Domain Specific Ontology based Query processing System for Urdu Language

Rukhsana Thaker
CSE Department, CoET
BGSB University, Rajouri, J&K

Ajay Goel, PhD
Head, CSE Department
Baddi University, Baddi, HP

ABSTRACT

The tremendous increase in the multilingual data on the internet has increased the demand for efficient retrieval of information. Urdu is one of the widely spoken and written languages of south Asia. Due to unstructured format of Urdu language information retrieval of information is a big challenge. Question Answering systems aims to retrieve point-to-point answers rather than flooding with documents. It is needed when the user gets an in depth knowledge in a particular domain. When user needs some information, it must give the relevant answer. The question-answer retrieval of ontology knowledge base provides a convenient way to obtain knowledge for use, but the natural language need to be mapped to the query statement of ontology. This paper describes a query processing system based on ontology. This system is a combination of NLP and Ontology. It makes use of ontology in several phases for efficient query processing. Our focus is on the knowledge derived from the concepts used in the ontology and the relationship between these concepts. In this paper we describe the architecture methodology and algorithm with results of query processing system for Urdu

General Terms

Query Process, Ontology, Properties, Instance.

Keywords

Domain, Adjacent Keyword, Classes.

1. INTRODUCTION

The growing interest in providing different information on the web has increased the need for a complicated search tool. Most existing information retrieval systems only provides documents, and this often makes users read a relatively large amount of full text [1].The study of question answering systems (QAS), which enable people to locate the information they need directly from large free-text databases by using their queries, has become one of the important aspects of information retrieval research[2].

Question-answering (QA) study emerged as an effort to deal this in sequence-excess problem. QA systems are classified in two main parts: namely open domain QA system and closed domain QA system. Question which deals with nearly everything and can only relies on worldwide information, such type systems are called as open domain question answering system. On the other hand, closed-domain question answering deals with questions under a particular domain (music, weather forecasting etc.) The domain specific QA system involve deep use of natural language processing systems formalized by building a domain specific ontology[3]

From the last decade internet users are increasing in more tires. Information is present not only in English language but also in many other languages. This is due to the advent of Unicode scheme that user can contribute their knowledge over the web in their own language. Various scientists have worked on QA with English, French, and Chinese etc languages. But from the last few years scientists move their research into regional language like Punjabi, Tamil, and Hindi etc. Among variety of languages, the Urdu language is among the largest spoken languages of the world and state language of J&K state. Due to the availability of large amount of Urdu language documents over the web, it is required to develop a sophisticated information retrieval system to utilize the information efficiently. We propose a Query Processing System Architecture that uses words of the sentence (question) typed as a source to search answer. Principle of our Question Processing system is that the system gives user a set of sentences in Urdu language containing the words of sentence typed. Here we described a model of a system, which accepts user queries in natural language (Urdu) after analyzing those queries on the bases of ontology match them with the information stored and result is displayed to users, thus helping users to get the most wanted information without penetrating the huge amount of information existing on the web.

The outline of the paper is as follows. The paper is organized as follows In Section II we discussed related work of this object; Section III discussed the proposed architecture of the query processing system. Section IV discusses the methodology in detail. Section V presents Experiments and Section VI presents the conclusion.

2. LITERATURE REVIEW

Machine translation (MT) was the first computer-based application related to natural language which came into existence in 1940’s but first project was used in 1946 in World War II to break enemy code [4]. In 1969 Natural Language Question Answering Systems was made which focused on syntactic, semantic, and logical analysis of English strings [5].

In 2004 a web based Chinese Automatic Question Answering System was made by Cai Dongfeng and Cui Huan [6] which uses Google Web services API. In 2006 the geographic core model was made which supported every Geographic Feature (GF) detected in collection of text or image documents [7]. Later on in 2008 A Question Answering Systems was made which automatically generates questions related to the document. QA System based on the analysis of interrogative sentence and the answer type was made for Chinese language in 2009 by CunLi Mao and others which use to carry the text retrieval with the help of the domain knowledge and obtain the relevant paragraphs of the question [8].

Kanaan et el [9] described Arabic QA system which makes use of data redundancy rather than complicated linguistic analyses of either questions or candidate answers, to achieve its task. Akour et el introduced a QA system (QArabPro) for Arabic Language. The system handles all types of questions including (How and why). But generally similar to those used in a rule based QA for English text , the authors uses a set of rules for each type of WH question.

Recently, researchers have been attracted to the task of developing Ontology based QA systems such as Querix [10], and AquaLog [11]. Querix introduced by Kaufmann et al. is another ontology-based question answering system that translates generic natural language queries into SPARQL. In case of ambiguities, Querix relies on clarification dialogues with users. In this process users need to disambiguate the sense from the system-provided suggestions. AquaLog Proposed by Lopez et al. [11] is a portable semi-automatic QA system that combines natural language processing, ontologies, logic, and information retrieval technologies. We say that AquaLog is portable, because the configuration time required to customize the system for a particular ontology is negligible. In AquaLog the ontology is used intensively as it is utilized in the refinement of the initial query, the reasoning process, and in the similarity algorithm

Vanessa Lopez et al "Question Answering on the Real Semantic Web" to conclude with, Power Aqua balances the heterogeneous and large scale semantic data with giving results in real time across ontologies, to translate user terminology into distributed semantically sound terminology, so that the concepts which are shared by assertions taken from different ontology's have the same sense. The goal is to handle queries which require to be answered not only by consulting a single knowledge source but combining multiple sources, and even domains.

3. ARCHITECTURE OF QUERY PROCESSING SYSTEM

This section presents the architecture and functionality of system. The working of the proposed model is as follows. Here user is provided with the facility to type his question in Urdu. This question is used to extract all the possible answers for it. The architecture for System is as revealed in Figure 1.

1. Query interface is used to retrieve the question posted by the user in Urdu language.
2. NLP parser is used to parse the query.
3. Interpreter is used to remove the stop words and for stemming.
4. Query formulation is used to formulate the query based on the domain ontology
5. Ontology is used to define classes and concepts of the dataset for a particular domain in Urdu language

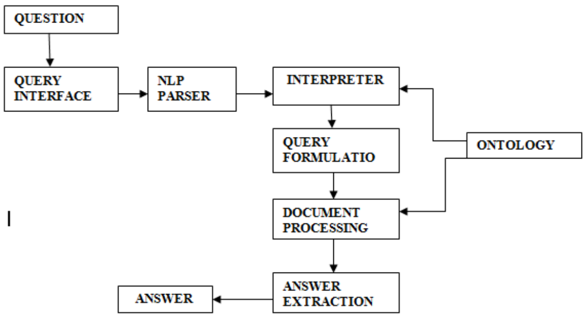

Fig 1: Architecture of query processing system

4. METHODOLOGY

This section describes the proposed methodology for query processing in detail. It includes the following components:

1. Domain Selection and Ontology Construction

2. Parse User Query

3. Find information from constructed ontology based on parsed query

4. Data retrieval from the document based on extracted information from Ontology

4.1 Domain Selection and Ontology Construction

4.1.1 *Domain Selection*

The present work utilizes unstructured and ungrammatical documents written in Urdu Language. Ontology can work on open domain as well as Close domain. In our work we have chosen closed domain of Car Ads where user can get information regarding the Car available for sale. Figure below shows example of Car with associated properties in graphical form.

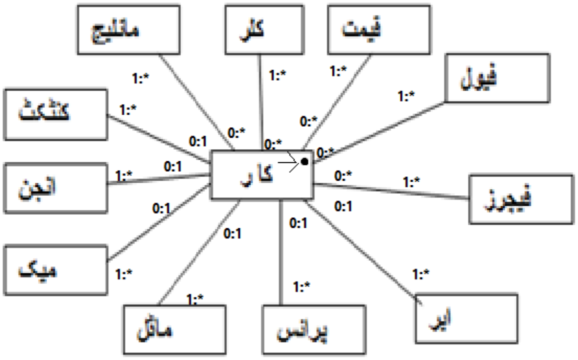

Fig2: Concept of Car

The boxes represent concept and properties. The arrow with dot (→.) in the car concept denotes that it is the primary object, which means it is the main idea behind describing the ontology. Lines represent the relationship sets. Number by colons such as "0:1" represent the participation constraints for object in relation. "*" denotes unlimited cardinality.

4.1.2 *Ontology Construction*

Ontologies are content theories about the sort of objects, properties of objects and relations between objects that are possible in a specified domain of knowledge. They provide potential terms for describing our knowledge about the domain.

The knowledge about the domain can be represented in the form of concept of properties. So first of all we have to select concept and properties associated with that concept . For example, in case of Car ads, domain is car and properties that are associated with Car are Model, Make, Year, Mileage etc. These concepts and properties are classes in the ontology. Next step is to define relationship between these classes by defining Object Properties and data properties. Both Object properties and Data properties have domain and range. Object properties define relation between classes where both domain and range are classes whereas Data properties are relation between classes and data values where domain is a class and range is set of values. For Example, in case of Car ads, one

Object property, has_price, has domain Car class and range is Price class and for Data Property, price_value, domain is Price class and range is set of values. Next we have to give some additional knowledge in the ontology according to the data which are present in our data set. In the Figure 2 below shows the car ad in Urdu Language with underlined words. These words are additional knowledge associated with our properties. For Example, 1000 cc, represents engine value. But how do we know this is engine value? This word cc represent engine and value adjacent to it i.e. 1000 represent engine value. So these kinds of words, used for additional knowledge are adjacent keywords. They shows where the data is present in the data set and the values adjacent to these adjacent keywords are actual data values.

پاور cc1000 کیمت 450000 نسان سنی ماڈل1986 ،
کلیئر ریٹرن فائل سفید CPLCاسٹرئنگ ٹیوب لیس ٹائر
کلر رابطہ0300-2582180 کیش میمو21880

Fig 3: Car Ad with underlined adjacent key words

Ontology is Constructed using Protégé Tool and is stored as RDF/XML format which can be converted to other format like Turtle or N3 using RDF convertor. Table below shows a part of ontology for Car Ads.

Table 1: Car Ontology

<table>
<tr><th colspan="3">DOMAIN KNOWLEDGE</th></tr>
<tr><td colspan="3"><b>CONCEPT</b>: کار</td></tr>
<tr><td colspan="3"><b>PROPERTIES:</b> کلر ، ایر ، پرائس ، کنٹکٹ ، فیول ، انجن، ماڈل ، میک ،مائلیج ، فیچرز</td></tr>
<tr><td colspan="3"><b>ADDITIONAL KNOWLEDGE</b></td></tr>
<tr><td><b>ADJACENT KEYWORDS</b></td><td><b>DATA TYPE</b></td><td><b>ASSOCIATED PROPERTY</b></td></tr>
<tr><td>KM/کلومیٹر</td><td>Integer</td><td>Mileage/مائلیج</td></tr>
<tr><td>کلر ، رنگ</td><td>String</td><td>Color/کلر</td></tr>
<tr><td>کیمت ، ڈیمانڈ، /-</td><td>String</td><td>Price/پرائس</td></tr>
<tr><td>Cc</td><td>Integer</td><td>Engine/ انجن</td></tr>
<tr><td>ماڈل ، ء</td><td>Integer</td><td>Year/ایر</td></tr>
</table>

4.2 Parse User Query

Although documents are typically the target for information extraction, first step is to perform extraction over the user query. To illustrate the Process, suppose the user enter the query:

“ مجے سب سفید رنگ کی نسان کار کی کیمت اور مائلیج بتائی جائے
جسکا ماڈل ۲۰۰۵ کے بعد کا ہے؟”

Consider the Ontology in Table 1. The word” رنگ “mentioned in the query matches the adjacent keyword of property Color/ کلر . Had the user used the word”کلر” instead of” رنگ”, “کلر” would still match with the Color/کلر Property. The value associated with the “رنگ” keyword is “سفید”. As with the “رنگ”, the word “مائلیج” matches the Mileage/مائلیج property in the ontology. The Price /“پرائس” property matches with the “کیمت” keyword. Similarly نسان is not an adjacent keyword , its an instance of property Make/میک . The model keyword matches with the Year /“ایر” property in the ontology and it has value i.e. ۲۰۰۵. Lastly the greater-than-or-equal operator recognizer matches with the words “کے بعد”.

So after Parsing the query using ontology we will get that user want to get information کیمت and مائلیج about نسان car which has the following features: رنگ with value سفید, ماڈل with value > ۲۰۰۵.. We will store this information for further processing.

4.3 Find Information from Ontology

Once query is processed the next step is to find information from the ontology based on user query. For that we need to generate some rules to extract exact information from the ontology. As mentioned in the ontology we have some properties for which adjacent keywords are mentioned and some properties for which no adjacent keyword is available. So we will consider both cases.

Case I: When Adjacent keywords of properties are mentioned.

When the Adjacent keywords for properties are mentioned, it means the information can be retrieved from the document based on those adjacent keywords. For every Adjacent keyword some value is associated with it, this value will be adjacent to the Adjacent keyword. That is the actual value that user want to get. Now what type of value is that, it depends on the data type of the adjacent keyword that we have mentioned in the Additional knowledge in the ontology construction? Next we will generate rules for each data type used in the ontology to extract value of each property. Data value is extracted using regular expression depending upon the data type of that property. For example regular expression for integer data type and string data type are defined as “\w” and “d+”.

Case II: When adjacent keyword of properties are not mentioned

When adjacent keywords of properties are not mentioned than value of that property are considered as string data type and information is extracted using the instance of those properties.

Once this information is extracted from the ontology, it is used to retrieve relevant information from the document.

4.5 Data retrieval from the document based on extracted information from Ontology

Next step is to extract values / data from the document based on the rules generated in the previous step. Formal description of the algorithm is given below:

```
Let O be the Ontology created for domain
Let q be the user query in natural Urdu language
Let Ri be the rule formed for data type DTi using regular expression
While q is not empty {
Tokenize the query q into tokens t
Remove the stop words
For each token tin q{
Identify t in ontology O
If (t is a property)
```

```
Store t in a property set P
If (t is adjacent keyword) {
Find corresponding P and store all the adjacent keywords in set q_Akw
Find data type q_DT corresponding to P
Find adjacent value based on rule Ri defined for DTi and store in q_value
Find relational condition R_cnd }}}
For each Pi in the property set P, search Ontology O {
If Pi has no adjacent keyword
Add Pi to set Pa
Else
Add Pi to set Pa`}
For each Pi in Pa`{
Find adjacent keyword set Akwi corresponding to Pi from Ontology O
Find data type DTi corresponding to Pi from Ontology O}
For each Ad A in the document {
If (Pa is in A) {
If (any adjacent keyword from set q_Akw is in A){
Find adjacent value based on rule for q_DT data type from the property of that adjacent keyword and store the value in A_value}
Compare the value A_value & q_value using R_cnd condition
If the condition is true {
For each Pi in Pa` {
If (Akwi in A) {
Apply rule Ri corresponding to data type DTi for that Pi to find the adjacent value and store the value in some variable A_VARi}}}
Formulate the answer using Pa, Akwi and A_VARi to get the desired results.
```

5. EXPERIMENTS

This section evaluates the performance of the presented algorithm on car ads in Urdu language.

In order to read or write the Urdu language Google input tool has been used. To create Ontology Protégé tool has been used. To conduct the experiment a data set of car ads has been chosen.

Recall and precision value is calculated as:

$$\text{Recall} = C/A$$

$$\text{Precision}=C/(C+I)$$

Where A is the total number of ads in which properties were available

C is the correctly extracted answer

I is the incorrectly extracted answer.

Average recall and precision values are 85.4% and 97.6% respectively.

6. CONCLUSION

In this paper we have presented an ontology-driven query processing system in Urdu language. It presents the ontological approach to find the answer of the user question. Information is retrieved using ontology and some additional knowledge in the form of adjacent key words. A dataset of car ads in Urdu language is used for conducting experiments. Results show that the method gives better results for unstructured and ungrammatical data. In this method we have used additional information for extracting data. So as future work , we can modify this system so that information can be extracted without using additional knowledge. Moreover we can combine this ontology with other Ontologies to generate top-level ontology. In this process we have constructed ontology manually so for future work we can also create ontology automatically from user queries.

7. ACKNOWLEDGMENTS

We would like to thank our colleagues for their technical support.

8. REFERENCES

[1] R. Baeza-Yates & B. Ribeiro-Neto, (2011) "Modern Information Retrieval", second edition.

[2] Breck, E., Burger, J., House, D., Light, M. & Mani, I. (1999) ``Question answering from large document collections'', Question Answering Systems: Papers from the 1999 AAAI Fall Symposium, 5-7 November, North Falmouth, MA, AAAI Press, Menlo Park, CA, pp. 26-31.

[3] Muthu krishnan Ramprasath1 and Shanmuga sundaram Hariharan2, 2012, "A Survey on Question Answering System", International Journal of Research and Reviews in Information Sciences (IJRRIS), pp171-179.

[4]] Liddy, NY. Marcel Decker "Natural Language Processing" , in Encyclopedia of Library and Information Science in 2001

[5]] ROBERT F. SIMMONS, "Natural Language Question Answering Systems: 1969"

[6] Cai Dongfeng, Cui Huan, Miao Xuelei, Zhao, Ren Xiangshi 2004, "A Web-based Chinese Automatic Question Answering System", pp1141-1146, IEEE.

[7] Sallaberry Christian, Etcheverry Patrick, Marquesuzaà Christophe, 2006, "Information Retrieval and Visualization Based on Documents" Geospatial Semantics", International Conference on Information Technology: Research and Education, pp277- 281.

[8] Cun-Li Mao, Li-Na Li, Zheng-Tao Yu, Lu Han, Jian-Yi Guo , Xiong-Li Lei ,2009,"Research on Answer Extraction Method for Domain Question Answering System (QA)", 2009 International Conference on Computational Intelligence and Security, pp79-83, IEEE.

[9] Kanaan, G., Hammouri, A., Al-Shalabi, R. and Swalha, M, (2009). "A New Question Answering System for the Arabic Language". American Journal of Applied Sciences 6 (4): 797-805Y.T.

[10] E. Kaufmann, A. Bernstein, and R. Zumstein., "Querix: A natural language interface to query ontologies based on clarification dialogs," In proceeding 5th International Semantic Web Conference (ISWC 2006), pp 980–98.

[11] V. Lopez, and E. Motta, 2004, "Ontology-Driven Question Answering in AquaLog," Lecture Notes in Computer Science, Vol. 3136. Springer-Verlag, Berlin, pp. 89–102.

[12] Vincent Barbier, Brigitte Grau, Anne-Laure Ligozat, Isabelle Robba and Anne Vilnat "Semantic Knowledge in Question Answering Systems"