CHAPTER 5

SEARCH ENGINE USING SEMANTIC CONCEPTS

In recent years, everybody is in thirst of getting information from the internet. Search engines are used to fulfill the need of them. Even though the conventional search engines are used to retrieve information, the people get irrelevant information also for their query. A few pitfalls in current search engines are information overload, lack of getting of accurate information and content that is not machine understandable. The new generation of the web, the so-called Semantic Web, appears as an optimistic technology for implementing the search engine.

If a user gives a query like “apple”, Conventional Search engine retrieves the information about Apple Company also (i.e) www.apple.co.in, apple ipods, iphones etc. But semantic search engine retrieves the document related to apple fruit and its properties, relations etc. It retrieves the documents which are related to the user query. It is proposed to use ontology concepts to implement search engine.
5.1 SEMANTIC SEARCH ENGINE

The semantic search engine system consists of Semantic Parser, Reasoner, Document Retriever and Document Ranker.

![Figure 5.1 General Architecture of Semantic Search Engine](image)

The general architecture of the proposed system is given in Figure 5.1. In Semantic parser, the user query is processed and the processed query is matched with the ontology structure in semantic reasoner module. Ontology is constructed for three different domains such as bird, flower and river. Document retriever module is constructed to retrieve the documents. Finally, documents are ranked in the ranker module. The performance of the system is measured using precision and recall methods.

In the proposed system, two types of documents are considered for analysis. They are HTML documents and XML documents. Among them, XML documents have taken less time for retrieving results than HTML documents. Also, XML concepts are used for the efficient semantic information retrieval.
5.2 ARCHITECTURE OF THE PROPOSED SYSTEM

The detailed architecture of the semantic search engine system is shown in Figure 5.2.

5.3 MODULES OF THE SEMANTIC SEARCH ENGINE

The proposed semantic search engine system is constructed using the following four modules.

1. Semantic Parser
2. Reasoner
3. Document Retriever
4. Document Ranker
5.4 SEMANTIC PARSER

The proposed Semantic Parsing system accepts the generic natural language queries and returns SQL queries and RDF triples as output.

5.4.1 Algorithm of Semantic Parser

1. Parser splits the user queries into tokens using string tokenizer algorithm. The split keywords are called tokens.

2. Stop word elimination algorithm has been developed to eliminate stop words.

3. Stemming algorithm is developed to find the root word of the keyword. If the keyword is “searching”, it will reduce the word as “search”.

4. Processed keywords are called targets. The combination of stop word elimination technique and stemming process are called modifiers.

5. Then, targets are given as input to the WordNet. WordNet is a dictionary which is used to find synonyms of the targets. In this regard, the proposed system is linked with the WordNet software.

6. Using the WordNet dictionary, the keywords are classified as either noun or verb or adjective. If the keyword is noun or verb, it is classified as subject. If it is adjective, it is classified as predicate.

7. Using the subject, predicate and object concept, RDF triples are formed. The RDF triples are in the format of “<subject predicate object?>”
5.5  MODULES OF SEMANTIC PARSER

The proposed semantic parser consists of the following modules:

1. Tokenizer
2. Stop Word Elimination
3. Stemming
4. WordNet
5. Generation of SQL
6. Formation of RDF triple

5.5.1  Tokenizer

String tokenizer is the process of break up of a line of text into tokens. A token is a categorized block of text consisting of indivisible characters. In the proposed system, the user query is split into tokens and the implementation results are shown in Figure 5.3 on the next page.

5.5.2  Stop Word Elimination

Stop words are very common words that appear in the documents with little meaning; they serve only a syntactic function but do not indicate subject matter. These stop words have two different impacts on the information retrieval process. They can affect the retrieval effectiveness because they have a very high frequency. The removal of stop words changes the document length and subsequently, affects the document weight. The removal of stop words can increase the efficiency of indexing process 30% to 50% of the tokens in large documents. In semantic parser module, the stop words are eliminated using stop word elimination algorithm. The implementation result is shown in Figure 5.3 on the next page.
5.5.3 Stemming

Stemming is a process which transforms a word into its root form. Removing suffixes by automatic means is an operation which is especially useful in the field of information retrieval. Terms with a common stem have similar meanings, for example, “CONNECT, CONNECTED, CONNECTING, CONNECTION, CONNECTIONS”. Frequently, the performance of an IR system will be improved if term groups such as this are conflated into a single term. This may be done by removal of the various suffixes -ED, -ING, -ION, -IONS to leave the single term CONNECT. In addition, the suffix stripping process will reduce the total number of terms in the information retrieval system, and hence reduce the size and complexity of the data in the system, which is always advantageous.
Figure 5.3. shows tokenizer module and stop word elimination. Enter the User query (What is semantic search engine) and click the “Parse” button. The user query is split and stop words are eliminated. Now, the number of keywords is 3 (semantic, search, engine).

5.5.4 WordNet

WordNet is a database comprising nouns, verbs, adverbs and adjectives in a lexical relational structure. These words are then grouped by synonyms called synsets that are easily used to find related words and usage. The result of these synsets can be navigated and explored using a web browser. This basic structure of WordNet allows it to be easily used and researched for computational linguistics and language processing for a variety of applications such as voice recognition software and dictation software. Two kinds of relations are used in WordNet, semantic and lexical. The lexical relationships hold between semantically related forms of words and the semantic relationships hold between related word definitions. JWordNet is a pure Java standalone object-oriented interface to the WordNet database of lexical relationships. It is intended for Java programmers who wish to write portable Java applications that use a local copy of the WordNet files. The WordNet provides noun, verb, adjective and adverb for a word or sentence.

In this research work, the synonyms of keywords are found from the WordNet. Also, noun, verb, adjective and adverb for a word is also found. The experimental results of interface to WordNet dictionary is shown in Figure 5.4.
5.5.4.1 Noun and Verb

A noun is a word used to name a person, animal, place, thing and abstract idea. A noun is a member of a large, open lexical category whose members can occur as the main word in the subject of a clause, the object of a verb, or the object of a preposition. Lexical categories are defined in terms of how their members combine with other kinds of expressions. The syntactic rules for nouns differ from language to language.

A verb is a word (part of speech) that usually denotes an action, an occurrence, or a state of being. Depending on the language, a verb may vary in form according to many factors, possibly including its tense, aspect, mood and voice.
Using WordNet, noun and verb in the processed user query are identified. If the keyword is noun, it is considered as subject and if it is verb, it is considered as predicate. This implementation result is shown in Figure 5.5.

Figure 5.5 Identification of Subject and Predicate

5.5.5 Formation of RDF Triple

Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) which is originally designed as a metadata data model. The RDF metadata model is based upon the idea of making statements about Web resources in the form of <subject-predicate-object> expressions, called triples in RDF terminology. The subject denotes the resource, and the predicate expresses the relationship between the subject and the object.
After identification of subject and predicate, RDF triples are formed.

The experimental result shows the formation of RDF triples which is shown in Figure 5.6. The RDF triples are given as input to the document retriever module.

![Figure 5.6 Formation of RDF Triples](image)

5.5.6 GENERATION OF SQL

The SQL query is also generated for analysis of results. The processed query is converted into SQL query. The following Figures shows the generation of SQL queries.
5.6 COMPARATIVE ANALYSIS

The proposed system is analysed using WordNet. It is connected with the WordNet dictionary. The user query word is matched with the dictionary. The meanings of the user query are retrieved from the dictionary. With the meanings of the query, the documents are retrieved from the database. Table 5.1 shows the comparison between WordNet based Search Engine and Non-WordNet based Search Engine. A graph is drawn using the data given in Table 5.1.

There are two measures used for evaluating the effectiveness of semantic search engine system. They are precision and recall. Precision is the ratio of total number of retrieved documents to the total number of documents.
in the database. Recall is the fraction of the documents that are relevant to the query that are successfully retrieved.

\[
\text{Precision} = \frac{\text{Total number of documents retrieved by a search engine}}{\text{Total no. of Documents}} \quad (5.1)
\]

\[
\text{Recall} = \frac{\text{Total number of relevant documents retrieved}}{\text{Total number of retrieved documents}} \quad (5.2)
\]

### Table 5.1 Comparison of Search Engines

<table>
<thead>
<tr>
<th>Types of Search Engine</th>
<th>Total number of Documents in the database</th>
<th>Total number of Retrieved Documents</th>
<th>Total number of relevant documents in retrieved documents</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-WordNet based Search Engine</td>
<td>100</td>
<td>52</td>
<td>35</td>
<td>52%</td>
<td>67.3%</td>
</tr>
<tr>
<td>WordNet based Search Engine (Proposed System)</td>
<td>100</td>
<td>80</td>
<td>75</td>
<td>80%</td>
<td>94%</td>
</tr>
</tbody>
</table>

1 - Number of documents in the database  
2 - Number of retrieved documents  
3 - Number of relevant documents in retrieved documents

**Figure 5.8. Comparative Analysis using WordNet**
The following Figure compares the precision and recall for the WordNet based search engine and Non-WordNet based search engine.

![Figure 5.9 Comparison using Precision and Recall](image)

5.7 SEMANTIC REASONER SYSTEM USING ONTOLOGY

Semantic Reasoner is the second module of semantic search engine system. The reasoner system is constructed using the ontology concepts. The system has aimed to provide classes, sub-classes, relations, properties and instances of keyword using ontology.

**Algorithm of the Semantic Reasoner System:**

1. Select the ontology from the GUI of Semantic reasoning system.
2. Three ontologies are constructed such as bird ontology, river ontology and flower ontology.
3. TBox reasoning and ABox Reasoning methods are used for reasoning the ontology.
4. TBox reasoning is used to retrieve the sub-classes of a domain.
5. ABox Reasoning is used to retrieve the properties and instance of a class or keyword.
5.8 MODULARS OF THE REASONER SYSTEM

The reasoner system consists of two modules

1. Development of ontology
2. TBox reasoning and ABox Reasoning

5.8.1 Development of Ontology

Practically, the construction of ontology is a difficult task (Kuhanandha Mahalingam 1997). In this research work, a database has been created which simulates the ontology structure. Three ontology structures have been developed for three different domains. The domains are birds, flower and river. The three types ontology structures have been developed which are shown in the following Figures.

5.8.1.1 BIRD ONTOLOGY

![Figure 5.10 Ontology for Bird](image-url)
The ontology for bird is shown in Figure 5.10. Bird is the class. Bird names are the subclass and Kingdom, Phylum, Class, Order, Family, Genus, Species are the properties.

5.8.1.2 FLOWER ONTOLOGY

Figure 5.11 shows the ontology for flower. Flower is the class. Names of flower are the subclasses. The Kingdom, Division, Class, Order, Family, Genus, Species are the properties and the other values are the instances.
5.8.1.3 RIVER ONTOLOGY

Figure 5.12 shows the ontology for river. River is the class. Names of the river are sub-classes. The district, origin, joins, length of river are the properties.

5.8.2 TBox Reasoning and ABox Reasoning

In the proposed system, the user selects the ontology in the GUI as shown in Figure 5.13.
Description logics are a family of knowledge representation languages which can be used to represent the concept definitions of an application domain in a structured and formally well-understood way. The name description logic refers to concept descriptions which are used to describe a domain and the logic-based semantics.

There are two types of reasoning techniques in description logic. They are Tbox reasoning and ABox reasoning. TBox reasoning describes the relations between concepts, while the ABox reasoning describes relations between individuals and concepts. For example,

(1) Every bird is a flying creature belongs to the TBox, and

(2) American robin is a bird belongs to the ABox.
5.8.2.1 Tbox Reasoning

TBox reasoning describes the relations between concepts. The implementation results show TBox reasoning for bird which is shown in Figure 5.14.

![BIRD ONTOLOGY]

**Figure 5.14 TBox Reasoning for Bird**

Western birds are class of birds. In Tbox reasoning, western bird’s names are retrieved. Similarly, in flower ontology the names of Gardening Flower are retrieved which is shown in Figure 5.15
In river ontology, the types of river in India are retrieved as shown in Figure 5.16.

Figure 5.15 TBox Reasoning for Flower

Figure 5.16 TBox Reasoning for River
5.8.2.2 ABOX REASONING

ABox reasoning describes relations between individuals and concepts. The experimental results of Abox reasoning are shown in Figure 5.17. In ABox reasoning, the SQL query is given as input. For example, “Select kingdom from Chipping sparrow”. The location of the bird Chipping sparrow is Animalia (country). The result (Animalia) is returned using ABox reasoning. Also, RDF triple is formed using the result of the query which is shown in Figure 5.18.

![Figure 5.17 ABox Reasoning](image-url)
5.9 PERFORMANCE ANALYSIS

The query “semantic search engine” is given in a search engine and it retrieved 491 documents and the number of relevant documents is 215. In the proposed system, these 491 documents are used as database and it retrieves 312 documents out of 491 documents. The collected data is given in Table 5.2
Table 5.2 Comparison of Ontology based Search Engine

<table>
<thead>
<tr>
<th>Search Engine type</th>
<th>User Query</th>
<th>No. of keywords</th>
<th>Total number of Retrieved Documents</th>
<th>Total number of relevant documents in retrieved documents</th>
<th>Recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Search Engine</td>
<td>Semantic Search Engine</td>
<td>3</td>
<td>491</td>
<td>215</td>
<td>43.7</td>
</tr>
<tr>
<td>Ontology based Search Engine</td>
<td>Semantic Search Engine</td>
<td>3</td>
<td>312</td>
<td>256</td>
<td>82.1</td>
</tr>
</tbody>
</table>

Figure 5.19  Comparison of proposed system with the General Search Engine

1 - User Query
2- No. of keywords
3- Total number of retrieved Documents
4- Total number of relevant documents in retrieved documents
Figure 5.19 shows that more number of relevant documents are retrieved using ontology concepts.

5.10 CONCLUSION

The proposed semantic search engine is developed using the semantic concepts. In Semantic Parser System, the query is preprocessed and sample ontologies are developed. The Proposed system is compared with the Non WordNet (Tikk et al, 2006) based Search Engine. The performance is improved in the implemented System.

The semantic reasoner system is constructed by using ontology to improve the efficiency of the system. Ontology (Jun Fang et al, 2007) is developed for retrieval of classes, sub classes relations and properties of keyword. The proposed Ontology based Search Engine is analysed with the General Search Engine System (Iosif et al, 2007). The implementation results show the improvement in the performance in the information retrieval. The performance of the system is measured by using precision and recall methods.