A Conceptual Architecture for Semantic Search Engine

Qazi Mudassar Ilyas, Yang Zong Kai and Muhammad Adeel Talib
Department of Electronics and Information Engineering Huazhong University of Science and Technology, Wuhan – 430074, P.R. China
qmudassir@yahoo.com, zkyang@public.wh.hb.cn, adeeltalib@hotmail.com

Abstract

The future search engines on World Wide Web, the semantic search engines, will not be based merely on a word index. They will be able to understand the meaning of the content on the Web pages and carry out logical reasoning on them to perform complex search queries and return accurate results. In this paper we propose a conceptual architecture for a semantic search engine. We have discussed the components required by the engine, the requirements of these components, and their interaction with each other. Ontology editor, ontology mapper, ontology translator, Web page annotator, ontology crawler, Web crawler, query builder, knowledge base, and inference engine are the basic components of the semantic search engine. We particularly focus on inference engine and propose the use of relational database to store the knowledge base and the ontologies to eliminate the shortcomings of the current inference engines.

Keywords: Semantic Web, knowledge representation, ontology, inference engine, Web technologies, metadata, semantic search engine.

1. Introduction

World Wide Web has grown haphazardly with a great pace since its creation. Due to the huge size of the Web, it has become difficult for the users to find their required information. Search engines appeared on the scene to help the users find their required information. They maintain a word index that is used to accomplish search. Although the search engines are very helpful in finding information on the Internet and are getting smarter with the passage of time, they suffer from the fact that they do not know the meanings of the terms and expressions used in the Web pages and the relationships between them. The problem is intensified due to the presence of the polysemy (one word having several meanings) and synonymy (several words having same meaning) in natural languages. Hence search engines are handicapped by being unable to figure out the context in which a word is being used.

Thus when we give a search query like “pipe +“Computer Science”” to find the definition of “pipe” in Computer Science domain, the most accredited search engine, Google, is unable to find the right document (no document is relevant among the top ten results returned). This is because Google does not know which pipe we are talking about; a pipe of any kind, a device for smoking, a musical instrument or a portion of memory that can be used by one process to pass information along to another. It was possible for Google to find the right document only if it knew the relationship between the two terms given to it; pipe and “Computer Science”.

This was just the example of a simple search query. These queries can sometime be improved by choosing different expressions for the search string. Now let us consider a complex case. Assume someone wants to find the names of all Chinese pedagogues who have written documents on Semantic Web during the last year. This query is simply out of question for the present search engines.

Semantic Web [6] makes it possible to successfully execute the query given above and other much more complex queries. Semantic Web depends on the ability to associate formal meaning with content. Ontologies – formal vocabularies to define the worlds by declaring the concepts and their relation in a domain of interest in an unambiguous way – are the building blocks of Semantic Web. The focus of Semantic Web is to make the Web machine-understandable where automated agents will be able to understand the content on the Web, establish relationships between them and take logical decisions to accomplish complex tasks with minimum human intervention.
2. Semantic search engine

After carefully analyzing the requirements, we propose a conceptual architecture for semantic search engine (Figure 1). A typical semantic search engine should consist of the following components:

2.1. Ontology development

Ontology is defined as a formal, explicit specification of a conceptualization [7]. Knowledge sharing and machine-understandability are the two main objectives of ontologies. Thus, if someone performs analysis and reaches at a satisfactory set of conceptualizations and their representative terms for some area of knowledge and finally represents this knowledge in the form of ontology in any standard ontology language, then this knowledge can be reused by other people in that domain thus eliminating the need of replicating the whole process.

W3C is efficiently working on developing languages and tools for creating ontologies. RDF [8] and RDF Schema have been declared recommendations. OWL (Web ontology Language) [2] was recently added in the list of recommendations. OWL is very close to DAML+OIL [1] and is very expressive. It has three increasingly expressive sub-languages – OWL Lite, OWL DL, and OWL Full. It provides advanced features like data types, expression for enumerations, properties of properties, disjoint classes, and cardinality constraints.

Our proposed system uses ontologies created in DAML or OWL. We recommend OWL DL for creating new ontologies because it is expressive enough to represent complex concepts and at the same time guarantees decidability and computational completeness in finite time. The ontologies are first created in an ontology editing tool. A number of proprietary and open source tools have been created for creating ontologies in the recent years. Some of them are listed in Table 1.

Table 1. Ontology editing tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Product or Project Web site</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAML-Edit</td>
<td><a href="http://www.w3.org/standards/semanticweb/ontology">http://www.w3.org/standards/semanticweb/ontology</a></td>
</tr>
<tr>
<td>Epsilon</td>
<td><a href="http://www.cs.soton.ac.uk/ontology/ontology">http://www.cs.soton.ac.uk/ontology/ontology</a></td>
</tr>
<tr>
<td>Ontology Editor (OE)</td>
<td><a href="http://www.w3.org/standards/semanticweb/ontology">http://www.w3.org/standards/semanticweb/ontology</a></td>
</tr>
<tr>
<td>OntoEdit</td>
<td><a href="http://www.csc.ncsu.edu/ontology/ontology">http://www.csc.ncsu.edu/ontology/ontology</a></td>
</tr>
<tr>
<td>Danish</td>
<td><a href="http://www.cs.cmu.edu/ontology/ontology">http://www.cs.cmu.edu/ontology/ontology</a></td>
</tr>
<tr>
<td>OntoHead</td>
<td><a href="http://ontowise.dcc.fc.up.pt">http://ontowise.dcc.fc.up.pt</a></td>
</tr>
<tr>
<td>OntoLingo</td>
<td><a href="http://www.ontology.org">http://www.ontology.org</a></td>
</tr>
<tr>
<td>OntoBuilder</td>
<td><a href="http://www.ontology.org">http://www.ontology.org</a></td>
</tr>
<tr>
<td>OntoPolicy</td>
<td><a href="http://www.ontology.org">http://www.ontology.org</a></td>
</tr>
<tr>
<td>OntoReproducible</td>
<td><a href="http://www.cs.rice.edu/ontology/ontology">http://www.cs.rice.edu/ontology/ontology</a></td>
</tr>
<tr>
<td>OntoExpression (OEDE)</td>
<td><a href="http://www.w3.org/standards/semanticweb/ontology">http://www.w3.org/standards/semanticweb/ontology</a></td>
</tr>
</tbody>
</table>

2.2 Ontology Crawler

An ontology crawler can crawl through the Web to find new ontologies and populate the database with them. Note that the ontologies crawled by the ontology crawler will not be directly dumped into the database. Ontology translator, with the help of ontology mapper, will translate them into the database tables. These ontologies will be used for performing semantic search. In the sphere of current Web, keyword index is the main criterion for search engines. A search engine that has crawled the Web more and has larger word index is considered to be better. Similarly, in the sphere of knowledge-based Web of tomorrow, a semantic search engine having larger collection of ontologies will be considered better because it has more “knowledge” of the world. Size of knowledge base, discussed in section 2.4, will be the second major yardstick to measure the quality of the search engines.

In a real system, the users will be using different languages (DAML, OWL, etc.) for creating ontologies. The ontology crawler should be able to handle ontologies written in different languages.
2.3. Ontology Annotator

Once the ontologies have been created, we have to annotate the Web pages with this metadata. Ontology annotator should be able to read the ontologies from the database or plain text files and allow the users to annotate their Web pages with this knowledge. The process is very simple when we are creating new Web pages but when it comes to annotating the content already there on the Web, it becomes very challenging because the Web is read only and we do not have write access to the Web pages created by others. The only possible solution is to motivate the owners of the Web sites to annotate their content. For prototype systems to evaluate different algorithms and techniques, one possible solution is to duplicate the Web pages related to the selected domain locally and test the system. The prototype system can perform reasoning on this knowledge and perform semantic search. Once the search is performed, the users can be directed to the original locations on the Internet. Finally, tools can be developed for annotating the content automatically with the meta data. These tools are limited only to the
domains where all the Web pages follow a common pattern.

2.4. Web crawler

The purpose of Web crawler is to look for the annotated Web pages. Just like the traditional search engines look for key words in Web pages and build a word index, a Semantic Web crawler finds the concepts in the annotated Web pages and builds a knowledge base. The authors propose that the knowledge base should be generated in a relational database management system. Relational databases provide high performance, allow dynamic changes to the database and impose no limit on the number of ontologies.

Performance is the main point of concern in a multi-user environment where thousands of users will be accessing the database concurrently. Any system that does not allow multi-user access is doomed to failure in such an environment. World Wide Web is dynamic. One should expect addition, editing, and deletion of ontologies on a regular basis. The knowledge base should be able to cope with this situation allowing dynamic changes to the database to avoid any interruption to the users and run the system smoothly.

Finally partitioning of knowledge base should be possible. Some of the knowledge bases used in the projects, like Parka in SHOE (discussed in section 2.7), do not allow the partitioning of the knowledge base thus limiting the system to only one ontology. A real semantic search engine will need to have thousands of ontologies to perform efficient search. We, therefore, recommend both the ontology database and knowledge base to be created in relational database management system to allow multiple ontologies to be used and gain maximum efficiency in a multi-user environment.

2.5. Performing semantic search

Until now we have been discussing the server side of the semantic search engine. Now that all the pieces are at their respective places and the stage is set to perform semantic search, we come to the client side and discuss how the user can perform semantic search. One more component on server side - the heart of the system - inference engines, however, is yet to be discussed.

2.6. Query builder

Users are not supposed to have the knowledge of Semantic Web languages or ontologies to carry out semantic search. Thus we need to provide intuitive tools to the users to construct search queries. These tools should be able to provide the powerful features of semantic search without exposing the underlying complexities.

The query builder should be able to load the ontologies from the database and allow the users to build powerful search queries. The users should be presented with a list of available ontologies to choose the exact context in which to carry out search. It will eradicate the traditional problem of ambiguity with which present search engines are plagued. The user will be able to accurately express the context in which a term is being used and the search engine will no more be confused about other connotations of these terms.

2.7. Query pre-processor

The purpose of the query pre-processor is to convert the queries into a form understood by the inference engine. A sophisticated query builder should also provide lexical checks to remove spelling mistakes and give suggestions to the user to improve search query.

If the inference engine returns no results for the given query then the query is sent to traditional search engines. Query pre-processor can also be handy in this situation and can improve the search query by adding “+” to mandatory terms or enclosing the strings within quotation marks. The users usually ignore these simple syntax options that have a great impact on the results returned by the traditional search engines.

The finished queries are finally sent to the inference engine for performing semantic search.

3. Inference engine

This is the heart of the system. Its basic purpose is to infer new knowledge from the available knowledge of ontologies by performing logical reasoning on it. Ontologies provide the meanings of the words in a given context and the relationships between the terms used in the search query.

Consider the search query discussed earlier in section 1.
“Find the names of all Chinese pedagogues who have written documents on Semantic Web during the last year”.

The query mentioned above is not as simple as it looks. There are seven concepts involved in it namely name, Chinese, pedagogues, to write, document, Semantic Web, and year that have to be considered by the search engine. Assuming ontology database contains all the necessary information about these concepts, the inference engine will first parse the query to separate the concepts. The first thing it will come to know is that name is the property of a person and last_name, first_name, and full_name all qualify for the search. Second Chinese is a person who lives in China, which is the name of a country. Thus all persons who have the property inhabitant_of or resident_of equal to China qualify for the search. All kinds of faculty members (professor, associate_professor, assistant_professor, and lecturer etc.) qualify for pedagogue. Document can be a conference_paper, journal_paper, or book etc. To write is the relation between pedagogue and document. Semantic Web is a kind of web technology and year is a measure of time. The emphasized words (in italics) indicate the classes, properties, and relations in the ontologies. Person, faculty_member, and journal_paper represent classes while name, resident_of, and year_of_publication are their properties. Regarding relations the ontology will describe that a person is writer of a document. Once the inference engine knows about the meanings and relationships among all terms used in the query string, it is in a much better position to carry out the search accurately.

3.1. Related Work

Now we briefly discuss some inference engines and logical reasoners developed by others and provide a critical analysis of them. Then we discuss the model proposed by us.

Two of the main inference engines are CWM (Closed World Machine) [4] and Euler [5]. The former was developed by Tim Berners Lee and Dan Connolly and the latter by Jos De Roo. Both of them have been developed for experimental purpose and lack performance capabilities. Their purpose is just to play/demonstrate code and thus cannot be deployed commercially on a large scale. CWM is implemented in Python and uses RDF. A CWM Clone in Prolog is under development by Bijan Parsia. Euler is implemented in Java enhanced with Euler path detection. Both, CWM and Euler, do not support knowledge base and can only tell you whether given set of facts and rules supports a given solution.

Another project worth mentioning is SHOE (Simple HTML Ontology Extension) developed by University of Maryland in 2001 [3]. It is based on ontologies created in SHOE—a language developed by the same university—and has been implemented in two domains, Computer Science and food safety. However, it suffers from the following drawbacks:

- SHOE is not a standard. W3C has worked hard on standardizing the Semantic Web languages. They are widely accepted by the community and are based on XML in contrast to SHOE, which is based on HTML.
- It is implemented in XSB and Parka. XSB is a single-user system that makes it completely unsuitable for use on Web.
- Parka knowledge base suffers from the fact that it cannot be partitioned that makes it suitable for the systems using only one ontology. This is practically impossible in a real semantic search engine that will need to have thousand of ontologies to perform reasonable search.

3.2. Proposed architecture

After discussing the existing inference engines, we describe our proposed architecture and state how it eliminates all the problems indicated in other engines. The problem with first two inference engines is that they do not support full-fledged knowledge base thus restricting the functionality to only code demonstration. This problem does not exist in our proposed system as we are providing a complete knowledge base. We suggest Prolog as the implementation language for the inference engine that is the language used for logical reasoning for years and has been optimized for best performance over the course of time. Thus the problem of efficiency is also resolved. Relational database is also expected to increase the efficiency of the system. Our proposed architecture also eliminates the drawbacks of SHOE. By using OWL, a recommendation by W3C, the language standardization problem is solved. By using relational database, the other two problems are also solved; RDBMSs are multi-user and do not impose any limitation on the number of ontologies used.

Finally we describe the process of carrying semantic search in our proposed system. Search queries are fed into inference engine from the query pre-processor. It consults the knowledge base for the definitions of concepts used in ontology database. Once it knows the exact definitions of the concepts and relations between different terms used in the search
query, it tries to find matches in the knowledge base by performing logical reasoning on the terms. The results are finally sent back to be viewed by the user. If inference engine is unable to find any result relating to the search query, the query is sent to the traditional search engines and the user is presented with the search result returned from them.

### 3.3. Implementation

Different components of the search engine have different requirements. The client side tools – Ontology annotator, query builder, and query pre-processor – must be built in Java to ensure platform independence. For server side components – ontology translator, ontology mapper, ontology crawler, and web crawler – there is freedom of choosing implementation language. Inference engine should be implemented in Prolog for the reasons stated above.

### 3.4. Scalability

The main issues to consider for scalability are annotation of Web pages, storing and retrieving ontologies, manipulating knowledge base and carrying out reasoning.

As discussed earlier, there is no way to annotate the Web pages created by others. The simplest possible way is that the users annotate the content on the Web owned by them. The users need two things for this annotation. First the ontologies should be available to make this annotation possible. Second intuitive tools are required to make this annotation possible. Hence it is the responsibility of research community to provide these necessary ingredients to the ordinary users.

For ontology manipulation and knowledge base, we recommend the use of relational database management system that scale well to the large sizes required by semantic search engine.

Finally, Prolog is proposed to carry out reasoning. Prolog is a very old language specifically designed for reasoning, has been extensively tested in various environments, and has proved successful to cope with different situations.

### 4. Conclusions

We have presented a complete conceptual architecture for a semantic search engine in this paper. We have discussed all the components required by a semantic search engine. Our main focus, however, is on the inference engine. We propose a model to overcome the drawbacks in the existing inference engines. It should be noted, however, that the real semantic search engines are still far from reality. For them to become a reality, we need to have all the content on the Web annotated with metadata. The good news is that we have infrastructure in place and we only need to motivate the users to annotate their content on the Web. As ontology creation is still a demanding task, the duty of research community is to create ontologies that can fulfill the requirements of different communities and to develop the tools that make the annotation of the Web pages as easy as the creation of Web pages.

### References


