Abstract—The aim of this paper is to develop a ‘Semantic Search Engine’ which is basically used to search content by using an understanding of the user's intent and also the contextual meaning of the search query. Search results are made more relevant and accurate as compared to usual search engines by further involving factors such as query generalization and specialization and concept matching. It is an artificially trained search engine to make intelligent inferences and associations between various important properties of the response object.

Index Terms—Semantic Search, Forward and Backward Changing, Search Algorithm Complexity.

I. INTRODUCTION

Semantics is the study of meaning. It focuses on the relation between signifiers, like words, phrases, signs, and symbols [3]. The term was coined by Tim Berners-Lee, the inventor of the World Wide Web who defined it as "a web of data that can be processed directly and indirectly by machines." [9] Semantic Web involves three technologies Resource Description Format (RDF), Web Ontology Language (OWL) and Extensible Markup Language (XML).

RDF is a method for conceptual description of web resources in variety of formats. In Semantic Web applications, and in relatively popular applications of RDF like RSS and FOAF (Friend of a Friend) [12], resources tend to be represented by URIs that intentionally denote, and can be used to access, actual data on the World Wide Web. OWL is a language for semantic web with pre-defined meaning. It contains classes, properties and data values which are stored as web documents. XML is a markup language which defines rules for encoding both human readable and machine readable documents.

Our semantic web search engine is using API key of Wolfram|Alpha [10] to get access to data. When a user enters a query, the keywords from the query are broken and sent to an API. The output of API engine is the XML document. The XML document is collected in an object oriented PHP data structure. This xml document is then parsed to collect relevant data that we have defined in the ontologies [2] and convert it into xml tree. We store the xml tree in a response object and then we represent the results in form of pods and sub-pods. The most significant ability of our semantic search engine is to make its own decisions and relationships between various attributes of the response object. Our search engine also takes into account the polysemy associated with a keyword and we represent all those meanings in our search result in a separate section named ‘Assumptions’. Our semantic search engine is an artificially trained program to make intelligent inferences and relationships between various properties of the object.

II. BACKGROUND

A. Natural Chaining Methods

In “Semantic Search”, there are two natural chaining methods[11]:

- Forward, data driven, antecedent approach
- Backward, goal driven, consequent approach

1) Forward Chaining Approach:

As facts are asserted, the LHS of all rules are checked for a consistent matching set. If so the RHS is asserted.

Examples: If (and (Parent ?x ?y) and (Parent ?y ?z)) Then (Grandparent ?x ?z)

2) Backward Chaining Approach:

If you want to prove the RHS, try to prove the antecedents in the LHS.

If (and (Grandparent ?x ?y) (Gender ?x male)) Then (Grandfather ?x ?y)

B. Process of collecting results:

- Hypothesis generation via data-driven triggering
  - Frame moves into short term memory
  - “Nearby” frames become semi-active
- Hypothesis testing via calibrating match of data & frame
  - Match of frame and data
  - Sufficiency, exclusionary rules
- Scoring
  - Ability to explain the findings
- Additional data gathering to fill terminals

C. Ontology

1) Aspects of an Ontology

- Content
- Form
- Purpose
- Development
  a) Content
  - types of objects, relationships
  - e.g. the blocks world conceptualization includes:
    - Object Classes: Blocks, Robot Hands
    - Properties: shapes of blocks, color of blocks
    - Relationships: On, Above, Below, Grasp
    - Processes: stacking plan for a tower
  b) Form
  - Is the taxonomic relationship (instance-of, subclass) primary?
**SEMANTIC SEARCH ENGINE**

- Are definitions of, or constraints on, terms provided?
- Is the definitional language as rich as a full logic?
- Is it process-centric or object-centric?
  - **c) Knowledge sharing (E.g. Between people, software systems, agents)**
    - Knowledge reuse
      - E.g. When models or systems change
    - General (common sense) or domain specific
  - **d) Development**
    - Is it acquired or engineered?
    - If acquired, what about:
      - Quality of knowledge
      - Diversity of content
      - Trust in knowledge
      - Unpredictable use

**III. METHODOLOGY**

**A. Handling Polysemy**

When a user enters a search query, it can have many possible meanings. We represent those meanings in the form of assumptions. These different meanings can be identified by comparing the types of classes [7]. When there are two or more classes with same parent entity name but different types, they are identified as assumptions. There are two types of assumptions:

- **Clash**
  - In this type of assumption, an entity has an altogether different meaning. There is no relationship between types of the entity.
- **Subcategory**
  - When types of two or more entities have some relationship between them, they fall under subcategory.

**B. K-Nearest Neighbour Paradigm**

1. **Step 1** Convert fetched data (XML File) to a graph, with each node representing the particular class associated with the center entity.
2. **Step 2** Assign the value of K which is a finite number, depending on the number of edges in the graph.
3. **Step 3** If the graph has K edges, pick up the K nearest edges, starting from the first edge.
   - If a pod (node) has subpods, repeat the above algorithm recursively.

**C. Decision Boundaries**

Depending on the number of neighbours, we implement bisectors at each neighbour to create a unique class for each neighbour. In our semantic search engine, whenever we discover a new pod we create a boundary around it and present all information associated with that pod in a separate class. The name of the class is same as the name of the pod. The decision boundaries help in presenting information to the user in a structured manner.

**D. Decision Trees**

We are applying forward chaining approach to the XML file fetched in the backend. Starting from the first pod, we create a decision tree, with each node representing a particular class associated with the entity. The tree is terminated when we reach the last pod of the XML file.

**E. Nodes as vectors of tags**

The nodes of the graph can be theoretically viewed as vectors of tags. Each class (pod) can contain subpods and states as dimensions of the vector. The dimensions of each vector can differ, depending upon the number of tags inside the parent tag, e.g., the pod can be viewed as:

\[
\text{pod} = \{\text{subpod}, \text{images}, \text{states}, \ldots\}
\]

**F. Semantic Similarity (Comparisons Module):**

In the comparison module, we take as input two entities from user between which a comparison is desired. While computing the results, we take into account the semantic similarity of tags associated with each of the entities [4]. Only the information associated with similar semantic nodes is displayed for comparisons. If \(R(a,b,c,d,e)\) and \(S(a,b,c)\) are tags associated with two entities \(R\) and \(S\), we compute the intersection of associated tags (in this case \(a,b,c\)) and project the results associated with only those nodes.

**G. Complexity**

If \(k\) are the number of pods in the fetched xml file and each pod has \(m\) subpods and \(n\) states, then the algorithmic complexity of our algorithm is \(O(k*m + k*n)\)

**IV. CONCLUSIONS AND FUTURE SCOPE**

Our research paper on ‘Semantic Search Engine’ is one of the applications of broad field of ‘Semantic Web’. Our semantic web search engine is a fact and knowledge oriented search engine. It understands the context of the user when displaying the results. Rather than displaying the web links as results, it presents only the information intended by the user. We have developed various specific modules related to semantic search, like comparisons module, weather module, images module, etc. In the future, many specific rule set for other domains can be integrated into the semantic search engine. More advanced natural language processing algorithms can be developed to understand the meaning of the query.

**V. SCREENSHOTS**

**Figure 1.** User enters word ‘gandhi’

**Assumptions:**

- **Clash:**
  1. Person a person to change search to this assumption [click here]
  2. Movie a movie to change search to this assumption [click here]
  3. Book a book to change search to this assumption [click here]
- **SubCategory:**
  1. MahatmaGandhi Mahatma Gandhi to change search to this assumption [click here]
  2. IndiraGandhi Indira Gandhi to change search to this assumption [click here]
PAPER
SEMANTIC SEARCH ENGINE

Input interpretation
Mahatma Gandhi (activist)

Basic information

<table>
<thead>
<tr>
<th>full name</th>
<th>Mohandas Karamchand Gandhi</th>
</tr>
</thead>
<tbody>
<tr>
<td>date of birth</td>
<td>Saturday, October 2, 1869</td>
</tr>
<tr>
<td>(age: 73 years ago)</td>
<td></td>
</tr>
<tr>
<td>place of birth</td>
<td>Porbandar, Gujarat</td>
</tr>
<tr>
<td>date of death</td>
<td>Friday, January 30, 1948</td>
</tr>
<tr>
<td>(age: 78 years)</td>
<td></td>
</tr>
<tr>
<td>place of death</td>
<td>Delhi, Delhi</td>
</tr>
</tbody>
</table>

Image

Figure 3. Search results for Mahatma Gandhi (Person)

Input interpretation
Gandhi, Fighter Without a Sword (book)

Basic properties

<table>
<thead>
<tr>
<th>author</th>
<th>Jeanette Eaton</th>
</tr>
</thead>
<tbody>
<tr>
<td>publisher</td>
<td>Morrow</td>
</tr>
<tr>
<td>awards</td>
<td>Newbery Honor</td>
</tr>
</tbody>
</table>

Figure 4. Search results for ‘Gandhi’ (Book)

Basic information

<table>
<thead>
<tr>
<th>full name</th>
<th>Sachin Tendulkar</th>
</tr>
</thead>
<tbody>
<tr>
<td>date of birth</td>
<td>Tuesday, April 24, 1973</td>
</tr>
<tr>
<td>(age: 39 years)</td>
<td></td>
</tr>
<tr>
<td>place of birth</td>
<td>Mumbai, Maharashtra</td>
</tr>
<tr>
<td>date of death</td>
<td>Friday, January 30, 1990</td>
</tr>
<tr>
<td>(age: 38 years)</td>
<td></td>
</tr>
<tr>
<td>place of death</td>
<td>Dehradun, Uttarakhand</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>full name</th>
<th>Ricky Ponting</th>
</tr>
</thead>
<tbody>
<tr>
<td>date of birth</td>
<td>Thursday, December 19, 1974</td>
</tr>
<tr>
<td>(age: 38 years)</td>
<td></td>
</tr>
<tr>
<td>place of birth</td>
<td>Launceston, Tasmania, Australia</td>
</tr>
</tbody>
</table>

Images

Figure 5. Comparisons Module

Figure 6. Results for previous query

REFERENCES
[10] Wolfram|Alpha, computational knowledge engine

AUTHORS
Senthil J, Margaret Anouncia, and Abhinav Kapoor are with the School of Computing Science & Engineering, VIT University, Vellore, India.

Submitted 24 April 2013. Published as re-submitted by the authors 15 November 2013.