relational degrees between different websites indexed from full text.

	#1	#2	#3	#4	#5	#6	#7	#8
#1	1.0	0.793	0.761	0.755	0.756	0.756	0.752	0.751
#2	0.768	1.0	0.883	0.883	0.875	0.881	0.876	0.876
#3	0.743	0.888	1.0	0.96	0.973	0.974	0.964	0.973
#4	0.753	0.897	0.963	1.0	0.967	0.97	0.982	0.975
#5	0.742	0.884	0.974	0.965	1.0	0.98	0.975	0.984
#6	0.742	0.889	0.975	0.969	0.98	1.0	0.981	0.982
#7	0.748	0.89	0.967	0.982	0.976	0.982	1.0	0.987
#8	0.744	0.889	0.975	0.975	0.984	0.982	0.986	1.0

Table 6. The grey relational degrees between different keywords indexed from full text.

	neural	network	cognitive	system	research	brain	learning
neural	1.0	0.966	0.937	0.943	0.947	0.936	0.937
network	0.955	1.0	0.941	0.943	0.954	0.94	0.946
cognitive	0.916	0.939	1.0	0.979	0.962	0.984	0.992
system	0.91	0.93	0.975	1.0	0.961	0.97	0.971
research	0.927	0.952	0.962	0.967	1.0	0.958	0.96
brain	0.93	0.948	0.987	0.978	0.964	1.0	0.984
learning	0.916	0.945	0.992	0.976	0.961	0.981	1.0

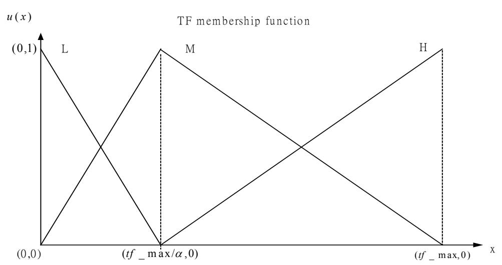

Fig. 7. Membership functions for term frequency.

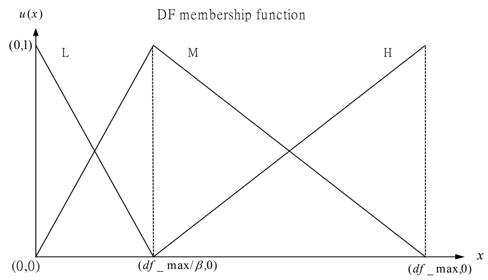

Fig. 8. Membership functions for document frequency.

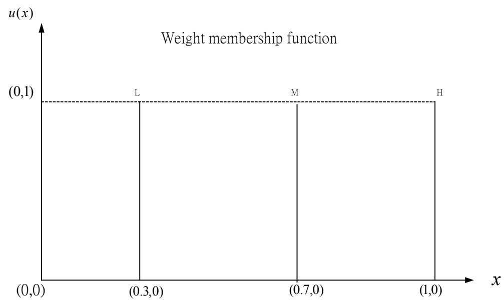

Fig. 9. Membership functions for term weight.

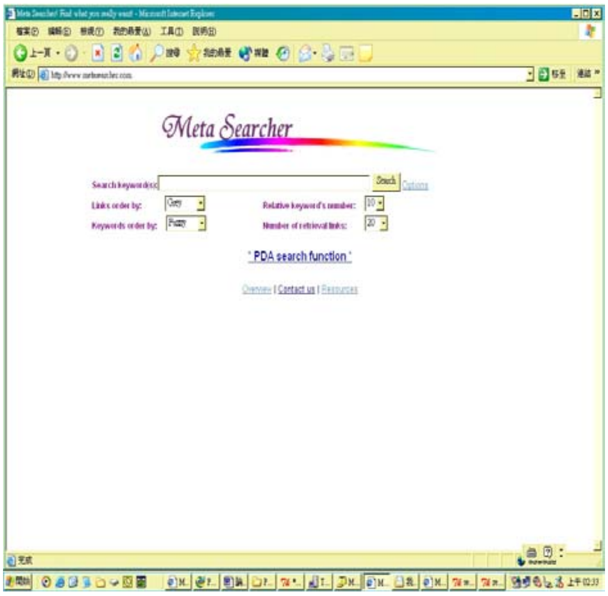

Fig. 10. The homepage of the MetaSearcher system.

Fig. 11. The searched result from our MetaSearcher system.