Protégé 2004 Conference: Presentation Proposal
Title: Ontology-Aided Web Search Assistant

Authors: Ann Lee, Edward Powers
Advisor: Dr. Magdi Kamel, Naval Postgraduate School, Monterey, CA

Background – Web Search Difficulties

When we perform a web search on a word or a phrase using search portals such as Google and Yahoo, what comes back as the result are hundreds, even thousands, of pages of web sites. Often there is so much information, it is difficult to go through all of them. Currently, many search engines use a method of indexing based on words contained in a web page or number of links to pages containing the key words. Therefore, the search engines are based on a brute force approach that prioritizes the results according to the simple ranking method. While the sheer speed of the engines provides the results in a matter of seconds, the relevance is another matter.

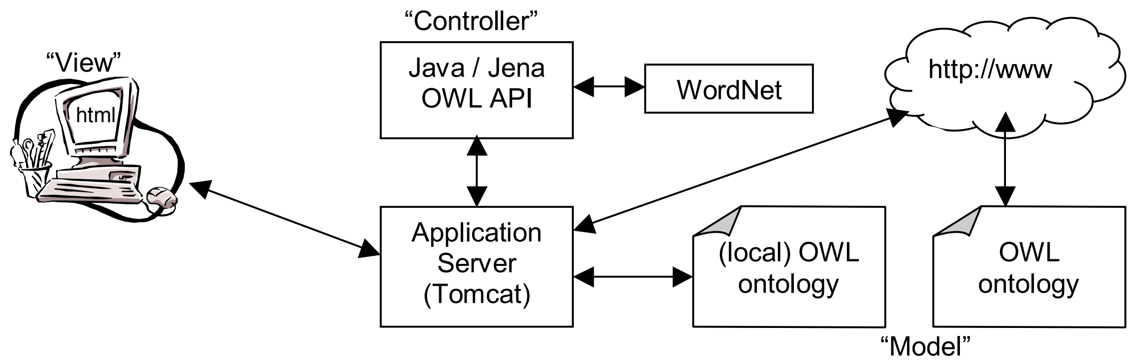
We can contrast this method of searching to that of a library. When we go to at a library for a specific information, it is likely that we will walk into a place with thousands of books on the shelves. Rather than looking for the information book by book, shelf by shelf, we will go to the catalog and search on the metadata. That is, the library will have an index system that contains information about every single book it houses. For example, most library catalogs will have data on the author, title, published date, and a short description about the book. The catalog, however, will not indexed based on "all books containing the word 'ontology,'" as in the web indexing system, because that is not an effective method of cataloging any source of information.

Although extensive metadata about the content of web pages do not exist, as Tim Berners-Lee envisions for the "Semantic Web," there are other methods to enhance the effectiveness of our searches. This project proposes to use ontologies to add context or narrow the focus to achieve a more meaningful set of results. An effective search is defined by getting a set of highly relevant results and eliminating the useless overload of information. However, achieving effectiveness is more difficult than it seems. One way is to use a list of applicable keywords to narrow the search. Properly crafting a list of relevant keywords requires a level of expertise in that particular domain of knowledge. This is where ontologies play a crucial role. An ontology can be thought of as a Knowledge Base (KB) that describes a concept and all the relationships that it can have with other related concept. In other words, the ontology fills the role of a domain expert. By building an application that uses the ontologies as a layer between the user and the web search engine, the user is provided with assistance in building a list of search terms.

The goal of this project is to build an application that can interface between the user and the domain expert, the ontologies, to improve the effectiveness or relevance of the web searches.

Project Implementation

The architecture of this application can be organized as a Model-View-Controller paradigm. The "Model" or data source is an ontology KB written in OWL. The foundation of this application is to have access to ontologies that are written in OWL DL, which represent a class type hierarchy for a given domain. For the purposes of this project, we assume that the user has access to a myriad of useful ontologies. The "View" of the query result will be rendered in HTML. And, the "Controller" will consist of Java programs using the Jena APIs and WordNet. Jena is the Java toolkit for developing semantic web applications. Jena can parse and manipulate OWL and is used as the primary method of access for query to the back-end OWL KBs. Communication between components is organized via an application server and HTTP.



The user experience is much like using the Web Search portal. The following use case describes how this would work.

	Actor	System
1	The user wants to look up a term that she overheard in a conversation. She has some vague ideas about the meaning of the term but is otherwise unsure. The user decides to research the term on the web. The term is "SLR". The user invokes the "Search Assistant" program.	An HTML based screen appears with a text entry field and submit button. Below that is a pick list allowing the selection of ontology KBs.
2	The user picks from a list of ontology KBs she wants to use and enters "SLR" in the text entry field and presses the submit button.	a) System begins searching for "SLR" in the KBs. b) If the term is found in a given ontology, it searches for other terms related to the original. All found words are added to a list. c) The System displays another screen with the expanded list of search terms. Each list of terms is organized by the originating ontology. Next to each term is a radio button which allows the users to make a selection. All the selected terms are used in the web search. Also, a term is displayed as hyperlink when there is an associated WordNet definition. There is also a text entry field if the user wants to manually enter their own search terms.
3	The user reads and selects the terms she wishes to include in her search and hits the submit button.	The system submits an HTTP request to the web search portal (such as Google) using the selected terms. The user is directed to the search engines results page.

Mining OWL

The "Controller" part of the application queries the ontologies for a given term using Jena API calls. Once the term is found, it becomes the starting node of the tree structure from which we can begin searching for other related information. Assuming that our beginning node is a property of some class, we would access related nodes closest to the starting node, which will represent the strongest relationship to the original term. Examples of this are sibling properties, parent classes and siblings of parent classes. In addition, OWL provides semantic constructs that define relationships between classes and instances that indicate close relationships, such as "sameAs," "inverseOf," and "complementOf" to name a few. These constructs allow an application to further define the correlation between all the classes, properties and instances within a given ontology. In other words, we will navigate the OWL tree to extract the related information of the original term.

Post Processing of OWL Query Data

The information extracted from the OWL ontology KBs may or may not be useful to the user. Given that the OWL classes are named by authors of the ontologies, there could be misspellings or labels with concatenated words. It is important to perform post-processing of the results for the users to decide if the information is defined in a meaningful way. Also, the user may not be familiar with the terms that are returned from the query. It would be helpful if the user is presented with definitions for each term. This allows the user to get even more knowledge about the domain of their search. To achieve this, the labeled nodes will separate any word concatenations. Next, all words will be processed through the WordNet application. WordNet is 3rd party lexical database used (in our case) to provide definition or synonyms to our search term list. This is intended to be useful to the user in deciphering the knowledge domain of the search term that came out of the ontology. Words not found in WordNet are not hyperlinked.

Conclusions

The goal of this application is to leverage ontologies in order to provide a user with an effective tool for research. The main advantage of using the application is to assist the user in discovering important information about a knowledge domain. Successful use of ontologies for this application also provides a side benefit -- it encourages us to create more ontologies and help move us toward the "critical mass" of meta-data necessary to have a truly Semantic Web.