Chapter 4

New Directions in Building Next Generation Web: The Glow of Seamless Efficiency

4.1 Introduction

It is generally accepted that search engines based on IR (keyword and phrase matching between the query and index) alone tend to offer high recall and low precision. The user is faced with too many results and many results are irrelevant. The main reason for this is the failure to handle polysemy (a word that has two or more similar meaning) and synonymy (a word that has the same meaning as another word). The use of ontologies and associated metadata can allow the user to more precisely express their queries, thus avoiding the problems above. Users can choose ontological concepts to define their query or select from a set of returned concepts following a search in order to refine their query. This can improve the accuracy of a search. The average search engine query length of the web in a survey was 2.2 words. Having submitted a query, the user is then presented with a ranked list of documents relevance to the query. Therefore it is suggested here, the future of the search engines lies in supporting more of the information and management process, as opposed to incremental and modest improvements to relevance ranking of documents. In this approach, software supports the process of actually reading and analyzing relevant documents, rather than merely listing them and
leaving the rest of the information analysis task to the user. Corporate knowledge workers need information defined by its meaning, not by text strings. They also need information relevant to their interest and to their current context. They need to find not just documents, but sections and information entities and even digests of information created from multiple documents. The generation of ontologies and the creation of metadata attributing information to them is the key to access knowledge domain specific entities.

- Current knowledge management systems have significant weaknesses.
  - *Searching information*
    - Existing keyword based searches can retrieve irrelevant information that includes certain terms in different meanings. They also miss information when different terms with the same meaning about the desired content are used.
    - On the other hand, exploitation of interrelationships between selected pieces of information (which can be facilitated by the use of ontologies) can put otherwise isolated information into a meaningful context. The implicit structures so revealed help users use and manage information more efficiently.
- Extracting information

  - Currently, human browsing and reading is required to extract relevant information from information sources.
  - The explicit representation of the semantics of data, accompanied with domain theories (i.e. ontologies) will enable a web that provides a qualitatively new level of service. It will weave together a large network of human knowledge and will complement it with machine processability.

4.2 Creating a Knowledge Base

Ontologies offer a way to cope with heterogeneous representations of web resources. The domain model implicit in an ontology can be taken as a unifying structure for giving information a common representation and semantics. A semantic repository of information spanning multiple heterogeneous information sources can be created.

Ontologies are simply another class of model, although rather more complex than normally used in machine. Many search engines today, when requested to search for a particular string, do not just return the appropriate documents, but also highlight the occurrences of the string within the document. Using co-reference resolution, a search for ‘George W Bush’ would also highlight locations in the
documents where ‘the President’, ‘Mr Bush’, occurred or might even highlight ‘he’ or ‘him’ where the pronoun referred to George Bush.

Figure 4.1 illustrates the five basic operations of information extraction, applied to a simple text.

Ryanair announced yesterday that it will make Shannon its next European base expanding its route network to 14 in an investment worth around 180m. The airline says it will deliver 1.3 million passengers in the first year of the agreement, rising to two million by the fifth year.

- entities: Ryanair, Shannon
- mentions: it = Ryanair, the airline = Ryanair, it = the airline
- descriptions: European base
- relations: Shannon base_of Ryanair
- events: investment (180m)

Fig 4.1: Extracting information

Along with the ontology, containing the classes and properties, we also have a ‘knowledge base’ which contains the instances and the specific property instantiations. Together the ontology and the knowledge base make up a semantic repository. Whether the two parts are stored separately, e.g. in two distinct relational databases, depends on the practicalities of the implementation. As a text is analysed and named entities identified, hyperlinks are established to the instances in the knowledge base. Figure 5.2 illustrates a piece of text and an associated semantic repository. The repository contains
both ontology (classes) and knowledge base (instances). Also shown are the linkages between text and knowledge base. How such a repository can be used to, for example, enhance the search experience, is illustrated in a subsequent section.

Fig 4.2: Linking text to knowledge base

Ontologies and Semantic Web were mainly designed to tell the users sit back and bask in the glow of that seamless efficiency. When the user inputs a query it is a short one which is not expanded, hence it is difficult to retrieve the appropriate information each time. At times even if the query is long enough it may not be able to retrieve the semantically correct information. When a query is given using just one or two words it is not possible to retrieve the exact semantics of what the user wants. If this is not possible then the precision and recall parameters fall to a great extent. Hence we will have to first expand the query trying to retrieve the meaning of what it actually means and then search for the required results.
Ontologies are defined as a representation of shared conceptualization of a particular domain, a major component of semantic web. For our design we have used the University Ontology. For building the Ontology, first we have to list down all the classes and subclasses. Then we have used object properties like istaughtby, teaches, takes to indicate the relationships between the various classes. For instance, ComputerScienceModule is linked to the Academicstaff using the isTaughtBy relationship. Similar conditions hold for the other classes.

4.3 Proposed Algorithm for Next Generation Web.

Initially we restrict the algorithm to accept a maximum of two words as a query from the user because of implementation complexity. Then it expands the query to make a meaningful one by adding properties, instances, classes etc from the ontology, depending on their relationship.

The algorithm consists of the following steps.

1. User enters a query with maximum of two words from the domain.

2. Expand the query using the properties, relationships, instances etc.

   a) If the entered query is a concept then expand it in terms of its properties, subclasses and instances For e.g. if the concept is Room, it can be expanded it terms of its properties hasRoom or hasRent or ofSize, in terms of its
instances like place or in terms of its subclasses like singleRoom, doubleRoom, DeluxeRoom, Suite etc. So the expanded query will be (Room) and (hasRoom or hasRent or ofSize) or (singleRoom or doubleRoom or deluxeRoom or Suite).

b) If the entered query is a Property then expand the query in terms of its domain and range. If the query is hasRoom then the expanded one will be (Room and hasRoom) or (hasRoom and Accomodation).

3. Create a meaningful query with the relations.

4. Create a sub query by removing all conjunctions, prepositions (Like and, or, some, exactly etc)

5. Prioritize the words in the subquery.

6. Search using a search engine for the documents containing the words in the query.

7. Rank the documents.

8. The document which contains the maximum occurrence of the words will have highest priority.

<table>
<thead>
<tr>
<th>Sl No</th>
<th>Type</th>
<th>Query</th>
<th>Expanded version of Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Class</td>
<td>Department</td>
<td>Department and (ComputerScienceModule or EconomicsModule or MathsModule) and (isTaughtby only AcademicStaff)</td>
</tr>
<tr>
<td>2</td>
<td>Property</td>
<td>Takes</td>
<td>UndergraduateStudent and takes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Instance</td>
<td>Student1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Student1 and UndergraduateStudent or (takes only 2 Things)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Class-Class</td>
<td>i. Module - GraduateStudent (when they don't map to the same superclass) ii. EconomicsModule - MathsModule (when they map to the same superclass)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GraduateStudent and takes and (Module or ComputerScienceModule or EconomicsModule or MathsModule) (EconomicsModule and MathsModule ) or isTaughtby only Academicstaff</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Class-Property</td>
<td>Module-istaughtBy</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Module or istaughtBy (Academicstaff or Prof1 or Prof2 or Prof3 or Prof4 or Prof5 or Prof6 or Prof7 or Prof8 or Prof9)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Class-Instance</td>
<td>AcademicStaff - Prof1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Academic Staff and taught by some Prof1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Property-Property</td>
<td>Teaches - taught_by</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Academic_Staff and teaches or taughtby some (ComputerScienceModule or EconomicsModule or MathsModule)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Property-Instance</td>
<td>teaches-Prof1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Prof1 and teaches and (ComputerScienceModule or EconomicsModule or MathsModule)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Instance-Instance</td>
<td>Student1 - Student2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UnderGraduateStudent and (Student1 or Student2) and takes exactly 2 things</td>
<td></td>
</tr>
</tbody>
</table>
Explanation of how the algorithm will work.

1) Class: When the query given is a class we expand the query in terms of its equivalent classes, properties, inferred subclasses.

2) Property: When the query given is a property we expand the query in terms of its domain and range.

3) Instance: When the query given is an instance we expand the query in terms of its super class, properties and other instances which are related to this.

4) Class-Class: When the query given is a class-class we first check if they have a common ancestor. If they do have then it is expanded it terms of their properties. If they have different super classes, then they are expanded in terms of their connecting properties. Those documents which contain nothing in common are left out.

5) Class-Property: Here the class should be either a domain or a range of that property. If it's a domain (range), then the query should be expanded in terms of range (domain) including all its inferred classes, equivalent classes.

6) Class-Instance: Here the instance could be a child of the same class or of a different class. If it is the child of the same class then they are expanded in terms of their properties and super classes, else they are expanded in terms of their super classes and their connecting properties.
7) Property-Property: whenever two properties are given, we first check the domain on which they are defined and expand them in terms of their domain and its subclasses.

8) Property-Instance: If the property and instance are related, then the query is expanded in terms of their relationship, else it becomes a keyword based search.

9) Instance-Instance: Here we look at the class through which they have evolved and expand using their connecting properties, the specific properties of each of the individuals and the class to which they belong.

### 4.4 Need for IE and IR in Next Generation Web

In natural language processing [NLP], Information Extraction (IE) is a type of information retrieval whose goal is to automatically extract structured information, i.e. categorized and contextually and semantically well defined data from a certain domain, from unstructured machine readable documents. IE should be able to get the latest information and club it with the existing knowledge base so that up to date information is available for customers. IE is a division of NLP, where the function of NLP is to analyze, understand and generates human languages which is in turn built on technologies like Artificial Intelligence.

For information to be readily available, the interface between the customer and the services must be flexible as users demands are dynamic and even the types of users keep varying. An IR component
will retrieve from the web all those pages which contain the topic we have searched. Now an IE component will again filter this data and extract only information that is required, i.e. an IR component will contain a lot of unrelated information too whereas as IE will display only the related data. Hence, the task of IR is to select the page and the task of IE is to select the relevant data. Fig.4.3 is used to demonstrate how the two agents work hand in hand. Here IR can be the Service Consumer while IE can be the service provider as it responds by giving the right information to the query. The information returned by the IE will be some form of formatted data like XML. In this architecture the user first gives a query to the WWW. The crawlers search the web through IR agents for all links which contain this topic. Then these links are given to the IE. The job of IE is to filter out all the unwanted data and retain only the needed data. If these agents are efficient enough to integrate the information, then they can display abundant amount of related information, thereby making itself an intelligent system. This architecture can take advantage of many SOA characteristics like reusability, integration etc.
IR is extracting information from some text that satisfies the information needed from a large collection like a grid usually stored on computers. The information on the computers may be structured data like a Boolean expression or unstructured data like another document we have more amount of unstructured data than structured data. In IR, we have a set of users who will put forward queries to retrieve a subset of the database. For e.g., the document should be about semantic web and the author should be Tim Berners Lee. In this example, the query is about a certain topic and the author should contain the specified value. IR components like web crawlers can be used to scour the web to find the relevant information. IE is required to convert the unstructured data into a structured format.
4.5 Experimentation and Results:

The algorithm was tested on University Ontology as shown in Fig 4.4, The query results is shown in fig 4.5, fig 4.6 and fig 4.7 shows the comparison between proposed semantic search and ordinary search. The queries are expanded as shown in Table 4.1. The precision and recall parameters was estimated manually. It was found that the average increase in search with the proposed next generation was 49% compared to normal search.

Precision and recall are set based measures. That is, they evaluate the quality of an unordered set of retrieved documents. To evaluate ranked lists, precision can be plotted against recall after each retrieved document as shown in the example below. To facilitate computing average performance over a set of topics each with a different number of relevant documents, individual topic precision values are interpolated to a set of standard recall levels.

Recall = A measure of the ability of a system to present all relevant items. Recall = no of relevant items retrieved/ no of relevant items in collection.

Precision: A measure of the ability of a system to present only relevant items. Precision = number of relevant items retrieved/total number of items retrieved.

Using the interpolation rule, the interpolated precision for all standard recall levels up to 0.5 is 1, the interpolated precision for recall levels is 0.6 and 0.7 is 0.75, and the interpolated precision for
recall levels 0.8 or greater is 0.27. The average increase in search is 49% compared to ordinary search.

The results are given below:

Fig 4.4: University Ontology

Fig 4.5: Results of the domain (University) search with one word user query
Fig 4.6: Results of the domain (University) search with two word user query

Fig 4.7: Comparison between Ordinary and Proposed Semantic Search

**4.6 Conclusion**

The proposed system is an effort to retrieve relevant documents in a domain using domain specific knowledge captured in the form of OWL ontology. Experimental results show that the system retrieves the documents which would have been missed and avoids retrieving documents which are irrelevant despite the presence of the keyword. It thus improves the precision and recall substantially.
Future Directions:

• When the query is too generic, the expanded query may be too lengthy. In such cases the keyword search applications which have limitations on the length of the query to search may not work here. So the appropriate interface has to be designed for the ontology guided search.

Publications: