SWARMS: A Tool for Exploring Domain Knowledge on Semantic Web

Liang Bangyong, Tang Jie, Wu Gang, Zhang Peng, Zhang Kuo, Xu Hui, Zhang Po, Yan Xuedong, Li Juanzi

Knowledge Engineering Group, Department of Computer Science, Tsinghua University
{liangby97, j-tang02}@mails.tsinghua.edu.cn

Abstract

This paper introduces SWARMS, a tool for exploring domain knowledge in semantic web. By domain knowledge exploration, we mean searching for or navigating the knowledge in a specific domain. We have found, through an analysis of survey result and an analysis of using log data, that requirements for domain knowledge exploration can be grouped into three categories. The categories include knowledge search, schema based navigation, and search results analysis. Traditional methods usually focus on one of the three types, for example, retrieval of 'relevant knowledge' by exploiting full-text retrieval methods. We propose a tool, called SWARMS, for exploring domain knowledge, in which we provide the ability to conduct domain knowledge exploration by the three categories. Specifically, users can conduct search for special kind of knowledge and they can also interact with the tool by navigating the knowledge base. Furthermore, we conduct analysis for the search or navigation results. The tool is applied to the software management domain. We use ontology as the mean for knowledge representation. The paper describes the architecture, features, and component technologies of the tool.

1. Introduction

Domain knowledge management has made significant progress in recent years, particularly after the emergence of Semantic Web. Many knowledge bases are constructed for managing domain knowledge [AMO03]. However, domain knowledge management does not seem to be so successful. One of the most challenges for domain knowledge management is the exploration of domain knowledge.

Several systems have been developed for domain knowledge exploration [NSD01]. However, most of them look on domain knowledge exploration as a problem of either conventional relevance search or knowledge browsing. In relevance search, when users type a query, the system returns a list of ranked 'targets' with the most relevant 'target' on the top. Here, the target can be document or object in the knowledge base. In knowledge navigation, users select the concept that they want to browse and input some specific constraints from the knowledge schema, and the system returns the 'targets' that belong to the concept and satisfy the constraints. Navigation also enables users to navigate to the objects that are 'similar' to the current browsing object.

In this paper, we try to address the domain knowledge exploration in a novel approach. We categorize the requirements for domain knowledge exploration into three categories, i.e. knowledge search, schema based navigation, and search results analysis.

Our proposal first is to take a strategy of divide-and-conquer, and then is to combine them into a unified system. Users can start their exploration on the knowledge base by typing a keywords-based query. The system returns the relevant objects. And then users select what they want to browse. The object is shown in a navigation view, in which users can browse its schema information, its value, and those objects related to it. In this view, users can navigate to other objects related by the help of a graphic user interface. Users can also specify some constraint and search directly in the navigation view. Finally, for the search results, we provide two kinds of analysis on it by using text mining technologies. The former is similarity analysis and the later is knowledge summary. In the paper, we refer to the approach as 'unified domain knowledge exploration'. The advantage of unified domain knowledge exploration lies in that it can accommodate the knowledge search, navigation, and knowledge analysis well. Furthermore, analysis helps users understand the knowledge easier. It is reasonable particularly in domain knowledge management, because in a domain knowledge is usually represented by a knowledge language (e.g. Web Ontology Language OWL) which makes it difficult for users to understand. Knowledge summary aims to represent knowledge by understandable natural language to users. Similarity analysis helps users to locate the similar objects to what they have obtained or to compare the objects in the knowledge base. We have developed a system based on the approach, which is called SWARMS.

The rest of the paper is organized as follows. In section 2, we introduce related works. In section 3, we explain our approach to the problem. In section 4, we describe the main viewpoints of SWARMS to end users and we introduce the architecture and implementation of SWARMS in section 5. Finally the conclusions are made in section 6.

2. Related Works

Knowledge search can be seen as one part of knowledge management. Knowledge search is concerned with finding the 'relevant' knowledge from knowledge base. For

example, Swoogle uses the techniques from information retrieval to build a search center of semantic web resources [DFJ04]. The search results by Swoogle can be ontology file, concepts, properties and instances. The results are not easily understandable for average user. Semantic Search project extends the keyword based search [GMM03]. It can find the instances that do no contain the keywords in the query. The project aims to enhance the traditional search by the semantic search techniques.

Knowledge navigation aims at 'focus+context' navigation in knowledge exploration. The focus means the object that satisfies current criteria (usually specified by user) and the context means the related objects to the current target. For example, Janecek and Pu propose an interactive visualization technique for exploring an annotated image collection [JP03]. The focus and context are considered and the search results provide both of them. Flink(http://prauw.cs.vu.nl:8080/flink/) gives a graphical view of researcher social network. For a researcher, the view displays his interest fields and researchers that have the same interests with him.

3. Our Approach to Domain Knowledge Exploration

The underlying data models in SWARMS are ontology. Domain knowledge base stores the information organized according to the domain knowledge schema predefined by domain experts. Different from traditional search in which the 'target' is only document and the corresponding search task is the retrieval of relevant documents, domain knowledge can have complicated schema. For example, in the software domain we have defined, there are 19 concepts, 109 properties and 2925 instances in total.

The knowledge schema can help users to organize their data well. It presents explicit semantics for the data, which makes it possible for more advanced applications such as reasoning. On the other hand, it has higher requirement for the knowledge exploration. Question Answering is an ideal form for knowledge access. When users type a natural language question or a query (a combination of keywords) as a description of his search criteria, it is ideal to have the machine 'understand' the input and return only the necessary information based on the request. However, there are still lots of research work to do before putting QA into practical uses. In short term, we need consider adopting a different approach.

We have found that we can group the users' needs into three categories. Specifically, when users don't know the knowledge schema or other domain knowledge, they can launch a search process by only typing several keywords. And the system returns all concepts/properties/instances that contains the keywords. Secondly, when users have specific object that they want to search, they can specify the concepts/properties/instances in the knowledge navigation view. They can specify more constraints before conduct the search. Finally, since data in knowledge base is represented by triples, general users without enough domain knowledge may have difficulty to understand it. We propose analysis technique to deal with the problem. We make use of two methods for analysis, i.e. knowledge summary and similarity analysis.

4. SWARMS

Features

Currently, SWARMS provides three types of exploration. 1) Knowledge Search. It searches the concepts, properties and instances in the knowledge base by making use of full-text search technology. 2) Knowledge Navigation. It provides three kinds of navigation, i.e. concept navigation, instance navigation and eagle eye navigation. 3) Search Result Analysis. It summarizes the knowledge into natural language. A text in natural language describing the meaning or the content of the concepts or instances that users select is returned. Users can also use it to find the similar concepts/instances to what they are interested in.

Ontology Definition and Knowledge Base Construction

We define a software ontology[1] by referencing the schema on SourceForge (http://ww.sourceforge.net), one of the biggest open source software development websites.

We have developed a rule-based wrapper to get the data from SourceForge and store them into the knowledge base according to the ontology.

Search View

There are four types of searches in Search View: full-text search (also called 'document' search), Instances Search, Classes Search, and Properties Search. In document search, users type the keywords, and the system returns a list of ranked entities. The entity can be concept, instance, or property. Each entity is assigned a score representing its relevance to the input keywords. We assign the scores using information retrieval model. The returned entities are grouped into concepts, instances, and properties respectively.

As model, we employ VSM (Vector Space Model) [SWY75], which computes the Cosine Similarity between the input keywords and entities in knowledge base. For computing the Cosine Similarity, we need to construct a document for each entity. We extract bag of words for a concept from its name and properties that are related to it and view the bag of words as the document for the concept. For properties, we further divide it into *object* properties and *datatype* properties. Document for both of the properties are defined by words in its name only. For instances, we only consider concept instances. We do not take into consideration of property instances. There are

[1] The ontology is available at http://www.schemaweb.info/schema/SchemaDetails.aspx?id=235

two reasons: almost all property instances are related to one or more concept instances and a preliminary survey indicates that usually user prefers concept instances to property instances. Score of each entity ranges from 0 to 1, where 0 indicates non-relevance and 1 indicates exact match.

In search, given a query, all entities matched against the query keywords are retrieved and presented in descending order of the relevant scores.

Figure 1 shows an example of instance search. There are two tab views: Text Search view and Visual Search view. Here as the search view, we mean the Text Search view, which is the default view in SWARMS. There are four radio buttons corresponding to the four types of searches. The check box "Summary" indicates knowledge summary (we will describe it in detail below). The left window displays the retrieved instances and the right window displays the detailed information for the selected instance. Detailed information of instance includes its name, value (e.g. string or numeric) of related *datatype* property, and value (i.e. another concept instance) of related *object* property. The bottom window is retained for knowledge summary.

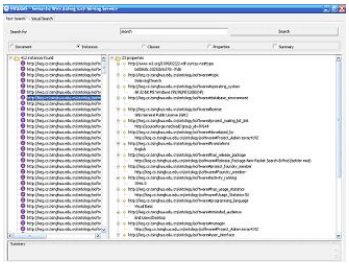

Figure 1. An example of instance search

Navigation View

There are two means to enter the Navigation View: users can double click the entity name in the Search View and users can directly switch to Navigation View by clicking the Navigation View tab.

When users directly switch to Navigation View, the system displays a graph with the concept "Project" in the middle of the view (we think the concept "Project" is a more important concept in software management) and concepts that related to it (as shown in figure 2). In the graph, round node denotes concept, directed edge denotes object property. Users may have different preferences to the concept for navigation. We provide a drill mode for facilitating the navigation. When users are interested in one of the concept, they can double click the round node denoting the concept. A new graph will be rendered which displays the clicked concept in the middle of the graph and surrounds it with concepts that related to it. We have tried displaying all the concepts and relations in the graph, but it results into a very complicated graph that is full of nodes and edges.

Figure 2 shows an example in concept navigation. The main window displays the concept graph, and the top-right window displays properties of the selected concept. The bottom right window is the eagle-eye window. Users can go to any part of the navigation view by selecting the zone in the eagle-eye window.

Figure 3 shows a concept navigation scenario. Users double-click the concept "Project_Admin" or "LatestNew", and then the system returns the corresponding graph that places them in the middle.

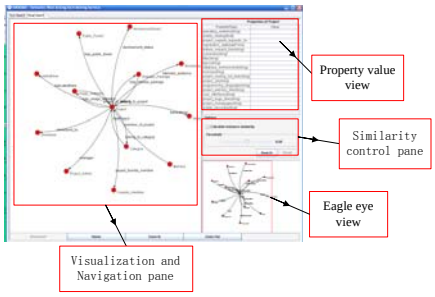

Figure 2. A concept navigation example

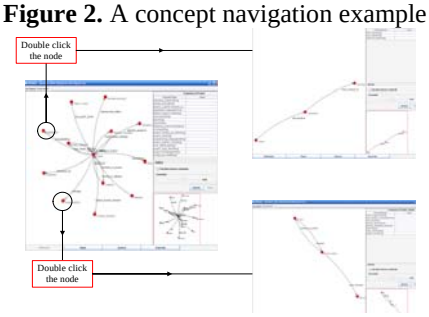

Figure 3. A concept navigation scenario

We also tried to combine knowledge search and navigation into a unified mode. We called it navigation based search. In navigation based search, when users click a concept in the concept navigation view, the top right window list its properties with none values. Then users can input some property values and conduct search by these constraints directly in the navigation view. For example, users may be interested in the projects which are developed by Java language. He can input "Java" in the datatype property "Programming_language", and clicks the "search" button to perform the search. Figure 4 shows the example.

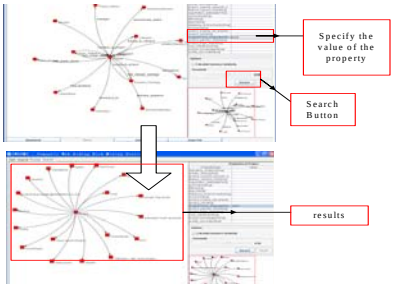

Figure 4. An example of navigation based search

Search Result Analysis

Similarity Analysis

We exploit VSM for computing the similarity between two instances. For instance, we construct the document as we did in the sub-section "Search View". We extract the bag

of words from the 'document' and compute the similarity between two documents by Cosine Similarity method. In similarity analysis, we compute similarity score for every pair of instances of a concept and display the similarity in the graph as shown in figure 5. A similarity threshold slider is placed in the middle of the right window. With the threshold slider, users can control the number of similarity links that displayed in the graph.

Knowledge Summary

Here we conduct the knowledge summary in the interface as an optional function. When search results are displayed, users can select a result and check the "Summary" checkbox. The summary of the entity will be displayed in the summary pane. Figure 5 shows an example summary. The bottom window displays the summary result

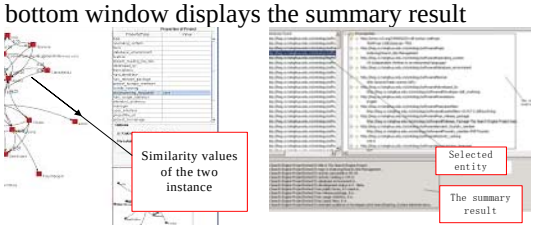

Figure 5. Similarity Analysis and Knowledge Summary

5. Architecture and Implementation

There are six main components in SWARMS: Knowledge Extractor, Domain Knowledge Base, Indexing, Knowledge Search, Navigation, and Search Results Analysis modules.

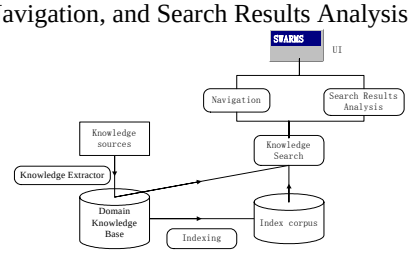

Figure 6. Architecture

We chose SourceForge (http://ww.sourceforge.net) as the knowledge data source. Totally, 1180 software projects are crawled into the knowledge base.

In Indexing module, we derive the ideas from the community of Information Retrieval and build an inverted table indexing. In the inverted table, besides indexing the entities, we also index properties that related to the entities. Indexing for concepts and instances are built independently. For knowledge exploration, we have implemented two kinds of search mechanisms. The first search mechanism makes use of the inverted table indexing. It is aimed for full-text search. The other mechanism is implemented by RDQL(RDF Data Query Language)[Sea03]. It is designed for complicated query. It is appropriate to allow for both high efficiency and advanced search functions.

The Knowledge Search makes use of inverted table indexing. The principle of obtaining the search list and ranking it are described in prior sections. In Navigation, we use both inverted table indexing and RDQL. For navigation based search, we use only RDQL, since the query can be very complicated. The graph visualization in navigation is implemented by JUNG(http://jung.sourceforge.net).

Both similarity analysis and knowledge summary have great computational costs. So they are processed in advance. When new instances come to the knowledge base, the analysis module is called to incrementally calculate the similarity scores among instances and conduct the summary for the new instances. We only calculate the similarity score between any two instances that belong to the same concept

Finally, we provide two kinds of versions: standalone application and web version. They are both available at http://keg.cs.tsinghua.edu.cn/project/pswmp.htm.

6. Conclusion

In this paper, we have investigated the problem of domain knowledge exploration. We have made clear the following issues in the work. 1) Through an analysis, we have found that exploration needs on domain knowledge can be categorized into three types. 2) Based on the finding, we propose a new approach to domain knowledge exploration in which we combine the search, navigation, and search result analysis into a unified method. 3) We have developed a system called 'SWARMS', based on the idea. In SWARMS, we provide features for knowledge search, knowledge navigation, and search result analysis.

References

[AMO03]Angele, J., Mönch, E., Oppermann, H., Staab, S., and Wenke, D. Ontology-Based Query and Answering in Chemistry: OntoNova @ Project Halo. International Semantic Web Conference 2003: 913-928

[DFJ04]Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V.C., and Sachs, J. Swoogle: A Search and Metadata Engine for the Semantic Web. In Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management, November 2004 : 652-659

[GMM03] Guha, R., McCool, R., and Miller, E. Semantic Search. In International World Wide Web Conference Proceedings of the twelfth international conference on World Wide Web. ACM Press. Budapest, Hungary. 2003:700-709

[JP03] Janecek, P. and Pu, P. Searching with Semantics: An Interactive Visualization Technique for Exploring an Annotated Image Collection. OTM Workshops 2003: 185-196

[NSD01] Noy, N.F., Sintek, M., Decker, S., Crubzy, M., Fergerson, R.W., Musen, M., Creating Semantic Web Contents with Protege-2000. IEEE Intelligent Systems 48(2): 60-71, 2001.

[Sea03]Seaborne, A. RDQL-A query language for RDF. http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/, 2003.

[SWY75]Salton, G., Wong, A. and Yang, C. S. A Vector Space Model for Automatic Indexing. Commun. ACM 18(11): 613-620 (1975)