LETTER

Improving Keyword Match for Semantic Search

Hangkyu KIM[†a)], *Student Member*, Chang-Sup PARK[††], *and* Yoon Joon LEE[†], *Nonmembers*

SUMMARY Semantic search can be divided into three steps. Keyword matching, the first step, significantly impacts the search results, since the following steps are based on it. In this paper, we propose a keyword matching methodology that aggregates relevance scores of the related text to define the score of an object. Validity of the approach is shown by experiments performed with three public data sets and the detailed analysis of the results.

key words: *keyword matching, ontology, semantic search*

1. Introduction

Recently, keyword-based search approaches have been proposed in the literature of ontology and Semantic Web [1], [2], [8], [9]. In the methods, semantic search over a graph structured data such as ontology is often performed in three phases: keyword matching, semantic expansion, and resource mapping. In the first phase, an initial set of search results is found by keyword matching, which has a significant impact on the final results since it is exploited in the following phases to search for other semantically relevant ones. Nevertheless, most existing approaches use information retrieval (IR) techniques only to search text data related to individual objects and do not consider semantic relationships between objects and text, hence they often include unwanted objects in the result set or rank less relevant objects highly.

For effective semantic search, some important features of ontology should be considered in keyword matching. First, semantics of the data properties of objects should be exploited in computing relevance of the objects to a given query. Second, keyword terms implied in logical and readable identifiers such as URIs should be also used as the searchable index terms. Lastly, open world assumption [10] observes that since ontology is shared, updated and evolved through Web, information not described in ontology does not mean 'false', but 'unknown'. The previous keyword matching approaches do not consider these characteristics of ontology. In this paper, we propose a new keyword matching method utilizing the ontology features above mentioned. We take an object node-oriented approach and suggest a new relevance scoring function for objects in ontology in the consideration of the semantics of relationships between objects and related text data. We show by experiments that the proposed method outperforms the previous ones with respect to the quality of the search results.

The rest of the paper is organized as follows. We present limitation of the existing approaches and our motivation in the next section. We propose a new keyword matching methodology and a relevance scoring function in Sect. 3 and Sect. 4. Experimental results to demonstrate performance of our method are shown in Sect. 5. Finally we draw conclusions in Sect. 6.

2. Background and Motivation

In this paper, we model ontology as a directed labeled graph, $G(V, E)$. We have $V = V_O \cup V_D$, where V_O is a set of *object nodes* representing individual objects and V_D is a set of *data nodes* representing literal values related with objects. We have $E = E_O \cup E_D$, where E_O is a set of edges (u, v) from an object node u in V_O to another object node v in V_O and E_D is a set of edges (u, v) from an object node u in V_O to a data node v in V_D. We call the edges in E_O *object properties* and the edges in E_D *data properties*, respectively.

We consider that typical semantic search process consists of keyword matching, semantic expansion, and resource mapping. In *keyword matching* phase, keyword terms in the given query are matched to relevant object nodes which include one or more keyword terms in directly linked data nodes. *Semantic expansion* phase finds additional nodes which are semantically related to the set of nodes obtained in keyword matching and includes them in the result set by the way such as spreading activation [2], [7], authority transfer [5], and common root node search [6]. Finally, the *resource mapping* phase connects the abstract concept represented by the object nodes in the result set to

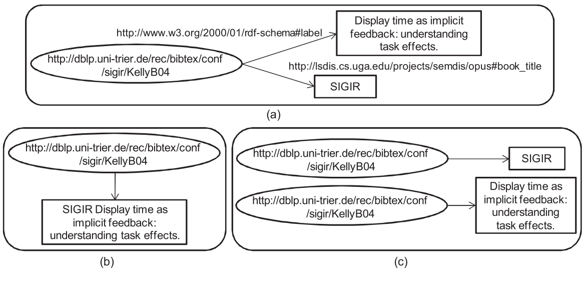

http://www.w3.org/2000/01/rdf-schema#label
Display time as implicit feedback: understanding task effects.
http://dblp.uni-trier.de/rec/bibtex/conf/sigir/KellyB04
http://lsdis.cs.uga.edu/projects/semdis/opus#book_title
SIGIR
(a)
http://dblp.uni-trier.de/rec/bibtex/conf/sigir/KellyB04
SIGIR Display time as implicit feedback: understanding task effects.
(b)
http://dblp.uni-trier.de/rec/bibtex/conf/sigir/KellyB04
SIGIR
http://dblp.uni-trier.de/rec/bibtex/conf/sigir/KellyB04
Display time as implicit feedback: understanding task effects.
(c)

Fig. 1 An example for data node-oriented keyword matching.

Manuscript received April 2, 2010.
Manuscript revised August 27, 2010.

[†]The authors are with KAIST, Daejeon, Republic of Korea.
[††]The author is with Dongduk Womens University, Seoul, Republic of Korea.
a) E-mail: hkkim@dbserver.kaist.ac.kr

DOI: 10.1587/transinf.E94.D.375

the real resources on the Web, such as documents, databases, and services. In this paper, we focus on keyword matching process which produces the initial set of object nodes and thus has a significant influence on the final result of semantic search.

The existing keyword matching methods can be considered *data node-oriented* since they search for relevant data nodes using traditional IR techniques and then return object nodes directly linked with the search results. The approaches can be divided into two groups, i.e. one that focuses on *coherence of data nodes* [1], [2], [8], and the other focusing on *independence of data nodes* [4]. Figure 1 shows an example of those approaches. In Fig. 1 (a), two data nodes are linked to the same object node by different data properties. The approach emphasizing coherence of data nodes merges them into one data node as shown in Fig. 1 (b) and then evaluate the relevancy of the single data node to the given keywords. Using a traditional IR technique, the relevance score of the merged data node with respect to a given keyword would be reduced than the original data node due to the increased length of the merged text string. For example, if 'SIGIR' is given as a keyword query, the object node in Fig. 1 (a) would get a high relevance score since one of its data node exactly matches the keyword term. In Fig. 1 (b), however, the object node would have a lower score since it has only a single data node where the keyword appears in a long text string.

Meanwhile, the other kind of approach focusing on independence of data nodes measures the relevancies of two data nodes independently as if they are associated with different object nodes, as shown in Fig. 1 (c). Thus, an object node can get a high relevance score only if all the keyword terms of a query appear in a single data node related with the object node. For example, the object node in Fig. 1 (c) partially matches the query 'implicit feedback SIGIR' and thus will be given a low score since none of the related data nodes contains all the keyword terms.

These observations show that the previous data node-oriented approaches have limitations in performing effective keyword matching based on the semantics of data properties between object and data nodes. We propose a new keyword matching scheme based on three strategies in next section.

3. Object Node-Oriented Keyword Matching

Object node-oriented approach

We use an object node-oriented approach to matching keywords in ontology, which is differentiated from the previous data node-oriented approaches. It supports both coherence and independence of data nodes related to an object node, without merging the data nodes or duplicating the object node. Our approach measures relevance scores of data nodes independently of each other considering independence and aggregates the scores with respect to the related object nodes considering coherence. This strategy overcomes the limitations of data node-oriented approaches. The detailed description of the relevance measure for object nodes are presented in Sect. 4.

Indexing on identifiers

In usual ontology, all objects have identifiers in the form of URIs, which often contain keywords describing the objects. In Fig. 1, for example, the object node representing the paper written by 'Diane Kelly' has a readable URI in which a keyword 'Kelly' appears. For more effective keyword-based search for ontology objects, we consider exploiting the identifiers of objects as supplemental data for indexing and search. For simplicity and consistency of keyword matching process, we insert a virtual data node for each object node which contains the object identifier, as shown in Fig. 2.

Weighting on data properties

In ontology there are semantic relationships between objects and data represented by data properties. We present a weighting mechanism on data properties considering importance and rareness of properties.

Importance of properties can be defined in different ways by ontology designers in schema level. In this paper, we consider three types of data properties: user-defined property, pre-defined property, and virtual property. User-defined properties are defined by the ontology designer. Pre-defined property is defined in the ontology language, such as 'rdfs:comment' and 'rdfs:label' in RDF. Virtual property means the property linking an object node and a virtual data node defined in Sect. 3. We weight user-defined property the highest, pre-defined property next and virtual property the lowest. The reason is that properties defined by users should be considered most important, while identifiers described in virtual data nodes may be defined by types and series of number, i.e. person1, person2.

For effective weighting on data properties, we also consider the rareness of data properties. It is measured by the number of *sibling* data nodes which are related to the same object node by the same property name. We consider that the smaller the number of sibling data nodes is (i.e. the higher the rareness of the data property shared by the data nodes is), the closer the semantic relationship between the data nodes and object node is. For example, if a paper object has one title and three sub-titles which are connected by 'title' and 'sub-title' properties, respectively, each sub-title is less relevant to the paper than the title is, since the paper has two more sub-titles. Note that the rareness is measured with respect to each property name since different property names mean different information on the object.

We present how the importance and rareness of data properties are exploited in the relevance function in the next section.

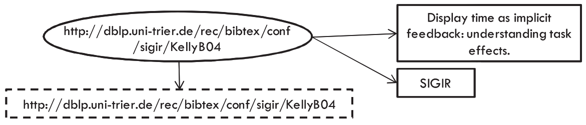

http://dblp.uni-trier.de/rec/bibtex/conf /sigir/KellyB04
Display time as implicit feedback: understanding task effects.
SIGIR
http://dblp.uni-trier.de/rec/bibtex/conf/sigir/KellyB04

Fig. 2 An example for virtual data node.

4. Relevance Scoring Function

We propose a new relevance function for object nodes in ontology based on the traditional TF·IDF measure [3] and three strategies which exploit ontology features described in Sect. 3. Let $D(o)$ be a set of data nodes related to an object node o, i.e. $D(o) = \{d | d \in V_D, \exists(o, d) \in E_D\}$, a relevance function $score(q, o)$ of a keyword query q and an object node o is defined as follows:

$$score(q, o) = \sum_{t \in q} \max_{d \in D(o)} (score(t, d)) \tag{1}$$

$$score(t, d) = \frac{1 + \ln(1 + \ln(tf))}{(1 - s) + s \cdot \dfrac{dl}{avdl}} \cdot \ln \frac{N+1}{df} \cdot \frac{w}{1 + \ln(ns)} \tag{2}$$

$$w = \begin{cases} 1 + 2\alpha & \text{if user-defined data property} \\ 1 + \alpha & \text{if pre-defined data property} \\ 1 & \text{if virtual property} \end{cases} \tag{3}$$

Note that the virtual data node is also included in $D(o)$. Independence of data nodes is guaranteed in measuring the relevance score of each data node, as well as coherence of data nodes is achieved by aggregating the measured scores with respect to the object node. Considering the open world assumption [10] mentioned in Sect. 1, we use the maximum value in aggregating relevance scores of the data nodes in $D(o)$ for a keyword term t. The open world assumption means that the fact not described is not false but unknown. Ontology is usually considered to satisfy this assumption since it is shared, updated, and evolves continuously. The updates of ontology have already been considered to the relevance score choosing the maximum in Eq. (1), since there needs no update unless a data node with higher relevance score is inserted. In Eq. (2), *tf*, N, *dl*, and *avdl* respectively denote the frequency of term t in the data node d, the total number of data nodes, the text length of d, and the average of *dl* over all data nodes. Symbol s is a constant, usually assigned as 0.20. Equation (2) incorporates the measure of both importance and rareness of data nodes into the relevance function used in document search. For importance of data nodes, the value of w is defined by the property type and a user-defined parameter α as shown in Eq. (3). The variable *ns* stands for the number of sibling nodes of d sharing the same property name. $score(t, d)$ is calculated in inverse proportion to it to take the rareness of data nodes into consideration.

5. Experiments

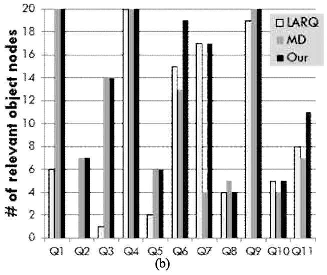
To evaluate performance of the proposed approach, we have conducted experiments using various ontology datasets. The experiments are performed in a machine with 2.40 GHz Intel Core2 Quad CPU, and 3.25 GB RAM. We implemented the proposed method in JAVA using Lucene[†] and Jena[††] libraries to index and access ontology datasets. For comparison, we also implemented the existing approaches shown in Fig. 1 (b) and (c). We call (b) as 'MD', which merges data nodes linked to the same object node and indexes it, and (c) as 'LARQ (Lucene + ARQ)', which is the approach provided by Jena. We used three different types of public ontology – SwetoDblp[†††], QuotatuinsBook[††††], and ODP[†††††] – in the experiments.

For the experiments, 11 queries are defined as shown in Fig. 3 (a). We used Q1 ∼ Q5 for SwetoDblp data set, Q6 ∼ Q8 for QuotationsBook, and Q9 ∼ Q11 for ODP. Though the selected data sets are used in many experiments, assessments to evaluate search validity are not shared in public. For the experiments, we defined the assessments of the queries checking the relevant nodes manually [11]. The number of relevant object nodes in top-20 query results for each test query and keyword matching method is presented in Fig. 3 (b). LARQ and MD approaches show different results according to characteristics of datasets and queries. Providing coherence, MD shows relatively good results in SwetoDblp, which has no exceptional long text in data nodes, with queries matching two or more data nodes. However, in QuotationsBook and ODP, which has noticeable long text in data nodes – such as quotation sentences or comments about Web documents –, LARQ shows better results, since the relevance matched in short data node does not lose the score by other long data nodes, guaranteeing independence. Note that, supporting coherence and independence, our approach returns most true sets in both of the two cases.

Precision and recall curves of Q1 are shown in Fig. 4 (a) and (b). X-axis of the graph stands for the number of nodes in result set. LARQ presents low quality because it did not consider coherence of data nodes. Let us focus on the difference between MD and our approach with a part of SwetoDblp dataset in Fig. 4 (c). Two ovals stands for two object nodes, which we abbreviate to 'Kelly' and 'Keskustalo', linked to two data nodes, label and book title, for each. When Q1 is given as a query, 'Kelly' node is matched by 'feedback' and 'SIGIR', while 'Keskustalo' node is matched by 'relevance' and 'feedback'. If MD method applied assuming data node lengths are the same, 'Keskustalo'

Q1	relevance feedback SIGIR
Q2	hidden Markov model ICML
Q3	visualization SIGMOD
Q4	web log usage mining
Q5	web log usage mining "intelligence"
Q6	Socrates Shakespeare life love
Q7	love mercy forgiveness forgive charity grace bible
Q8	vinci michelangelo mozart excellence genius ideas think thoughts vision inspiration
Q9	California national park
Q10	Halloween celebration festival
Q11	basketball league profile

(a)



(b)

Fig. 3 Queries and top-20 query results.

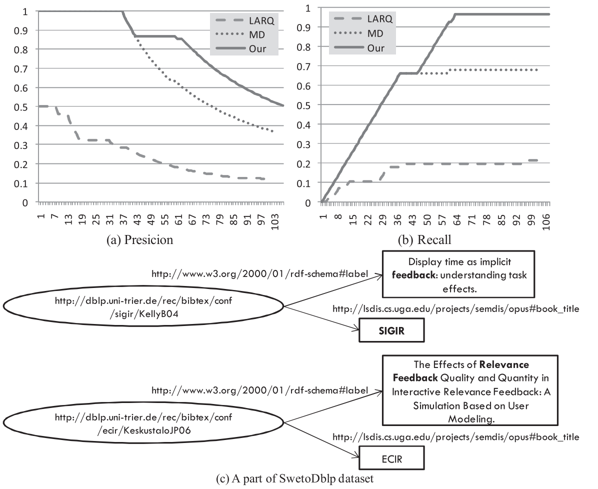
[†] http://lucene.apache.org/java/docs/index.html

[††] http://jena.sourceforge.net/

[†††] http://knoesis.wright.edu/library/ontologies/swetodblp/

[††††] http://quotationsbook.com/

[†††††] http://rdf.dmoz.org/



(a) Presicion

(b) Recall

(c) A part of SwetoDblp dataset

Fig. 4 Query result for Q1.

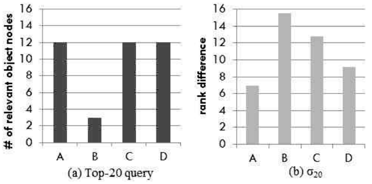
node is ranked highly since the terms are matched twice for each. However, if our approach applied for independence, ‘Kelly’ node is ranked highly because ‘SIGIR’ is matched fully in its own data node independently.

Indexing on identifiers and weighting on properties to consider importance of data properties and rareness of data nodes were exploited in our approach as heuristics to evaluate the relevance of object nodes. To show the effectiveness of each device, we reconstructed SwetoDblp and examined Top-20 precisions and ranking differences of the result sets. Author names were linked directly to article object nodes since the raw data set does not have enough sibling nodes. The ranking difference of the items in a result list A from the list of top-20 items from true set T is defined as follows:

$$\sigma_{20}(A) = \sqrt{\frac{\sum_{r=1}^{20} \{r - rank_A(item_T(r))\}^2}{20}} \tag{4}$$

where $rank_A(x)$ denotes the rank of item x in a list A, and $item_T(r)$ denotes the r-th ranked item in true set T. Note that, the value of $|r - R_A(x)|$ is bounded by 20 to limit the maximum of $\sigma_{20}(A)$ to 20.

Figure 5 presents experimental results of the methods excluding each heuristic (B, C, and D) in comparison with the method considering all heuristics (A). The given query is ‘Naish journals’ which means ‘the journals written by Naish’. Experiments with other queries showed similar results. Figure 5 shows that indexing on identifiers has most significant influence on both precision and ranking of the search result by supplemented keyword terms from URI. It also demonstrates that weighting on data properties and consideration of sibling data nodes improve the ranking of relevant objects in the result set, while they have little impact on the precision.



(a) Top-20 query

(b) σ_{20}

Fig. 5 Top-20 query and σ_{20}. A-fully exploited approach, B-without virtual node, C-without considering importance of data properties, D-without considering rareness of data nodes.

6. Conclusions

The existing, data node-oriented keyword matching approaches have limitations for searching ontology. We proposed a new keyword matching method including a new relevance function, based on an object node-oriented approach that supports both independence and coherence of data nodes as well as the characteristics of properties. The effectiveness of our approach including several heuristics was confirmed by experiments.

Acknowledgments

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. R01-2007-000-20135-0).

References

[1] R.V. Guha, R. McCool, and E. Miller, “Semantic search,” Proc. WWW, pp.700–709, 2003.

[2] C. Rocha, D. Schwabe, and M.P. de Aragao, “A hybrid approach for searching in the semantic web,” Proc. WWW, pp.374–383, 2004.

[3] A. Singhal, “Modern information retrieval: A brief overview,” IEEE Data Engineering Bulletin, Special Issue on Text and Databases, vol.24, no.4, Dec. 2001.

[4] HP Labs Semantic Web Programme, “LARQ—Free Text Indexing for SPARQL,” http://jena.sourceforge.net/ARQ/lucene-arq.html/

[5] V. Hristidis, H. Hwang, and Y. Papakonstantinou, “Authority-based keyword search in databases,” ACM Trans. Database Syst., vol.33, no.1, Article 1, pp.1–40, 2008.

[6] K. Goldenberg, B. Kimelfeld, and Y. Sagiv, “Keyword proximity search in complex data graphs,” Proc. SIGMOD, pp.927–940, 2008.

[7] F. Crestani, “Application of spreading activation techniques in information retrieval,” Artif. Intell. Rev., vol.11, no.6, pp.453–482, 1997.

[8] X. Ning, H. Jin, and H. Wu, “RSS: A framework enabling ranked search on the semantic web,” Inf. Process. Manage., vol.44, no.2, pp.893–909, 2008.

[9] C. Mangold, “A survey and classification of semantic search approaches,” International Journal of Metadata, Semantics and Ontology, vol.2, no.1, pp.23–34, 2007.

[10] A. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, and C. Wroe, “Practical experience of teaching OWL-DL: Common errors and common patterns,” Proc. Engineering Knowledge in the Age of the Semantic Web (EKAW), pp.63–81, 2004.

[11] J. Kamps, J. Pehcevski, G. Kazai, M. Lalmas, and S. Robertson, “INEX 2007 evaluation measures,” INEX 2007, LNCS 4862, pp.24–33, 2008.