Semantically Enhanced Keyword Search for Smartphones

Jihoon Ko[1], Sangjin Shin[1], Sungkwang Eom[1], Minjae Song[1], Dong-Hoon Shin[1],
Kyong-Ho Lee[1] and Yongil Jang[2]
Yonsei University, Republic of Korea[1]
LG Electronics, Republic of Korea[2]
{jhko, sjshin, skeom, mjsong, dhshin}@icl.yonsei.ac.kr,
khlee@cs.yonsei.ac.kr and yi.jang@lge.com

ABSTRACT
To apply semantic search to smartphones, we propose an efficient semantic search method based on a lightweight mobile ontology. Through a prototype implementation of a semantic search engine on an android smartphone, experimental results show that the proposed method provides more accurate search results and a better user experience compared to the conventional method.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

Keywords
Semantic search, smartphone, mobile ontology

1. INTRODUCTION
As smartphones are becoming popular, a large volume of mobile data is being created. Although there are studies on how to utilize mobile data, the conventional methods on mobile search on smartphones still have limitations [1, 2]. Most of the mobile platforms such as Android and iOS provide the well-known full text search (FTS), which has an advantage of being used intuitively. However, the keyword-based interface of FTS has limitations when a user tries to find the data that she really intends. Since FTS simply provides the search results that contain the given keywords as substrings, it does not represent a user's intention or the semantic relationship among keywords. It has difficulties in dealing with the cases where a user cannot remember the exact keywords about something to find or the number of search results is too many.

To overcome the limitations of the keyword-based FTS, researches on semantic search based on ontologies have been performed. Mobile devices do not have enough resources to process semantic data like RDF triples. It is hard to apply the existing semantic search techniques [3] to smartphones. Therefore, we propose an efficient semantic search method based on a lightweight mobile ontology. Mobile data are organized and stored into application databases such as contacts and calendar. The proposed ontology is a formal specification of the conceptual meaning of mobile data and the interrelationships among mobile databases. On the basis of the mobile ontology, the proposed method extracts the semantic relationship among the keywords given by a user. The prototype implementation of the proposed mobile semantic search method is evaluated and analyzed.

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577331

2. MOBILE SEMANTIC SEARCH
On mobile platforms each application stores users' data into its relational database and the databases are isolated for security reasons. Considering the limited hardware resource, the ontology contains a small number of concepts, which is essential for mobile semantic search. It is also devised in order to seamlessly integrate the isolated application databases. The proposed mobile ontology consists of a set of concepts and the semantic relationships among mobile databases. Specifically, the TN and FN concepts represent the table name and the field name of a table, respectively. The concept of type AC indicates an abstract concept which connects all FN concepts with properties in common. Figure 1 shows a part of the mobile ontology. Based on this ontology, our search method further infers semantic relationships among mobile data, which has a mapping relation with the keywords given by a user.

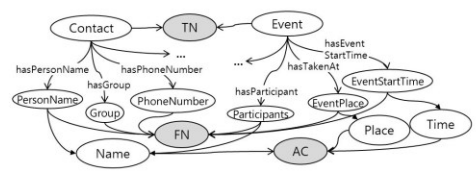
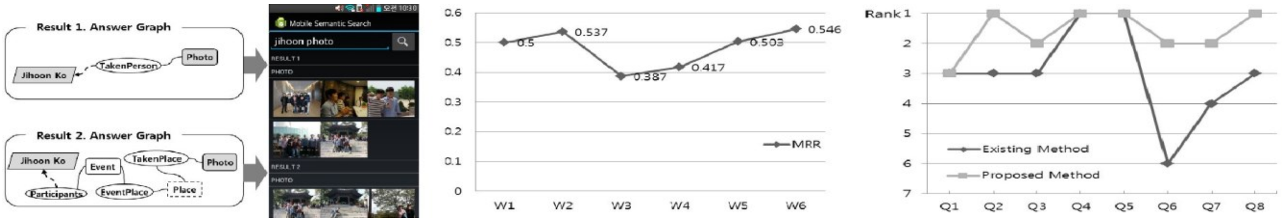


Figure 1. A part of the mobile ontology.

The proposed semantic search method consists of three steps. Firstly, it captures a user's query intention from the keyword input and extracts query graphs. Secondly, from a query graph, it computes answer graphs, which represent the user's query intention well. Finally, it makes SQL statements from each answer graph and executes them. All the search results are merged and displayed. The mobile ontology is used throughout all these steps. To optimize the performance, the proposed method selects a set of answer graphs, which are close to a user's query intention. For this purpose of ranking answer graphs, equation $\mathrm{Similarity}(K_A,G_A)$ computes a similarity between the keywords list K_A and an answer graph G_A acquire from a query graph G_Q. The equation is explained in detail in the following section.

$$\mathrm{Similarity}(K_A,G_A,G_Q) = w_1 * \mathrm{SimilarityWithData}(K_D) + w_2 * \mathrm{Correlation}(K_A) + w_3 * \mathrm{Size}(G_A) + w_4 * \mathrm{Popularity}(G_A)$$

2.1 User Query Translation
The proposed method takes a list of keywords as input and computes their mappings with ontology concepts. For the keywords unmapped with the ontology, the method searches for data in DB tables that contain the keywords as substrings. After that, they are also mapped to the FN concepts, which are connected to the corresponding fields that include the keywords as substrings. Through this step, the graph representation of the ontology is extended with the keywords. We call the extended graph a query graph G_Q.



a) an example of mobile semantic search. **b) MRR values by different weight sets.** **c) comparison with previous work.**

Figure 2. Experimental results.

Meanwhile, a keyword may be mapped to multiple concepts. It means that the same keyword may be duplicated in fields of different DB tables. As a result of considering all the possible cases, more than one query graph may be generated from a user's query. There may be query graphs different from a user's original intent. So we evaluate query graphs according to two criteria. First, for a list of keywords K_D that is mapped to field data, SimilarityWithData(K_D) computes the lexical similarity value with their matching fields via the Levenshtein distance metric. Correlation(K_A) computes the correlation degree of the adjacent keywords by considering how many of them belong to the same DB table.

2.2 Answer Graph Creation

In order to construct query statements, the proposed method computes answer graphs from the query graphs generated from the previous section. It finds subgraphs including all the concepts mapped to the given keywords. As the chain between two concepts get longer, the degree of their correlation decreases. Therefore, shorter chains have semantically stronger correlation [4]. We find K shortest subgraphs by applying the modified version of the Top-K answer graph traversing algorithm [4]. We estimate answer graphs by two measures: Size(G_A) computes the size of an answer graph G_A and Popularity(G_A) indicates how often the applications that correspond to the nodes of an answer graph are used by a user. An answer graph is composed of the concept nodes corresponding to application databases. Because current mobile platforms deploy different relational databases per applications, we have to execute SQL statements separately for application databases and merge and display the search results. Due to space limit, we omit how to construct a SQL statement from an answer graph.

3. EXPERIMENTAL RESULTS

To evaluate the performance and accuracy, we have implemented the proposed method on a smartphone (Android version 4.2.2, quad-core 2.26Ghz, 2GB RAM). The test set is composed of 8 queries. As experimental mobile data, we prepared 1000 contacts, 200 schedules, and 1000 photos. Figure 2.a shows two different answer graphs from a user query "jihoon photo". The first graph represents the photos taken by jihoon (person name) and the other indicates the photos taken at the place of the schedule that jihoon participated in. Although users originally intend to search for the first interpretation, she can additionally get another result, which may be meaningful to her, resulting in providing a better user experience on mobile devices than the conventional FTS does.

The factor K of our answer graph traversing algorithm has the most impact on the search speed. The increment of the K value creates more answer graphs and can provide more various search results. On the contrary, it slows the search speed and decreases the accuracy. So we performed experiments for selecting the optimal K value in terms of accuracy. We used the Mean Reciprocal Rank (MRR) to estimate the accuracy of search results. Here the higher MRR value indicates that the answer graph is located at the top of the alignment. A formula for computing a value of MRR is defined as

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

where Q is a set of queries and $rank_i$ is the rank of an answer graph intended by i^{th} of Q. the MRR value may be changed by the different selection of K and weight values (in Similarity(K_A, G_A)). To find the highest MRR while varying the weight values (six different sets), we set K to 3 and performed experiments. As shown in Figure 2.b, at the weight set W6 ($w_1 = 0.5$, $w_2 = 0.15$, $w_3 = 0.15$, $w_4 = 0.2$), we got the highest MRR. To find the highest MRR in terms of the K value, we set weights as W6 and performed experiments varying the value of K from 1 to 4. When K was 1, an answer graph intended by a user was not found because the answer graph that a user does not intend was ranked higher. The MRR values were evaluated as 0.728, 0.708 and 0.691 when we set K to 2, 3 and 4, respectively. When K was 2, we got the highest MRR. Also, we evaluated the performance for the test queries. The average search time was 0.655 seconds and the maximum time was 2.577 seconds.

Additionally, in terms of accuracy, we compared the proposed method with the work of Tran et al. [5]. While the work of Tran et al. does not target mobile devices, it is one of the most comparable works with ours. Figure 2.c shows that the previous method has 0.469 as the MRR value and our method is about 55% more accurate.

To cope with the limitations of FTS and the characteristics of mobile devices, we proposed a lightweight mobile ontology and a semantic search method. From the prototype implementation and experiments, we can find that our method successfully retrieves the data that a user intends to find from multiple application DBs based on the sematic relationships of mobile data.

Acknowledgements

This research is sponsored by LG Electronics.

4. REFERENCES

[1] S. Shin, J. Ko, D. Shin, and K. Lee, "Semantic Search for Smart Mobile Devices," *Proc. of the Companion Publication of Int'l Conf. on Intelligent User Interfaces*, pp. 95-96, 2013.

[2] S. Shin, D. Shin, J. Ko, M. Song, S. Eom, K. Lee, J. Park, S. Lee, and J. Jung, "Augmenting Mobile Search Engine with Semantic Web and Context Awareness," *Proc. of the IEEE/WIC/ACM Int'l Conf. on Web Intelligence*, pp. 278-279, 2013.

[3] J. Pound, A.K. Hudek, I.F. Ilyas, and G. Weddell, "Interpreting Keyword Queries over Web Knowledge Bases," *Proc. of the 21st ACM Int'l Conf. on Information and Knowledge Management*, pp. 305-314, 2012.

[4] T. Tran, H. Wang, S. Rudolph, and P. Cimiano, "Top-k Exploration of Query Candidates for Efficient Keyword Search on Graph-Shaped (RDF) Data," *Proc. of the IEEE 25th Int'l Conf. on Data Engineering*, pp. 405-416, 2009.

[5] T. Tran, D.M. Herzig, and G. Ladwig, "SemSearchPro-Using Semantic Throughout the Search Process," *Journal of Web Semantics*, Vol. 9, Issue. 4, pp. 349-364, 2011.