Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 14, No. 2, June 2014, pp. 122-129
http://dx.doi.org/10.5391/IJFIS.2014.14.2.122

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

A Context-Awareness Modeling User Profile Construction Method for Personalized Information Retrieval System

Jee Hyun Kim[1], Qian Gao[2], and Young Im Cho[2]
[1]Department of Computer Software, Seoil University, Seoul, Korea
[2]Department of Computer Science, The University of Suwon, Hwaseong, Korea

ljfis

Abstract

Effective information gathering and retrieval of the most relevant web documents on the topic of interest is difficult due to the large amount of information that exists in various formats. Current information gathering and retrieval techniques are unable to exploit semantic knowledge within documents in the "big data" environment; therefore, they cannot provide precise answers to specific questions. Existing commercial big data analytic platforms are restricted to a single data type; moreover, different big data analytic platforms are effective at processing different data types. Therefore, the development of a common big data platform that is suitable for efficiently processing various data types is needed. Furthermore, users often possess more than one intelligent device. It is therefore important to find an efficient preference profile construction approach to record the user context and personalized applications. In this way, user needs can be tailored according to the user's dynamic interests by tracking all devices owned by the user.

Keywords: User profile, Contextcontext-awareness, Multimulti-agent, Personalized, Big data

Received: Jun. 2, 2014
Revised : Jun. 24, 2014
Accepted: Jun. 24, 2014

Correspondence to: Young Im Cho
(ycho@suwn.ac.kr)

1. Introduction

Gartner research firm describes "big data" as the volume, variety, and velocity of structured and unstructured data pouring through networks into processors and storage devices, along with the conversion of such data into business advice for enterprises. However, it is difficult to find an appropriate or satisfying method to organize, manipulate, and manage big data. An appropriate platform for processing the big data personalized information retrieval system is necessary and important.

Traditional personalized query refinement technologies are based either on search results or on some form of knowledge structure. None of them account for the tremendous increase in the number of intelligent devices that a user has. A user often has more than one intelligent device, such as a desktop computer, notebook, smart phone, or pad. It is not sufficient to determine individual preference based only on personal information stored in one computer. Therefore, a personalized query refinement strategy should be studied that can comprehensively consider all devices a user has in order to determine user preference.

Historically, data analytics software has been incapable of using an entire large data setor at least most of itto compile complete analysis for a query. Instead, it has relied on representative

samplings, or subsets, of the information to render these reports, even though analyzing more information produces more accurate results. That approach is changing with the emergence of new big data analytics engines, such as Apache Hadoop, LexisNexis HPCC systems, and the 1010data cloud-based analytics service. These new platforms are eliminating the role of summarization and providing complete views of big data sets.

Query expansion techniques can be broadly classified into two categories: those based on search results, and those based on some form of knowledge structure. The former depends on the search process and uses relevance feedback from a previous search iteration as the source to identify the query expansion terms [1, 2]. The latter is independent of the search process; moreover, additional query terms are derived by traversing a semantic network built according to a knowledge structure. Knowledge structures used by this group of techniques can either be a general-purpose ontology (or thesaurus) [3], an ontology built for a specific domain [4], or an ontology constructed from document collection based on term clustering [5].

In this paper, we construct an architecture to efficiently process large-scale data sets according to actual user demand and realize seamless integration of different devices that belong to one user. Based on this architecture, we construct the user profile, which records the user's access logs and retrieval habits or areas of interest. We then update the user profile according to this behavior and share the dynamic profile with each user device.

2. The Proposed Modeling Method

2.1 Multi-agent Personalized Query Refinement Approach

We propose a multi-agent-based approach to address the limitations of traditional query refinement technology. Our approach first expands the initial query according to the semantics of the user's input query. It then expands the preliminary refined query by comprehensively considering three methods for determining term and document frequency, lexical compounds, and sentence selection. These will be applied to the documents stored in all intelligent devices of the given user.

The framework and workflow of the proposed multi-agent personalized query refinement approach in the big data environment are shown in Figure 1.

Step 1: User 1 submits the initial query in the form of "UseID.Query" to the client agent. For example, with User001. news ("001" is the user ID and "news" is the initial query; these are shown as Path ① in Figure 1), the client agent verifies the user identities and informs the device tracker of the devices belonging to User 1 (shown as Path ② in Figure 1).

Step 2: The device tracker runs a simple loop that periodically sends heartbeat-method calls to every device belonging to User 1. The heartbeat response from each device informs the device tracker that a device is alive; additionally, the device double as a channel for messages. If the device is alive, its active degree (as "UserID.ADj," with j being the device index) is plus one, which is used to determine the importance of the expanded terms created by the corresponding device. As part of the heartbeat, the device tracker indicates whether a device is ready to run a query expanded task; if it is, the device tracker informs the query expand agent.

Step 3: The query expand agent first expands the user's query using knowledge-based query expansion. It then returns the preliminary refinement query to the user. After user feedback, the knowledge-based query expansion method creates an ontology-based query expansion set ("OETS," as shown as Path ③ in Figure 1). The query expand agent sends the ontology-based query expansion set "OETS" to the user device-based query expansion subagent (shown as Path ④ in Figure 1) to create user device-based query expansion set "UETS" according to the documents and browsing history stored in the device. (Suppose User 1 submits the query and that the desk computer and smart phone are alive. The query expand agent then expands the query according to data stored in the above two devices.)

Step 4: After user device-based query expansion, the user device-based query expansion subagent sends "UETS" to the weighted query expansion subagent (shown as Path ⑤ in Figure 1). The weighted query expansion subagent further determines the final refinement query according to active degree "UserID.ADj" of each device. It then copies both the initial query and refinement query to a shared filesystem (shown as Path ⑥ in Figure 1). Therefore, the next time the same user submits the same query, the query expansion agent can directly expand the query according to the shared filesystem.

Step 5: For more effective query expansion, when the device tracker monitors a previously unalive device that is currently alive, it informs the query expansion agent, which implements the same process as above to again expand the initial query and update the shared file system.

Step 6: In this step, a refined query is used as the final query to retrieve academic papers that have been processed by the big data process platform (shown as Path ⑥ in Figure 1).

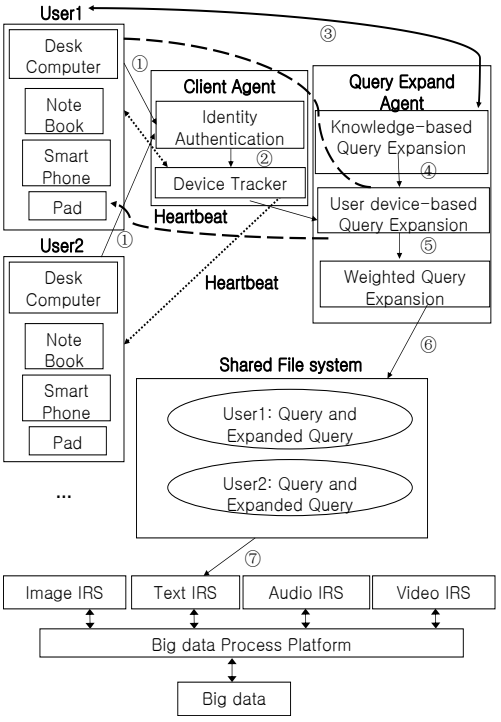

Figure 1. Framework of the multi-agent personalized query refinement approach in a big data environment.

3. Multi-agent Functions

Assume $O_1, O_2, O_3, \ldots, O_n$ are the ontologies of the domain $D_1, D_2, \ldots, D_n$, respectively, and $T_i = \{t_{i1}, t_{i2}, \ldots\}, i(1, n)$ comprise the terms set determined from ontology O_i. Assume $Q = \{q_1, q_2, \ldots, q_m\}$ is the initial query of the user. If $Q \cap T_i = \varnothing$, $1 \leq i \leq n$, the query has no relevance to this specific domain. Thus, we can obtain all relevant domains D_{f1}, D_{f2}, $\ldots$, D_{fk} (here, $Q \cap T_{fj} \neq \varnothing$, $f1 \leq fj \leq fk$). Then, add the name of relevant domain $D_{f1}, D_{f2}, \ldots, D_{fk}$ as the additional feature of the initial query and return them to the users.

Assume the suggested query set is $((Q, D_{f1}), (Q, D_{f2}), \ldots, (Q, D_{fs}))$. If different users input the same query, the query returned by the users is $((Q, D_{fi1}), (Q, D_{fi2}), \ldots, (Q, D_{fis}))$ after their selection. Finally, after feedback of the different users, the modified query set will be $(\alpha_1(Q, D_{f1}), \alpha_2(Q, D_{f2}), \ldots \alpha_s(Q, D_{fs}))$, where α_i is the weight of the query in a certain domain. The more attention users receive, the bigger the α_i value is, and the better that value can effectively reflect the needs of users. After a period of modification, the weight of the modified query that is smaller than a set value will be discarded. We typically consider these domains as the deviation of the definition about the query. Therefore, in the latter, when a certain user inputs the same initial query, the knowledge-based query expansion subagent will not consider the discarded domain.

With this method, the system can determine the domain about which most users are concerned, thereby providing more accurate information for the retrieval.

However, for a certain user, the query returned by the query input user must reside in an exact domain and in the form of (Q, D_{fi}). Considering that the initial query as provided by the user may be an inadequate representation of the user's information needs, the knowledge-based query expansion subagent will then expand the query returned with hypernym (kind-of relationship), hyponym (part-of relationship), and allomorph (instance-of relationship) according to the ontology tree. Hence, after this kind of expansion, the candidate query set may be OETS $= \{\text{Term1}, \text{Term2}, \ldots, \text{Term } n\}$.

The above knowledge-based query expansion only considers the semantic expansion of the query. The following part shows that the user device-based query expansion will consider the personal interest of all intelligent devices for the same user. Here, we adopt a three-level query expansion.

The first level is based on term frequency (TF) and document frequency (DF). We independently associate a score with each term in the user's device document based on TF and DF to match the similarity between them and the user's web query, thereby finding the candidate terms that can be used to expand the user's query.

For academic papers, more informative terms tend to appear toward the beginning; therefore, the position of the term appearing in these papers should be considered. Thus, we adopt a TF-based term score by multiplying the actual frequency of a term with a position score. The TF-based score is calculated by Eq. (1):

$$\text{Termscore} = \frac{\text{tnwords} - \text{pos}}{\text{tnwords}} \cdot \log(1 + TF) \qquad (1)$$

Here "tnwords" is the total number of terms in the document stored in one of the user's intelligent devices, "pos" is the position of the first appearance of the term, and TF is the frequency of each term in the document stored in one of the user's online intelligent devices that can match the user's web query.

Terms larger than predefined threshold TTS will be considered as the candidate terms that can be used to expand the query. Thus, after this level of expansion, the candidate query set may be ETS $= \{\text{Term1}, \text{Term2}, \ldots, \text{Term } n, \text{Term } n + 1, \ldots, \text{Term } n + m\}$.

The second level is based on lexical compounds. It can be used to automatically identify key concepts over the input document set according to the lexical dispersion of an expression; i.e., the number of different compounds in which it appears within a document or group of documents. It has been shown that simple approaches based on noun analysis are almost as effective as highly complex part-of-speech pattern identification algorithms. Therefore, after this level of expansion, the candidate query set may be ETS $= \{$Term1, Term2, $\ldots$, Term n, Term $n+1$, Term $n+2$, $\ldots$, Term $n+m$, Term $n+m+1$, $\ldots$, Term $n+m+s\}$.

We generate sentence-based summaries by ranking the document sentences according to their salience score. For academic papers, more important sentences tend to appear in the beginning. Here, we therefore consider the position of the sentence and the matching degree of the sentence, as well as the user's initial query. The sentence-based score is calculated by Eq. (2):

$$\text{sscore} = \begin{cases} \frac{(\text{average(NS)}-\text{pos})}{\text{average}^2\text{(NS)}} + \frac{QSN^2}{QTN} & \text{first 15 sentence} \\ 0 & \text{others} \end{cases} \tag{2}$$

Here "average(NS)" is the average number of sentences of all device items, "pos" is the position of the sentence, QSN is the number of query terms present in the sentence, and QTN is the total number of terms from the query.

Terms larger than predefined threshold TSS will be considered as the candidate terms that can be used to expand the query. Therefore, after this level of expansion, the candidate query set may be ETS $= \{$Term1, Term2, $\ldots$, Term n, Term $n+1$, $\ldots$, Term $n+m$, Term $n+m+1$, $\ldots$, Term $n+m+s$, Term $n+m+s+1$, $\ldots$, Term $n+m+s+t\}$.

For different intelligent devices owned by different users, the more frequently the intelligent device is used, the better it can reflect the preferences of the user. Therefore, in the stage of weighted query expansion, we calculate the score of the candidate term obtained at the stage of the user device-based query expansion. This is performed to determine the final extension term according to use frequency of the device.

For the intelligent device that is alive for a certain user, different intelligent devices may create different query expansion sets because different intelligent devices may store different documents and browsing history. The longer the intelligent device is online, the greater its contribution to determining the individual preferences of the user, and the larger the corresponding weight of the candidate query expansion set generated by the internal storage of documents and browsing history.

The weighted expanded query set is created by Eq. (3):

$$WETS = \bigcup_{i \in \text{alive device}} \beta_i \cdot UserID.ADi.ETS \cdot tagi \tag{3}$$

where,

$$\beta_i = \frac{UserID.ADi}{\sum_{i \in \text{alive device}} UserID.ADi} tagi$$
$$= \begin{cases} 1 & \text{if devicei alive} \\ 0 & \text{if devicei not alive} \end{cases}$$

For each term in "WETS," the weight smaller than 0.8 will be discarded. To avoid noisy suggestions, we limit the output expansion set to contain only terms appearing at least five times on the user's intelligent device; the remaining terms will be selected as the final expanded query term.

4. Experiments

To evaluate the performance of the proposed query refinement approach, we performed experiments several times. The test data was drawn from "TREC" conferences [6]. We used part of these data sets crawled in 1997 [7] for TREC 9 and 10, which have sets of ten topics and accompanying relevance determinations. For user input queries, we used the title field from each TREC topic; moreover, we used the Wilcoxon signed rank test to evaluate the significance of the effectiveness of the results [8]. Five computer science doctoral students and two master's students participated in the experiments.

We evaluated the performance of the proposed academic query expansion method on knowledge-based mapping between the input query and its semantic meaning, as well as the extracted important characteristics of the local document. In this experiment, we constructed an "HDFS" cluster to simulate our proposed method. We used a single computer with an Intel Core 4 CPU and 4 GB RAM as NameNode, and five computers with Intel Core 2 CPUs and 4 GB RAM as DateNode. We employed the Java programming language and Eclipse integrated development environment. We analyzed and operated the personalized ontology profiles using the Jena open-source framework. Our proposed system was distributed in IP networks. The client and server sides communicated with each other via TCP/IP and UDP protocols. The server side accessed the database side through a MySQL interface that supported remote access via IP networks.

The goal of this experiment was to assess the relevance of

the searched results returned to the user. For the baseline used as a comparison for our experiment, we used conceptual query expansion and two other lexical approaches for query expansion: synonym-based expansion (exact match) and WordNet-based expansion (soft match with Lesk-based similarity).

Conceptual query expansion is based on the knowledge base of concept networks. In this method, query terms are matched to those contained in the concept network; the concepts are deduced from the network and additional query terms are selected. On the other hand, synonym-based expansion expands the query by adding all synonymous expressions of the terms to the query. WordNet is an extensive lexical network of English words. Nouns, verbs, adjectives, and adverbs are organized into networks of synonym sets (synsets). Each synset represents one underlying lexical concept, which is interlinked with others with a variety of relations. Accordingly, WordNet-based expansion expands the query by synonym, hypernym, and hyponym within a limited length and depth.

Instead of using standard precision and recall measures to evaluate the performance of this experiment, we evaluated the top three and ten hits for each query and search method mentioned above. Furthermore, to evaluate the time cost of the proposed methods, we additionally compared the resource consumption of the above methods.

We defined four sets of queries to evaluate the performance of the search. These sets and their characteristics are listed below.

- Single concept: the query terms together identify a single concept in the personalized ontology profile.
- Multi-concepts: each single query term identifies a single concept in the personalized ontology profile.
- Similar concept: the query terms are closely related to one of the existing concepts in the personalized ontology profile.
- Image query: the query is not a text query; rather, it is a picture in any format, such as .jpg, .bmp, and so on.

For each of the three query sets, we defined two queries. The queries were executed with our proposed query expansion method, conceptual query expansion, synonym-based expansion, and WordNet-based expansion. The total number of test cases was eight. Users were required to evaluate 300 documents, including text and image documents. The experiment was performed under the premise of the user preference profile's convergence; accordingly, it could more precisely reflect

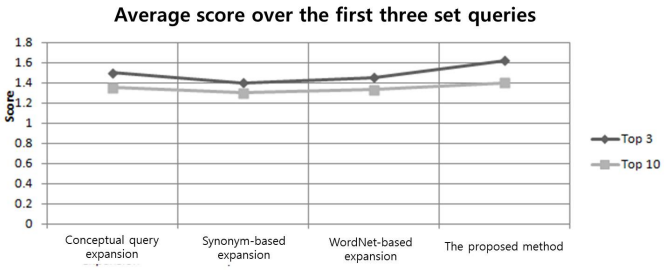

Figure 2. Average score for the top three and ten hits over the first three queries for each of the query expansion methods.

the user's interest. That is to say, realization of the proposed query refinement approaches was based on the convergent user preference profile. For the fourth set of the image query in our experiment, we used the Baidu image recognition search engine for the baseline comparison.

When users searched information about academic papers, they had to click the "academic paper retrieval" option in the "Advanced Search" page. Then, the experiment refined the initial input query based on user preference. This automatically refined the query and provided recommendations to the user according to the user's click and the given method.

The user could select some of them according to their demand; the experiment then realized the second instance of personalized information retrieval depending on the refinement query. More than 85% of our testers were satisfied with the expanded query terms.

Figure 2 shows the average score of each expansion technique mentioned previously with all queries of the second set. From the figure, we can see that the overall results were at least similar to, or better than, the other three query expansion methods for all query types. This is because the proposed method comprehensively considers the semantic relationship of the different terms and the semantic relevance between different terms in one input query. However, we can see that the performance of the proposed method on the top three was slightly better than that of the top ten. This may indicate that, in some academic papers, important sentences may also tend to appear near the end of the documents; therefore, it is not sufficient to only consider the words appearing in the beginning to expand the query.

Figure 3 shows the results of an average value calculated over all test subjects for the first query set using the four expansion techniques mentioned above. From the figure, we can see that Query 1 showed a significantly better result than the conceptual query expansion method and other two lexical approaches. This is because the user preference was built on the basis of the

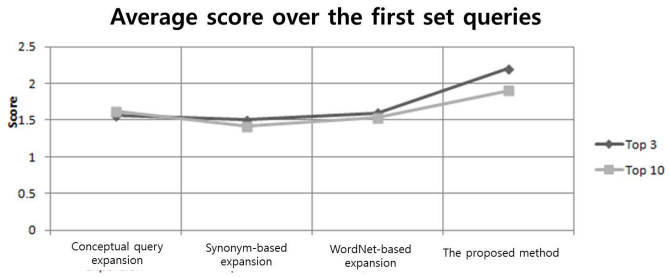

Figure 3. Average score for the top three and ten hits over the first set of queries for each of the query expansion methods.

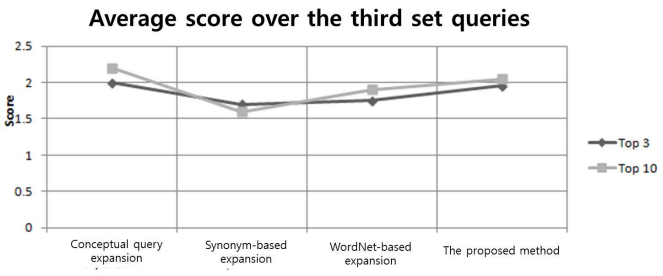

Figure 5. Average score for the top three and ten hits over the third set of queries for each of the query expansion methods.

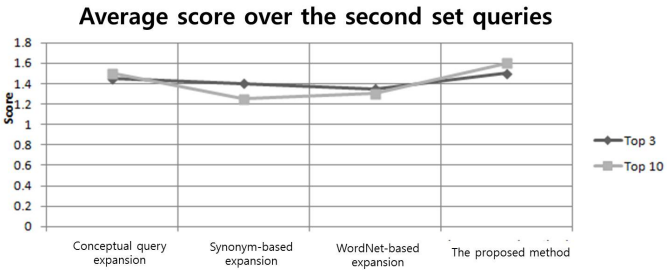

Figure 4. Average scores for the top three and ten hits over the second set of queries for each of the query expansion methods.

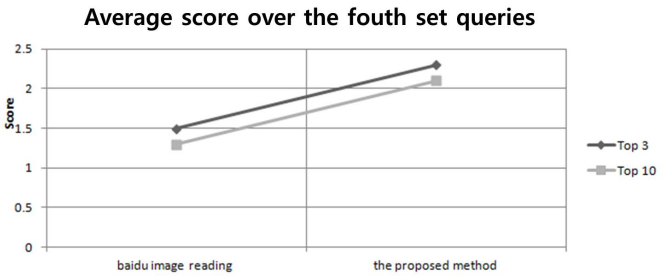

Figure 6. Average score for the top three and ten hits over the fourth set of queries with Baidu and the proposed method.

personalized ontology profile, which was closely related to the user's daily operating behavior. Moreover, the first set of the query was matched with the personalized ontology profile to conform to the user's interest. Therefore, the proposed query expansion method could more clearly comprehend the query and provide a more accurate expansion suggestion.

Figure 4 displays the results of an average value calculated over all test subjects for the second query set using the four expansion techniques mentioned above. From the figure, it is evident that our method performed better than the other three approaches. However, the difference was not significant.

Figure 5 shows the results of values calculated over all test subjects for the third query set using the four expansion techniques. We can see that the average score was less than that of the conceptual-based expansion method. The difference between the conceptual-based approach and our proposed query approach was not significant because the conceptual-based query expansion is based on concept networks of the knowledge base. In this method, query terms are matched to those contained in the concept network; concepts are deduced from the network and additional query terms are selected. Consequently, it can more clearly comprehend the semantic meaning of the two different terms of the same query. Moreover, our proposed method additionally considers the connection between two isolated query terms. Therefore, even though our method scored somewhat lower than the conceptual-based query expansion, the score difference was not large between the two methods. In addition, as shown in the figure, the top three ranked hits for three of the four approaches scored lower than the top ten ranked hits on average, which may suggest that the ranking of the documents was not optimal.

Figure 6 reveals the results of an average value calculated over all image test subjects for the third query set compared with the Baidu image recognition engine. Baidu has no personalized query expansion for image query; therefore, the average precision was significantly lower than that of the proposed method.

Furthermore, resource consumption of our approach ranked fourth among the four approaches for the first two sets of queries, third among the four approaches for the third set of queries, and first compared to Baidu image recognition.

5. Conclusion

In this paper, we proposed a multi-agent-based query refinement approach that determines the domain to which the initial query belongs. In this way, the method expands the query using knowledge-based query expansion. Moreover, it can compre-

hensively consider a user's interests according to all intelligent devices assigned to the user, thereby obtaining the optimized query expansion set. Furthermore, to address academic papers in big data, we used Hadoop as a platform to analyze and process large caches of data, which enabled the identification of a formalized model to represent different types of data and to realize academic paper retrieval in big data.

This proposed approach is the first step in creating a system implementing finely tuned query expansions. In the future, we will implement and test various query test terms to refine and redesign the proposed approach. Moreover, ontological relationship exploration is a vast area with many variations. Knowledge-based query expansion can be extended and enhanced in response to the results arising from the evaluation. Furthermore, more work should be conducted to filter expansion terms and thereby avoid ones that may be too generic. In this way, noisy information can be eliminated.

The results of a comparison between our method, the text-based retrieval baseline method, and the lexical-based query expansion method showed that our method is better than the other two methods in terms of average recall ratio and average precision ratio. Nevertheless, additional research should be conducted in the future to reduce the average response time and resource consumption.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgments

The present research has been conducted by the Research Grant of Seoil University in 2013.

References

[1] A. Spink, B. J. Jansen, C. Blakely, and S. Koshman, "A study of results overlap and uniqueness among major Web search engines," *Information Processing & Management*, vol. 42, no. 5, pp. 1379-1391, Sep. 2006. http://dx.doi.org/10.1016/j.ipm.2005.11.001

[2] D. Cai, C. J. van Rijsbergen, and J. M. Jose, "Automatic query expansion based on divergence,"in *Proceedings of the 10th International Conference on Information and Knowledge Management*, Atlanta, GA, November 5-10, 2001, pp. 419-426. http://dx.doi.org/10.1145/502585.502656

[3] E. Voorhees, "Query expansion using lexical-semantic relations," in *SIGIR 94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, B. Croft and C. J. van Rijsbergen, Eds. London, UK: Springer, 1994, pp. 61-69. http://dx.doi.org/10.1007/978-1-4471-2099-5_7

[4] K. Jrvelin, J. Keklinen, and T. Niemi, "ExpansionTool: concept-based query expansion and construction," *Information Retrieval*, vol. 4, no. 3-4, pp. 231-255, Sep. 2001. http://dx.doi.org/10.1023/A:1011998222190

[5] R. Mandala, T. Tokunaga, and H. Tanaka, "Combining general hand-made and automatically constructed thesauri for query expansion in information retrieval," in *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, July 31-August 6, 1999, pp. 920-925.

[6] D. Harman, "Overview of the second text retrieval conference (TREC-2)," *Information Processing & Management*, vol. 31, no. 3, pp. 271-289, May-Jun. 1995. http://dx.doi.org/10.1016/0306-4573(94)00047-7

[7] P. Bailey, N. Craswell, and D. Hawking, "Engineering a multi-purpose test collection for Web retrieval experiments," *Information Processing & Management*, vol. 39, no. 6, pp. 853-871, Nov. 2003. http://dx.doi.org/10.1016/S0306-4573(02)00084-5

[8] J. Zobel, "How reliable are the results of large-scale information retrieval experiments?," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne*, Australia, August 24-28, 1998, pp. 307-314. http://dx.doi.org/10.1145/290941.291014

Jee Hyun Kim received her B.S at Ewha womans university in 1978 and M.Sc. and Ph.D from the Department of Computer Science, Dankook University, Korea, in 1994, 2005 respectively. Now she is a professor at Seoil University. Her interesting area is SQM, DB, information retrieval etc.
E-mail: jhkim@seoil.ac.kr

Qian Gao received her B.S at Shandong University of Science and Technology in 2001 and M.Sc. at Shandong University and Ph.D from the Department of Computer Science, The University of Suwon University, Korea, in 2014. Now she works at Sandong University. Her interesting area is AI, Big data, information retrieval etc.

E-mail:james_qq @suwon.ac.kr

Young Im Cho received her B.S., M.Sc., and Ph.D from the Department of Computer Science, Korea University, Korea, in 1988, 1990 and 1994, respectively. She is a professor at The University of Suwon. Her interesting part is AI, Big data, information retrieval etc.

E-mail: ycho @suwon.ac.kr