Providing Prioritized Search Result with Tag Coupling-based Boolean Query Matching

WonKyun Joo[1], MinWoo Park[1], KiSeok Choi[1], Yong Kim[2] and Young-Kuk Kim[3]

[1] Department of R&D System development, KISTI, 245 Daehak-ro, Yuseong-gu, Daejeon, South Korea
[2] Department of Library & Information Science, Chonbuk National University, 567 Baekje-daero, Deokjin-gu, Jeonju-si, Jeollabuk-do, South Korea,
[3] Department of Computer Science & Engineering, Chungnam National University, 99 Daehak-ro, Yuseong-gu, Daejeon, South Korea
{joo, pminwoo, choi}@kisti.re.kr, yk9118@jbnu.ac.kr, ykim@cnu.ac.kr
**Corresponding author: MinWoo Park*

Abstract

Since information systems for providing only Boolean search do not provide prioritized search result, users have to carry out time-consuming checking for lots of results one by one. The method proposed in this study is to provide search result prioritized by using coupling information between tags instead of index weight information in Boolean search. Since document queries are used instead of general user queries, key tags to be used as queries in a relevant document are extracted. A variety of groups of Boolean queries depending on tag couplings are created in the process of creating queries, and prioritization is processed by means of differentiation information between relevant query groups, and tag significance information in the process of matching. The proposed method was applied to the process of finding related trend analysis information for the emerging technology information consisting of 1500 technologies to prove the usability of the proposed method.

Keywords: *Boolean search, tag-based matching, search by means of non-weight information, query decomposition and extension, extraction of expert tag*

1. Introduction

Documents are identified and stored by means of key words or index words in typical information search. User's diversified search request has resulted in various search technologies, which have developed into exemplary Boolean search and similarity search, e.g., vector space models [1]. As a result of Internet information search, for example, Google, information search is generalized, which uses ancillary information, for example, similarity search and link relationship on the basis of weight information [2, 3]. However, many information services [4] are still focused on typical document information search, and provide Boolean search that does not use the index weight information. In general, users in the field of document information use Boolean queries and ask the very accurate number of searches.

Although Boolean search is very useful for specific purpose, typical Boolean search has some disadvantages [5]. First, it is hard to control the size of a search result set because words-based matching is used. There may be sometimes too many results, and sometimes no result. Second, since the search result is not ranked, all of the searched documents are of the same priority. Third, it is impossible to give a weight to words connected to a document or a query. Words included in the document or query have the same priority. Fourth, it is possible to show non-intuitive search result. For example, since it is necessary to include at least one word in the OR query statement, the same significance is shown in the result of both of including one word and including the entire words. Similarly, in the AND query, both of the document missed out in the search result because only one word among the words consisting of the entire queries is not included, and the document which does not include any query word are regarded, the same, uselessly.

Unlike Boolean search, similarity search is better than Boolean search in all respects. The only disadvantage of similarity search is that it cannot process structural queries used in Boolean search. The extended Boolean model [5, 7] was proposed to address the disadvantage of the aforementioned two methods of search. The extended Boolean model can embrace both of the Boolean model and the similarity search model by using weight information and controlling specific elements. However, since both of similarity search and the extended Boolean model are based on weight, weight information should be constructed in advance, or replacement information to substitute the weight information should be provided. Another study provides a method of extending queries on the basis of thesauri [8, 9], and it is another restriction to use huge external resources of thesauri.

The method proposed in this paper is to use simple information of tags which appeared together with Web 2.0 instead of using the index weight information or the huge external resources like thesauri to prioritize the Boolean search result on the Web.

2. Tag-based Boolean Search

2.1. Introduction to Tag-based Boolean Search

In this study, tag-based Boolean matching is proposed so as to achieve the object suggested in Introduction. Remember user’s request or given question before detailed description

- A user submits request of document type and intends to find data closely related to the document.
- The relevant system provides only typical Boolean search, and does not use any weight information or ancillary information in calculating search result.
- The user wants to have list-type search result.
- Searching should be carried out almost in real time.

Figure 1 shows the key idea and the procedure for solving the problem. The process from user’s submitting a document to providing search result gets through 3 steps.

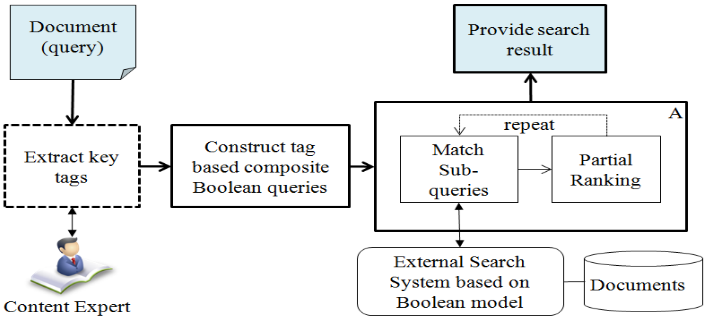


Figure 1. Schematic diagram of tag-based searching

- Step 1. Since the input document provided by the user is of the type that is difficult to be used in Boolean matching, key tags (important words) are then extracted.
- Step 2. Combine key tags to create composite Boolean queries of the possible type.
- Step 3. Differentiate query groups depending on the number of tags used in combinations to use the tag information for prioritizing search result.

In Step 1, key tags are extracted from the query document provided by the user. Key words are the information identified by the search engine or a program, but tags are the information given by a contents editor. The method that a system can automatically give key words [10] may be used, but relevancy with contents may be enhanced with key words which can express the relevant document best and given by the expert [11]. Also, such tags can be used for classification and searching. The selection used in this study is the method of extracting tags by contents experts, chosen from the aforementioned methods of extracting tags. The reason is that prior-extraction of key tags is significant because the query document to be searched does not frequently change.

In Step 2, the Boolean query differentiated on the basis of tags is constituted. All of sub-queries which can be created with the tag relationship are created to use the sub-queries for searching. Only the AND operator is used in constituting queries to reduce query complexity and ensure result accuracy. If it is allowed to use the OR operator, the number of derived sub-queries and the result list increases by several times as compared to the case of using only the AND operator. It is natural that more search results contribute to increase user's searching time and low user's satisfaction.

Step 3 is the process of matching sub-queries and integrating partial ranking results, and repeated as many times as the number of sub-queries. An external search system is used in matching sub-queries. Because sub-queries are created as many as the number of extracted tag combinations, query processing takes long time. Sub-queries are calculated and stored in advance in order to quickly provide search result to users. Sub-queries of high frequency are first processed on the basis of the statistics information on the usage. We will omit the detailed description in this paper.

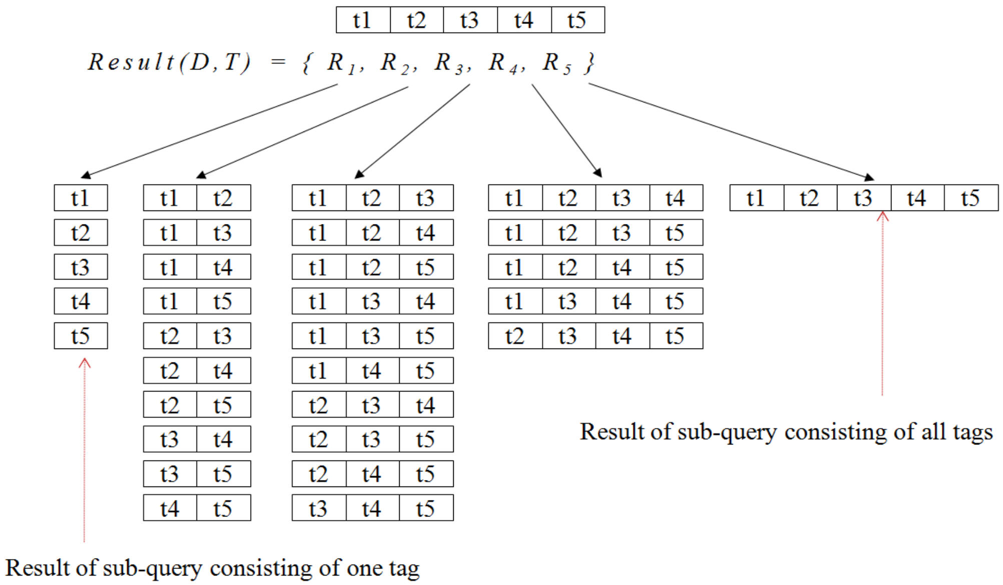
2.2. Constructing Tag-based Boolean Query Sets

The method of constituting a composite Boolean query will be described in the following to search result from a target document by using the tag information extracted from the query document.

The matching result set *Result(D,T)* of the entire tags *T* for the entire documents *D* to be searched is defined with equation (1). *Result(D,T)* consists of the set of partial search result R_i, where R_i is the partial search result set of *i* tags. *Result(D,T)* consists of sub-$\sum_{i=1}^{N} C(N,i)$ sets. For example, if *N* is 5, R_3 represents the partial search result set where only 3 tags are used, and R_3 has 10 sub-sets according to the combination *C(5,3)*. Figure 2 shows all of the partial search results created if *N* is 5 and $t_1, t_2, \ldots, t_5 \in T$.

$$RESULT(D,T) = \{R_1(D,T), R_2(D,T), \ldots, R_i(D,T), \ldots, R_N(D,T)\} \qquad (1)$$

where T consists of maximum N tags.



Result of sub-query consisting of all tags

Result of sub-query consisting of one tag

Figure 2. Exemplary method of constructing Boolean query set by means of tags

2.3. Result Matching for Composite Query Created on The Basis of Tags

In the following, the method of providing prioritized result will be described by using the composite Boolean query created in 2.2 to match the partial result sets. The tag matching technique to be described has the following 3 assumptions.

- Higher probability of good result as the number of matching tags is greater.
- The same number of matching tags results in the same weight.
- It is possible to give a tag a weight showing its significance. In this case, this precedes the assumptions one and two.

R_i which constitutes the result set *Result(D,T)* is designed to reflect the three assumptions defined in the above to provide a group weight. Similarity for the R_i group is defined with $sim(D,T_{and(p)})_i$. In this study, the equation in the extended Boolean model [5] is modified to conform to the current situation. R_i uses only the AND operator, and similarity for R_i in the proposed Boolean model is calculated with equation 2.

$$Sim(D,T_{and(p)})_i = 1 - \frac{(1-wt_1)^p + (1-wt_2)^p + \ldots + (1-wt_i)^p + \ldots (1-wt_N)^p}{(wt_1^p + wt_2^p + \ldots + wt_i^p + \ldots + wt_N^p) + N}) \qquad (2)$$

where wt_i represents a word weight for t_i, provided that $1 \leq i \leq N$ and $0 \leq wt_i \leq 1$, respectively. In a sub-group consisting of one tag word, it is possible to keep the normalized weight of which the search similarity for the group is between 0 and 1 by applying the N value. If the N value is omitted, it is impossible to ensure the range of normalized weight. The p value is used to make a difference between typical Boolean search and similarity search in the extended Boolean model [5]. It is necessary to allow the p value to be adjusted to make a difference among the search groups, so as to make a great weight deviation between tag words t_i. In this case, larger p results in larger weight deviations between search groups. However, if the tag weight is fixed to *1*, the partial search result does not highly depend on *p*.

3. Experimental Results

3.1. Experiment Environment

Query and search target collection is used in this experiment. The search query collection provided by NTIS R&D outcome service is targeted at the emerging technology information, which consists of fifteen hundred records that include thirteen key items [13]. The search target collection consists of 160 thousands records provided by NDSL, which includes the latest trend analysis information in the field of S&T (Science & Technology) and policies for S&T, and differentiated trend information from 1999 [14]. The experiment was carried out with an Intel PC (Intel Core 2 Quad CPU 2.83GHz, 8GB memory).

3.2. Extracting Key Tag Words

S&T technology contents experts construct tag information by tagging search query collections. As the analysis on the tag information consisting of query documents, documents have one to eight tag words and most of documents consist of two to six tags. The unique number of tag words is 3,341 and 2,506 tags appear in only one document. The number of tag words that appear very frequently in more than ten documents is approximately forty-eight. For example, the tag ‘nano’ appears in forty query documents and exhibits the highest appearance frequency. Exemplary tags of high frequency include system, protein, gene, sensor, network, etc. It implies that these tags are highly discriminative and well generated.

3.3. Selecting Sub-query Parameters

The process of determining *p* and the threshold of tag weight *wt* used in equation (2) will be described in the following. First, the process of determining *p* is derived through simulation. If $N=4$, change *p*. In this case, changing values of document similarity are described in Table 1. Each tag weight at this time is 0.5, 0.9, 0.3, and 0.8, respectively. It is seen that greater *p* results in distinctive discrimination between search groups, by checking changes of partial similarity values depending on *p*. If the approach is adopted that the more number of coupled tags results in better result, it is necessary to employ greater *p*.

The tag weight *wt* is used for prioritizing tag words, rather than for discrimination between search groups. In the underlined items in Table 1, the result (0.571) obtained by using three tags if $p=1$ shows lower similarity than the result (0.596) obtained by using two tags. This implies that *t2* is a word having a priority over *t1* or *t3*. From the experiment by using various available tag weight combinations, it is identified that tag weight *wt* is negligible if *p* is greater than about *3*.

Table 2 shows changing partial similarity where the tag weight *wt* is fixed to *1* and p changes. In this case, discrimination between groups is achieved regardless of p and changing p affects the range of similarity numerals visibly shown.

In conclusion, the tag weight *wt* can be used for determining a tag priority where a content expert selects tags. The effect is to identify important tags to present the relevant search result on top. The experiment revealed that setting *p* to *1* or *2* is better option.

Table 1. **Calculating partial similarity (N=4, wt=0~1, p=1~100)**

Tag Depth	Tag Weight				Sub-Similarity				
	t1	t2	t3	t4	p=1	p=2	p=3	p=10	p=100
4	0.5	0.9	0.3	0.8	0.769	0.631	0.554	0.395	0.310
3	0.5	0.9	0.3	-	0.596	0.417	0.330	0.134	0.014
	0.5	0.9	-	0.8	0.710	0.522	0.404	0.139	0.014
	-	0.9	0.3	0.8	0.667	0.473	0.365	0.136	0.014
	0.5	-	0.3	0.8	0.571	0.402	0.319	0.129	0.014
2	0.5	0.9	-	-	0.519	0.332	0.241	0.075	0.007
	0.5	-	0.3	-	0.333	0.205	0.159	0.066	0.007
	0.5	-	-	0.8	0.491	0.316	0.228	0.069	0.007
	-	0.9	0.3		0.462	0.286	0.210	0.073	0.007
	-	0.9	-	0.8	0.596	0.387	0.274	0.077	0.007
	-	-	0.3	0.8	0.431	0.269	0.197	0.068	0.007
1	0.5	-	-	-	0.222	0.126	0.088	0.028	0.003
	-	0.9	-	-	0.367	0.209	0.141	0.036	0.003
	-	-	0.3	-	0.140	0.076	0.060	0.027	0.003
	-	-	-	0.8	0.333	0.191	0.126	0.031	0.003

Table 2. **Calculating partial similarity (N=4, wt=1, p=1~100)**

Tag Depth	Tag Weight				Sub-Similarity				
					p=1	p=2	p=3	p=10	p=100
4	1	1	1	1	1	1	1	1	1
3	1	1	1		0.857	0.622	0.477	0.177	0.019
2	1	1			0.667	0.423	0.307	0.104	0.011

Table 3. Number of average search results per query document

Top Search Result	Tag coupling								Sub-total
	1	2	3	4	5	6	7	8	
10	22.04	30.41	15.63	4.55	0.87	0.10	0.01	-	73.62
20	41.25	54.59	25.97	6.75	1.11	0.10	0.01	-	129.78
30	59.36	76.58	34.28	8.38	1.22	0.10	0.01	-	179.93
100	168.91	194.79	70.78	13.26	1.26	0.10	0.01	-	449.11
Unlimited	1,77.90	948.10	169.99	16.42	1.26	0.10	0.01	-	2,907.80

3.4. Running Sub-queries

It is possible to create 29,064 sub-queries by using tag information extracted in 3.2. The time for searching the entire sub-queries in the experiment environment was measured and took approximately eighteen hours. It is estimated that one query comprised approximately nineteen sub-queries, and it took approximately forty-three seconds for one query document on the average. It is necessary to apply the cache policy to the sub-queries so as to provide search result within the time that a user can accept.

The search results were analyzed with the query documents. Where the number of tag collections was not considered, one query document showed 2,908 search items on the average 44,533 items to the maximum, one item to a minimum. Tables 3 describe the search result in various ways in consideration of the number of tag couplings. In the Tables, the top search result represents the number of searches to be used for limitation in each sub-query. When a user checks the search result, no problem occurs if he mainly checks the result of higher tag couplings (5~7), but he may have a problem due to the huge number of tag couplings to be checked if he checks the result of lower tag couplings (1~3). As such, for low tag couplings, it is necessary to limit the number of search results to be used in sub-queries in each group. Since search results in one sub-query have the same similarity, it is necessary to specify the optimum number of the results in a heuristic way within top 30 results.

It is necessary to check the entire search results for key words that a user is interested in the typical methods. With the method proposed in this study, a user can check important documents according to the high priority of tag couplings, and check more documents while lowering the level of tag couplings. In Table 3, it is necessary to check 2,900 results in typical methods, but to check the result of 7 tag couplings in the proposed method, and the result of one tag coupling in the worst case. In general, it is possible to satisfy user's search request by checking the search result of at least 3 tag couplings. As described above, user's satisfaction in search was improved by ensuring preferential checking of the most similar documents and checking of minimum result. In fact, the R&D Outcome Information Service of NTIS [13] implemented this method to apply it to the service for providing trend analysis information related to specific emerging information. The service was provided to a plurality of users for approximately one year, and revealed high satisfaction.

4. Conclusion

This study proposes a method of providing prioritized search result by means of Boolean search that does not provide index weight information. The query document presented by a user is expressed with tags given by experts, and composite Boolean queries are created by using the tag information. The algorithm for matching composite queries depending on tag couplings is used to prioritize groups. The advantage of this study is that priority information is easily provided by means of less information than for Boolean search which extends similarity search or thesauri. Application to NTIS revealed satisfactory effects.

We will make an attempt to find a method of describing search performance objectively, not empirically. It is necessary to study a method of reducing the huge number of queries when a great number of queries are created as the number of tags is greater. It is also necessary to present a special user interface for checking search result depending on tag couplings.

Acknowledgements

This work was supported by Construction of NTIS (National Science & Technology Information Service) Program funded by the National Science & Technology Commission (NSTC) (N-11-NM-LU01-C01).

References

[1] G. Salton and M. J. McGill, Editors, “Introduction to Modern Information Retrieval”, McGraw Hill, New York **(1983)**.

[2] L. Page, S. Brin, R. Motwani and T. Winograd, Editors, “The PageRank citation ranking: Bringing order to the web”, Proceedings of WWW 1998, **(1998)** April 14-18; Brisbane, Australia.

[3] R. Baeza-Yates and B. Riberiro-Neto, Editors, “Modern Information Retrieval”, Addison-Wesley **(1999)**.

[4] National Discovery for Science Leaders(NDSL), http://www.ndsl.kr.

[5] G. Salton, E. A. Fox and H. Wu, Communications of the ACM, vol. 36, no. 1022, **(1983)**.

[6] A. Bookstein, J. ASIS, vol. 31, no. 275, **(1980)**.

[7] W. G. Waller and D. H. Kraft, “Information Processing and Management”, vol. 15, no. 235, **(1979)**.

[8] J. Yefeng and W. B. Croft, Editors, “An association thesaurus for information retrieval”, Proceedings of RIAO **(1994)**.

[9] B. Lee, Journal of Kyungwon College, vol. 21, no. 159, **(1999)**.

[10] Z. Xu, Y. Fu, J. Mao and D. Su, Editors, “Towards the semantic web: Collaborative tag suggestions”, Proceedings of the Collaborative Web Tagging Workshop, **(2006)**; Edinburgh, Scotland.

[11] C. Cattuto, D. Benz, A. Hotho and Gerd Stumme, Lecture Notes in Computer Science, vol. 5318, **(2008)**, pp. 615.

[12] National Science & Technology Information Service (NTIS), http://www.ntis.go.kr.

[13] National R&D Outcome Service, http://roots.ntis.go.kr.

[14] NDSL Trend Service, http://radar.ndsl.kr.

Authors

Won-Kyun Joo

He received the B.S. and M.S. degrees in Computer Science from Chungnam National University, Korea in 1997 and 1999 respectively. Currently, he is a senior researcher at KISTI.

MinWoo Park

He received the B.S. and M.S. degrees in Computer Science from Chungnam National University, Korea in 1996 and 2004 respectively. Currently, he is a senior researcher at KISTI.

KiSeok Choi

He received the B.S. degree in Computer Science and Statistics from Seoul National University, Korea in 1988 and M.S. degree in Computer Science from KAIST in 1997. Currently, he is a department manger of R&D System Development, KISTI.

Yong Kim

He received the B.S. degree in Library and Information Science from Chonbuk National University in 1993, M.S. degree in Information Science from University of North Texas in 1995, M.S. degree in Computer Science from Chungnam National University in 2000, and the Ph.D. degree in Library & Information Science from Yonsei University in 2006. He joined the Chonbuk National University as a faculty member of the Library and Information Science Department in September 2008.

Young-Kuk Kim

He received the B.S. and M.S. degrees in Computer Science and Statistics from Seoul National University, Korea in 1985 and 1987 respectively and the Ph.D. degree in Computer Science from University of Virginia, Charlottesville, Virginia in 1995. After his Ph.D., he visited VTT Information Technology, Finland and SINTEF Telecom & Informatics, Norway as an ERCIM research fellow during 1995-1996. He joined the Chungnam National University as a faculty member of the Computer Science Department in March 1996.