An Addop Enabled Semantic Text Analysis for Intelligence Rich Question Answering System

M. Venkatraman, V. Sriram and B. Vigneshwar

Rajalakshmi Engineering College, Affiliated to Anna University Chennai, Rajalakshmi Nagar, Thandalam, Chennai-602105, Tamilnadu, India

Abstract: The effective study on various technologies associated with web that enriches the idea of semantics and its services were introduced. An ADDoP, implemented technology that includes AJAX controlled document for mining Data and DOM enabled document for Pattern matching encapsulated to form this technology, services the web with rich context. The main idea is to retrieve or collect answers for the questions queried and return the answer associated with it. This system is designed for one particular domain or to replace a blog or page called Help/Forum/Complaint desk in a web page or even be designed as a system for assisting corporates. The efficiency of our system depends on the accuracy of search strategy implemented in it. The accuracy of the search is proportional to the satisfaction of the user in this system. This paper depicts the methods and peculiar usage of technologies for retrieving effective answers for the questions queried using the system’s intelligence.

Key words: Word Separation · Text analysis · Text Search · Pattern matching · Intelligence response

INTRODUCTION

Irrespective of the domain involved, there is always an option for clarifying the queries that get rise while performing one operation. For example, if a person is filling his registration form, he/she has a doubt in completing it. Now in the present case it is that we enter into the discussion forum /Help link to search the answers from the listed FAQs. It is done by searching the question that is already posted. The search time for retrieving the desired answer is difficult. Moreover the desired answer is available, if and only if posted by administrator. The Intelligence developed for overcoming these sorts of difficulty in a web document is Semantics Intelligence. The efficient usage of technologies in construction of our question answering system is quite typical. Of course, we have a typical question answering system that is used for displaying the answers for the corresponding questions. It is not developed in a web host as such it cannot be used as web document process engine. Here it is a web enhanced spontaneous reply delivering system that controls the web intelligence and the desired answer to be rendered. The main feature of this system is to render the required question with suitable answers as quick as possible. The optimal way for retrieving the answer is through our search strategy embedded in our system. First and foremost the retrieval as similar as previous implemented systems i.e. retrieving the details from the database that is initially fed with the concerned data. The extra intelligence in retrieving the details is mapping the root word with the associated keyword in the database. If the question is repeated by the user in different forms, the web intelligence is brought into picture by ADDoP technology by matching the data that is used and mining data associated with it. This explains the importance of data mining and pattern matching. The current WWW has a huge amount of clustered data that is often unstructured and usually only human understandable. The Semantic Web aims to address this problem by providing machine interpretable semantics to provide greater machine support for the user. The Semantic Web has a layer structure that defines the levels
of abstraction applied to the Web. At the lowest level is the familiar World Wide Web, then progressing to XML, RDF, Ontology, Logic and Proof. The optimality lies between the amount of usage and frequency in the usage of pattern particularly. This effect holds well in current trends of web.

**Literature Survey:** Though most of the search engines exist in the world of web, they are efficient in searching related posts in the domain and retrieves as list of details associated with that search keyword. The main part that motivated us, is to answer certain questions without any contradictions. To explain in detail, when a question is asked to our semantic search engine, the corresponding answer to that particular question is retrieved precisely. The Concept of Web mining and the semantic web [5] led the featured development of analysis and retrieval through the structure of data processed [1-3]. explains the structural form of data to be retrieved for the underlying system which in forecasts the XML schema. This is achieved through Ontology learning, Mapping and merging ontologies, Instance learning, semantics created by structure. Based on these principles, ADDoP is designed to make the system predominant like OWL. This motivation led to our system design, the question and response engine. To focus on this response engine, we framed an algorithm that portrays completely about the technique of parsing the sentence into words considered when separated by white spaces. Considering as a keyword, it searches throughout that particular domain and retrieves the answers in an efficient manner. The existing approach holds with search of posts related with the keyword entered in the text box of search engine. The keyword is mapped with the domain and related posts are retrieved. But our system approach will efficiently retrieve the exact answer form the domain wherein the keyword is set with the grammatical constraints of the language [7,8]. The question answering system based on the semantic web technology is a part of the Business Intelligence that ensures the technique of retrieving the desired answers in a more efficient way such that it encompass the operations of search and retrieval in milliseconds. The search technique involved is the AVL tree search strategy. The search is more optimal so that relevant meaning of one particular word is mapped to another and the answer is then retrieved; the specialty is that the word that has relevant meaning in a domain is mapped with the root node value and then if needed appended or even displayed with the result. This motivated us to bring more intelligence in a web and to make it more efficient. The spontaneity is more when mining data, since the intelligence is dependent on matching the pattern among various questions. Thus ADDoP is a pattern based rich technology that explains the usage of AJAX, Data Mining technique for a DOM based web page, servicing web for retrieving the answers as efficient as possible.

**System Functionalities**

**System Design and Implementation:** In the system, the action is initiated by entering the question and selecting the option called “Search”. The action commences by triggering the functionality of ADDoP. The search engine forwards the sentence as a string to the function i.e. it reads the sentence for parsing. Set the array size so that it holds the all the individual words parsed from the sentences. Whitespaces in the string serve as the reference for extracting individual words from the sentence. Moreover, every preposition in English language is followed by a verb. Separating the prepositions, conjunctions, articles and Interrogative words, leaving the rest of the sentence to form the verb and noun index used for searching the answers for the queried question. So, declare the prepositions, conjunctions, Interrogative words, articles as a separate array such that searching strategy wholly lies within the existing initializations. If a preposition is identified in the sentence array, the proceeding word is a verb in almost all cases. Add the latter word to the verb array. Based on the grammatical rules of the English language, the verb and noun from the sentence are separated. Nouns are added to the index [ ] and verbs are added to the verb [ ] array. The question inside the textbox is interpreted in such a way that the goal path is reached by the keyword fetched in the index [ ] and verb [ ]. The path from verb and the
index is accountable for the exact answer retrieval which is then displayed in the answer text box. The response time for the answer retrieval for a simple question is in milliseconds. The efficiency and the reliable property of ontology lie in the search strategy. The semantic web design enhances the look and feel of the system and attracts the user. In addition to this, the ADDoP package enables the technology of pattern matching and data mining techniques that relinquishes the features of AJAX, DOM and Scriptlets. "Pattern mining" is a data mining method that involves finding the patterns existing in data. In this context, patterns often means association rules. Then based on the algorithm that is defined the answer gets retrieved or mined from the database in more efficient manner.

Algorithm:

function OGP(question) returns verb[] and index[]
begin {OGP(question)}
    enter the question to get answer
    //sentence is always with at least one noun, one verb and it can have either any prepositions, conjunctions and other connectives etc.
    separate words from the sentence
    //words are separated by white space then consider as separate words
    //sentence should start with interrogative form
    set sentence[]={array of words};
    set interrogative[]={who,which,when,what,where,how};
    //{wh...and how }form
    set article[]={ a,an, the};
    set conjunctions[]={but,yet, between, or, while, if, for, neither..nor,either..or};
    set prepositions[]={on,of,among, above, below, beside, inspite,despite,under,up,down,for,with,to};
    for(variable i=0;i<sentence.length;i++)
        do
            for(variable j=0;j<=prepositions.length;j++) then
                do verb=sentence[i+1];
                    goto label;
                endif
            end{for prepositions}
        end{for variable i}
    end{for variable j}
    for(variable l=0;l<=article.length;l++)
        do
            if(strcmp(sentence[l].article[1])==0) then
                goto label;
            endif
        end{for article}
    for(variable m=0;m<=conjunctions[m]==0) then
        goto label;
    end{for conjunctions}
    index[]={sentence[i];
    label:
    end{for sentence}
    search for the answers using verb and index[];
end{begin}

Table 1: General Algorithm of the System

<table>
<thead>
<tr>
<th>Begin {answering system}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get question from the query box</td>
</tr>
<tr>
<td>OGP (question) returns verb and index[]</td>
</tr>
<tr>
<td>// Check verb and index[] with database</td>
</tr>
<tr>
<td>Label 1: If (verb and index[] found in database)</td>
</tr>
<tr>
<td>Return corresponding answer</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>value = checkList(verb)</td>
</tr>
<tr>
<td>if (value != null)</td>
</tr>
<tr>
<td>{ //replace verb with value</td>
</tr>
<tr>
<td>goto label1;</td>
</tr>
<tr>
<td>} end if (value != null)</td>
</tr>
<tr>
<td>else {</td>
</tr>
<tr>
<td>//mine the answer for index[] and verb from the web</td>
</tr>
<tr>
<td>//display answer to the user and update database</td>
</tr>
<tr>
<td>if (answer already in database)</td>
</tr>
<tr>
<td>if (index[] matches)</td>
</tr>
<tr>
<td>{ //find root verb</td>
</tr>
<tr>
<td>from database</td>
</tr>
<tr>
<td>append (rootverb,verb)</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>//create new node for verb and update database</td>
</tr>
<tr>
<td>add (verb)</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>} end if (answer already in database)</td>
</tr>
<tr>
<td>end {answering system}</td>
</tr>
</tbody>
</table>

The Operational algorithm of the system (Table 1) is depicted as follows so that the analysis is based on the system’s algorithm and its efficiency dependable on the spontaneity of the search strategy. The algorithm outlines the sequential action of the system. The following algorithm shows the activity and operation of the system considered.

OGP for Mining Data: In this system, the action is triggered by entering the question and pushing the “Search” button. After loading and reading the ontology
for preparing the model, the process of parsing the sentence is started. Initially when a search button is clicked, the search engine forwards the sentence as a string to the function i.e. it reads the sentence for parsing. Set the array size so that it holds all the individual words parsed from the sentences. Whitespaces in the string serve as the reference for extracting individual words from the sentence. Moreover, every preposition in English language is followed by a verb. Separating the prepositions, conjunctions, articles and Interrogative words, leaving the rest of the sentence to form the verb and noun index used for searching the answers for the queried question. So, declare the prepositions, conjunctions, Interrogative words, articles as a separate array such that searching strategy wholly lies within the existing initializations. Apart from it, iterating the sentence - array of words, compares the string for preposition, conjunctions and articles. If a preposition is identified in the sentence array, the proceeding word is a verb in almost all cases. Add the latter word to the verb array. Based on the grammatical rules of the English language, the verb and noun from the sentence are separated. Nouns are added to the index [ ] and verbs are added to the verb [ ] array. The question inside the textbox is interpreted in such a way that the goal path is reached by the keyword fetched in the index [ ] and verb [ ]. The path from verb and the index is accountable for the exact answer retrieval which is then displayed in the answer text box. The response time for the answer retrieval for a simple question is in milliseconds. The efficiency and the reliable property of ontology lie in the search strategy. The semantic web design enhances the look and feel of the system.

**Search Strategy:** From the OGP function the verb [ ] is obtained. If the user enters the question that corresponds for the noun and the interrogative word in the database but the verb used is unknown, a search strategy to find the relevant match for the verb is required. Moreover, this search technique is useful to render the verb from the asked questions and the because of the system’s intelligence the ability to understand the question asked by the user rapidly increases. The new verb found in the question, is not present in the knowledge base of the system, with the help of user interaction the system adapts itself to append the new verb under the appropriate root verb or to create a new root in the verb tree. The Search technique implemented here is the most optimal searching strategy- AVL-Tree Search Strategy. The AVL-tree search technique involves the principle of searching of technique using AVL nodes, each node holding a verb. The chief node is made a root node and similar meaning associated with that verb is appended as a child under the root node in binary fashion. The question that is requested by the user is separated as a sequence of words and the verb is identified from those array of words. This verb is searched in the verb tree and if present the root the tree it is present in is used for indexing the answer retrieval process from the database.

If it is not found, return null stating that no verb is found related to this. As stated in the above Figure the word ‘modify’ is to be searched. First and foremost the word starts its search from the index a[0] that holds 43211 as value. The root value is found to be ‘submit’. Searching in the binary fashion or recursively the word gets found. Else it moves to the other index value and proceeds further. This optimal feature of search is highly efficient. This AVL tree search retrieves the root node (verb) that is similar to the verb from the question and retrieves the answer that is mapped with the index from the database.

**Word/Pattern Matching:** The other case in which if the word is not found in the array indices or the tree, from the user interaction the system’s intelligence finds the relevant root node which has the verb that has a similar meaning to this new word. The new word is then appended under this root node thus forming a tree of verbs with similar meaning. This search strategy improves the process of retrieving exponentially. Consider for example, each node when unexpanded is a fringe.
It is expanded but not yet generated. In the same way the word can be appended by generating the node associated to the root node. In the below mentioned figure, the array a[0] holds the address value of 32319. If a word is to be appended, then initially create a node by allocating some memory to it. So that created node is generated at the left foremost fringe. The following algorithm states the technique for appending a word. In order to append a word after creating a node in the fringe, search for the verb from the verb list for making a step of verification. After the successful response then add verb to the node that is generated. The word gets appended to the verb AVL such that it balances itself to make a balanced tree. The tree then makes the data to get retrieval as its faster level. If while balancing, the root node is altered so that some other node forms the root, then the database is updated with the newly formed root. The optimal way of retrieving the data is made efficient and the data is rendered in the designated way. The address holds the value of root node and then matches the verb that is similar.

**RESULTS**

Complexity arises in the system only due to the operational delay or space latency. Since the web document is enabled with semantics intelligence, the complexity is less on the basis of search and retrieval.

**Time Complexity:** The web oriented systems always render the details based on the speed of the internet. The search follows the dynamic programming technique and the complexity is at its minimum.

**Space Complexity:** The amount of space required to make the verb list ordinate with its indices is the complexity and is minimum based on the language used to implement the tree.

**Completeness:** The answer that gets retrieved by any one of the cases is optimal. Thus concludes the completeness of the problem that is the answer to be retrieved for the question. The alternate solutions that get portrayed accounts for the efficient method of answering the question.

**Optimality:** Though the time and space complexity of the system is similar in the way they perform, the solution that is rendered should be satisfactory to the user. If not, the optional solution indexed using the keywords gets displayed. Thus the optimal way of getting a solution is realized by dynamic programming paradigm which is possible in the case of web programming using various technologies. This optimality is one of the pros of ADDoP.

**CONCLUSION**

This system described the question and answer system based on the principles of ontology and a web technology enabled to be not browser specific by
adopting semantic web techniques. The aim of implementing this system was to provide exact answers to the questions asked by the users and retrieves the exact details required. Performance issues can be enhanced. The future work can be in any different path and extended in any direction. Even the question pattern can be moderated such that it relies on a more optimal way than this, then the search strategy can be enhanced. The answers may be of any sort depending on the query requested. There might be precise answers for the questions and can be made logical too. Moreover, much trickier questions that require complex automated reasoning processes can also be handled by the system.

REFERENCES

2. Semantic Web Service Description by Matthias Klusch.
3. Semantic Web Mining by Khoi-Nguyen Tran, a Postgraduate Student from School of Computer Science, The Australian National University, Canberra, ACT 0200, Australia.
6. Automatic Composition of Semantic Web Services by Srividya Kona, Ajay Bansal, Gopal Gupta from The University of Texas at Dallas Richardson, TX 75083.
9. W3C Semantic Sensor Networks Ontologies, Applications and Future Directions by Cory Henson from Ohio Center of Excellence in Knowledge-enabled Computing (Kno.e.sis ) Wright State University, Dayton, Ohio, USA.
10. Semantic Web Ontologies by Chris Welty, an IBM Researcher (also W3C Semantic Web Best Practices WG).