A Natural Language Processing based Automated Knowledge Provider System with Speech Recognition

Prasenjit Mukherjee and Baisakhi Chakraborty  
National Institute of Technology,  
Durgapur, West Bengal, India,  
prasen.mscit09@gmail.com  
baisakhichak@yahoo.co.in

Narayan C. Debnath  
Winona State University  
Winona, USA  
ndebath@winona.edu

Abstract

An Automated Knowledge Provider System (AKPS) is a tool for knowledge extraction, utilization and dissemination of knowledge. It is grammatical rules based automatic model. It reads the query in natural language, using parsing technique to create the E-R diagram and retrieves the information from the default database for the users. The system may accept text as well as voice query. The AKPS proposed in this paper uses Speech Recognition (SR) technique to convert the voice query to textual query. A database for synonym word checking has been added in the system.

Keywords: Knowledge Management System, Natural Language Query, Speech to Text, AKPS, Rules based KMS, Speech Recognition, Synonyms Database.

1 Introduction

The Knowledge base (KB) system stores the knowledge in the explicit form and works as an effective tool for the extraction, utilization and distribution of knowledge. A good intelligent Information Extraction (IE) tool summarizes all available related information in the form of a summary which is defined as the set of text that is produced from one or more texts and it has a important portion of the information in the original text as in [1]. The important things are collaboration, cooperation and sharing information and knowledge with the objective to give them an added importance through attainment, combination, processing and communication of these elements [2]. The Knowledge Base system (KB) tools refer to Knowledge Management Systems (KMS) is very effective in many domains like hospitality, e-tourism, e-education etc.

There are many organizations accepted Knowledge Management System (KMS) to enhancing their performance and management. Ontology can be used for knowledge representation with respect to each domain and it has its advantages. Firstly it allows reuse, updating and sharing of knowledge between much ontology friendly environments and secondly it has its formal structure which makes it easy to extract knowledge from knowledge repositories as in [3].

Many Knowledge Management System (KMS) use the query-response model in their basic architecture. The KMS tools use databases to store knowledge in explicit form. NLP is a computational method to analyze and represent naturally occurring texts for one or more levels of linguistic study to reach human-like language spreading for a variety of applications as in [4]. NLP relates to human-computer interaction, discusses linguistic coverage issues, and explores the development of natural language widgets. Now days the World Wide Web has significantly improved demand for natural language processing (NLP) and NLP is incorporated into multi user interfaces as in [5]. The AKPS system is a rule based system and there the NLP technique has been used. The user send the query in natural language to the AKPS system and AKPS system uses the NLP technique to interpret the natural language query to the conceptual form of database and create sql query at the final stage using the synonyms database. The synonyms database works on to create the sql-query after retrieving the entries from semantic table. The query may be written in assertive or interrogative sentence. The proposed AKPS system is uses voice and text as a primary source to create queries. The natural language voice is converted to text query using SR techniques. The AKPS system will have the ability to read the voice from the interface and after processing the voice query generates a conceptual data model where Response to the Client Query is generated.

2 Related Works

The Knowledge Management System (KMS) facilitates the organizations to capture the knowledge in corporate repository of information. As in [6], a framework of e-Tourism Services has been proposed wherein it re-uses extracted individual knowledge through knowledge sharing in knowledge environment of the system and thus enhances organizational performance as in [6]. A new model of Knowledge Management System has been designed as a smart help desk to provide Responses to client queries, where the initial Database is dynamically extended through generation of tags.
from client queries in the knowledge environment of the organization as in [7].
A new online algorithm based on NLP system which changes the grammar of the formal meaning representation language and training on additional data has been collected from Amazon’s Mechanical Turk that is an order of magnitude faster and surpasses the state-of-the-art results and it can be improved the results as in [8]. Another interesting task on learning part-of-speech taggers from tag dictionaries is to learn hidden Markov model taggers from incomplete tag dictionaries. The system has taken the MINGREEDY algorithm as a starting point. It has defined a simple HMM emission initialization which takes advantage of the tag dictionary and raw data to capture both the openness of a given tag as in [9]. In natural language processing, the Word sense disambiguation (WSD) works on identifying the appropriate intended meaning of a word in a sentence. A new system introduced in a new WordNet database relation structure which enhances the WSD efficiency of knowledge-based contextual overlap dependent WSD algorithms as in [10]. The natural language based document retrieval systems have the ability to determine questions having the equivalent meanings and allow the optimization of the processing mechanism of questions in related systems. This retrieval system has been developed using Vietnamese questions having the equivalent meanings as in [11]. Extraction of semantic roles from a text contains some essential parts like Recognition of verb (s), recognition of noun phrases and labeling the role of each phrase in the sentence as a semantic argument of verb are general parts of a system. The NLP based system works on Persian sentence and it extracts proto-roles in a Persian sentence exploiting POS tags. The system used the Peykareh as input corpus and applies a rule based approach to extract actor as in [12]. An Unsupervised PCFG induction in NLP works on each sentence potentially refers to one of a small set of possible meanings, like sports casting task. The new PCFG model enhanced the PCFG approach that scales to such problems with highly-ambiguous supervision as in [13]. A new model in NLP stated holistic data-driven technique which generated descriptions of natural-language for videos. The model described the output object and activity detectors with knowledge to select the most probable subject-verb-object triplet for describing a video as in [14].
This paper proposes and discusses the model where user will enter query in natural language using an Interface. The Interface is enabled with the speech recognition engine which gives an added advantage to the user to enter query by the own voice of user. The system will follow the client-server architecture where input in natural language query at first produce the semantic table for the conceptual form of database and this conceptual form will further modify to create sql-query using system attached synonym word database to extract the actual entities and their relationship and form sql-query.

3 Principle

The Interaction of a good system with the database in Natural Language should follow the Alphabets knowledge, Lexicon, Grammar Rules, formation of Word and Rules for Sentence Formation of that Language. For the sentence interpretation, the system should generalize by the domain specific knowledge, general knowledge, and common sense knowledge in respect of knowledge extraction from the knowledge database.

3.1 Automated Knowledge Provider System (AKPS) Principle

The AKPS system model is based on extraction of knowledge to use the query-response model. The AKPS system works on assertive and interrogative sentences. The analysis of the internal structure of particular words has an important intermediate stage and morpheme concatenation is primary meaning of various word forms and word formation processes in this stage as in [15]. The formation of assertive sentence - Noun phrase + Verb phrase + Complement and the formation of interrogative sentence - WH Phrase (attached Auxiliary Verb) + Noun Phrase +Verb Phrase + Complement. A natural language consists of a set of symbols from an alphabet forming a set of grammar rules. Natural Languages such as English, Hindi, Kannada, etc., arise naturally in an unpredicted fashion where as C, C++, Java, and C# etc., are constructed languages in Computer-Programming Languages as in [16]. The word order is the syntactic structure of a language and the words are classified into the parts-of-speech (noun, pronoun, adjective, verb, conjunction, preposition, interjection) and the arrangement of these parts-of-speech in sentences is determined according to the structure the language like English language follows Subject-Verb-Object structure as in [17]. These formations are used to implement the grammar rules in AKPS system. The Flowchart below shows the entire process of AKPS system in Fig.1. The AKPS system receives the queries as input in natural language, tokenizes the query string and creates many rules. At first the query string is validated. The grammar rules are generated at runtime of the main process and grammar rules are matched to the query string rule. Then parsing techniques are automatically applied and the semantic table is created that represents the conceptual form of database and it will be used to generate the sql-query for response from the default database.
The result data will be stored into the resultant database from where the actual response will generate to the users. Sometimes it can be happened that the result of a query is not present in default database then the query will be stored into the temporary database. If the default database is updated then AKPS system will generate the response of pending query automatically.

### 3.2 The Speech Recognition (SR) Interface

The SR technique has been used to develop an Interface where user sends his queries in his own voice in the AKPS.

For any kind speech SR application, there are five operations as follows:
1. Initialization of SR engine.
2. Grammar creation
3. Connect grammar with the SR engine.
4. Mention event notifications to the SR engine.
5. Create SR event handler

The user’s voice is the main input to SR system where speech recognizer treats the voice as an audio stream and converts it to the text format. The speech recognition interface diagram is shown in Figure 2.

### 3.3 The Synonyms Database in AKPS System

The synonyms database has been attached with the AKPS system. The Nouns and verb are extracted by the process of AKPS from the queries in natural language. A Noun is treated as an entity and verb is treated as relationship between two nouns. The synonyms database contains three relations named Entity, Attributes and Relationship. The default database contains a set of values for the relations Entity, Attributes and Relationships. Each of these relations has an attribute called Synonyms. The synonym values should be predefined for each relation in synonyms database. Attribute relation stores each entity name and its attributes name along with corresponding synonyms. Relationship relation defines only the verb and its synonyms. The AKPS system processes the query and creates the conceptual form. Conceptual form is constructed by the Entities (Nouns) and the Verb is used to make the relationship. If the name of the entities derived in the Conceptual form do not match with the Entity Relation name in the default Synonym database or Verb is not the relationship name of two entities in the Conceptual form, then AKPS starts to identify the corresponding synonyms from the Synonyms attribute of Entity/Attribute/Relationship Relations in the default Synonym database. Relations of the synonyms database are given below.

1. **Entity Relation** – Attributes are Entity and Synonyms.

   Table 1: Entity Relation in Synonyms Database

<table>
<thead>
<tr>
<th>ID</th>
<th>Entity</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Universities</td>
<td>Burdwan university</td>
</tr>
<tr>
<td>2</td>
<td>course</td>
<td>courses</td>
</tr>
<tr>
<td>3</td>
<td>department</td>
<td>faculties</td>
</tr>
</tbody>
</table>

2. **Attributes Relation** – Attributes are Entity, Attributes and Synonyms.

   Table 2: Attributes Relation in Synonyms Database

<table>
<thead>
<tr>
<th>ID</th>
<th>Entity</th>
<th>Attributes</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>universities</td>
<td>university_id</td>
<td>university_id</td>
</tr>
<tr>
<td>2</td>
<td>universities</td>
<td>university_name</td>
<td>university</td>
</tr>
</tbody>
</table>
### Table 3: Relationship Table in Synonyms Database

<table>
<thead>
<tr>
<th>ID</th>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>department</td>
<td>department_id</td>
</tr>
<tr>
<td>2</td>
<td>department</td>
<td>department_name</td>
</tr>
<tr>
<td>3</td>
<td>Course</td>
<td>course_id</td>
</tr>
<tr>
<td>4</td>
<td>Course</td>
<td>course_name</td>
</tr>
</tbody>
</table>

3. Relationship Relation – Attributes are Verb, Synonyms.

#### Table 4: Phrases of Assertive Sentence

<table>
<thead>
<tr>
<th>Sl No.</th>
<th>Noun Phrase</th>
<th>Verb Phrase</th>
<th>Complement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Noun</td>
<td>Verb</td>
<td>Noun</td>
</tr>
<tr>
<td>2</td>
<td>Pronoun</td>
<td>Aux + Verb</td>
<td>Determiners + Noun</td>
</tr>
<tr>
<td>3</td>
<td>Determiners + Noun</td>
<td>Aux + Adverb + Verb</td>
<td>Preposition + Noun</td>
</tr>
<tr>
<td>4</td>
<td>Adjective + Noun</td>
<td>Preposition + Verb</td>
<td>Preposition + Determiners + Noun</td>
</tr>
<tr>
<td>5</td>
<td>Determiners + Adjective + Noun</td>
<td>Preposition + Preposition + Adjective + Determiners + Noun</td>
<td></td>
</tr>
</tbody>
</table>

3.4 The Architecture of AKPS System with Interface

The AKPS system receives queries in Natural Language. The queries use English language. The AKPS system has the ability to read the queries in assertive and interrogative sentences. The AKPS system follows some rules for assertive and interrogative sentence.

i. Assertive Sentence formation:

Assertive Sentence - Noun phrase + Verb phrase + Complement. Below the table represents the formation of assertive sentence phrases.

The rules of assertive sentence will be like -

a. Noun + Verb + Noun
b. Pronoun + Aux + Verb + Determiners + Noun
c. Determiners + Noun + Aux + Adverb + Verb + Preposition + Noun.

#### Table 4: Phrases of Assertive Sentence

<table>
<thead>
<tr>
<th>Sl No.</th>
<th>Wh Phrase</th>
<th>Noun Phrase</th>
<th>Verb Phrase</th>
<th>Complement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Wh</td>
<td>Noun</td>
<td>Verb</td>
<td>Noun</td>
</tr>
<tr>
<td>2</td>
<td>Wh + Aux</td>
<td>Determiners + Noun</td>
<td>Preposition + Noun</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Preposition + Determiners + Noun</td>
<td>Preposition + Adjective + Determiners + Noun</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The AKPS system will use the phrases to create the rules to identify the noun, adjective, verb and adverb at the runtime.

An example has been given here which shows the rules formation.

Example: “What are the departments opened by the Burdwan University?”

Sentence formation of this example will be-

Wh Word + Auxiliary Verb + Determiner + Noun + Verb + Preposition + Noun + Noun

The AKPS system uses this formation of assertive and interrogative sentence as a rule at runtime.

The AKPS system reads the Voice or Text using an Interface from users in Natural Language (NL) as a query followed by the system tokenizing the queries. When user speaks a query sentence, the speech engine recognizes the voice and produces the words with the help of the default grammar database. In the Fig. 3, the context diagram shows each component of the interface of the AKPS. The user will enter into the system after login. The user will get an interface into the system where user can send the queries in natural language. The interface has two modes to receive the queries from user either in speech format and text format. User selects speech mode to enter the query in natural language. Spoken words will be received by the interface. The interface process the user voice to detect the word. The SR engine handles the speech and converts it to the text format with the help of a grammar database which is attached with the interface. The grammar database is pre-defined database.
Figure 3: Context diagram of Automated Knowledge Provider System (AKPS) with an Interface

User can enter natural language query using speech mode or in text mode using the keyboards. The request validated and then processed by the AKPS. The system is associated with the three databases. The query is analyzed syntactically based on grammar rules, then parsing technique is applied on resultant data, conceptual form of database is generated and finally, the sql query is generated. The sql query retrieves the knowledge data as a response from the default database. The response will be stored into the result database from where the response will be generated to the client-end. If the requested knowledge data is not present in the default database, then AKPS system will automatically generate the response against the pending request. The Algorithm of the AKPS system with a speech recognition interface has been stated below.

**Process:**
1. User selects the mode in interface to posts Query in Natural Language.
2. If the speech mode selected then user speak the Query in Natural Language in user’s own voice.
3. The voice reads by the interface using speech recognition engine and now apply the algorithm.

**Algorithm of Speech Recognition:**

1. Create the speech recognition engine class instance.
   SpeechRecognitionEngine speechRecognitionEngine = null;
   SpeechRecognitionEngine = createSpeechEngine (“de-DE”);
2. To recognize the speech hook an event.
   speechRecognitionEngine.SpeechRecognized += new
   EventHandler<SpeechRecognizedEventArgs>(engine_SpeechRecognized);
3. Call the Load_Grammar method.
   Load_Grammar();
4. User use the system’s default microphone
   speechRecognitionEngine.SetInputToDefaultAudioDevice();
5. Start listening from user.
   speechRecognitionEngine.RecognizeAsync(RemoteMode.Multiple);
6. Converted voice in text format initialize to the string which is the query.
   Load_Grammar() Method : Call the method “AddWord” under Load_Grammar method. The “AddWord” method add each word from grammar database to a text file.
   AddWord();
   Speech.Recognition.Choices texts = new
   Speech.Recognition.Choices ();
   string[] lines =
   File.ReadAllLines(Environment.CurrentDirectory + "\ex1.txt");
   FOREACH (string line in lines)
   ii. skip if the lines are empty.
   IF line == String.Empty THEN
   continue;
   iii. split the line using “|” character.
   var parts = line.Split(new char[] { ‘|’ });
   iv. Call the Add method to add the text to the known choices of speech engine.
   texts.Add(parts[0]);
   END FOREACH
   GrammarBuilder(texts));
   speechRecognitionEngine.LoadGrammar(wordsList);
   AddWord () Method:
   i. Initialize the variables
   int i, j;
   string ss = “", fi = “";
   string[][] ds1=new string[100];
   string[] split = new string[3];
   FOR i = 0 TO ds1.Count STEP 1
   FOR j = 0 TO 2 STEP 1
   split[j] = ds1[i][j];
   value INCREMENTED BY 1
   END FOR
   Environment.NewLine;
   v. Add each word in text file.
   File.AppendAllText("\ex1.txt", ss);
ss = “”;
value INCREMENTED BY 1
END FOR
i. If user select the text mode then user can type the query in interface directly.
ii. Query is checked for validity. If valid then go for further Processing, if not, then prompt again.
iii. The Query in Natural Language is tokenized. Query (String) = {S1, S2, S3, S4,……….. Sn} Where, S1, S2 are tokens.
iv. Now apply the Algorithm on Token List.

A. Algorithm of main process:-
1. Read Input Statement S.
2. Split Input Statement S into the String Array called STR[n].
3. First check the STR[0] equal to the words- “who”, “which”, “what”….etc.
   IF STR[0]== “who”|| IF STR[0]== “which”|| IF STR[0]== “what” ….. THEN
   CALL Interrogative_Sentence(STR[n]) method
   ELSE
   CALL Assertive_Sentence(STR[n]) method
   END IF
4. Interrogative_Sentence(STR[n]) method:
   i. Create the each Rule and store it on Rules Array.
   ii. Check the STR. length with the each rule length from Rules[n]. If the rule match from Rules[n] then send the selected rule and STR[n] to Parsing method which will either return 0 or 1 to a variable.
   If variable value is 1 then Insert STR[n] in Sentence Table in Database and Insert selected Rules in Rules Table in Database.
5. Assertive_Sentence(STR[n]) method
   i. Create the each Rule and store it on Rules Array.
   ii. Check the STR. length with the each rule length from Rules[n]. If the rule match from Rules[n] then send the selected rule and STR[n] to Parsing method which will either return 0 or 1 to a variable.
   If variable value is 1 then Insert STR[n] in Sentence Table in Database and Insert selected Rules in Rules Table in Database.

6. int Parsing(string Rules, string STR)
   i. Declare Parts of Speech List: WH[n]={"what","which"….}  
      AUX[n]={"is","am","are"….}  
      PRE[n]={"of","to","in","for"….}  
      DET[n]={"a","an","the"….}  
      PRO[n]={"I","you","he"….}  
      UNK[n]={"Noun","Adjective","Verb"….}  
   ii. split Rules to R[n] and STR to S[n].
   Declare Result[0….n] = NULL. Check each word from S[n] with each word from WH[n], AUX[n], DET[n],PRO[n] and UNK[n] and store result in Result Array.
   iii. Initialize the variable value=0

    FOR i=0,j=0, TO Result.length -1 AND
    R.length-1 STEP 1
    IF Result[i] != R[j] THEN
    Value INCREMENTED BY 1
    END IF
    END FOR
   iii. If Value > 0 then the parsing method will return 0 else 1.
7. Select Row one by one from Rules Table and Sentence Table in Database.
   i. Each Row of Rules Table and Sentence Table initialize in SS1 [n] and SS2 [n].
   ii. Each word from SS1 array inserts it into the grammar table as per SS2 array.
8. From Grammar Table select the values from Noun, Verb columns and properly initialize in Semantic Table. Semantic table represents the conceptual form of database.
9. Select values from semantic table and check each word in the synonym database where the actual word is stored with the synonyms.
   i. Initialize the Entity1, Entity2, Relationship values from Semantic Table to string variables entity1, entity2 and verb.
   ii. value of entity1 and entity2 variables checks with synonyms words in Entity Relation, Attributes Relation and value of verb variable checks with synonyms words in Relationship Relation.
   iii. Initialize properly the Filter Table by this three string variables entity1, entity2 and verb.
10. Create the SQL-Query from Filter Table to retrieve data from default database.

4 Methodology & Tools used

The speech recognition engine has been used to implement the speech recognition technique. This SR engine is open source and available in web for the windows platform. The default culture is specified in this SR engine which is used at the time of recognition. It is not always mandatory to use the default culture in speech recognition; there are many other cultures on the basis of many human languages. A runtime language has been attached with the acoustic, language and other data models which help the SR engine to do the SR in a particular human language. The SR engine needs an audio source to recognize the speech. The audio sources may be the wave file, wave streams, and audio streams. The SR Interface is implemented in the visual studio.net integrated development environment. The two most important components needed for the development of SR technique are run-time language and SDK. The runtime language enables the speech recognition for a specific language. Speech platform SDK provides complete set of development tools for implementing SR applications that influence the SR engine. The .net framework manages all the classes from “System.Speech” class library. The AKPS system has been designed using C# at front end and MS-Access at back end as the designed tools.
1. The AKPS system recognizes the speech from user in natural language. Assume the user speaks the interrogative sentence “what are the courses provided by the Burdwan University” in Natural Language.

2. Each spoken word is recognized by the AKPS. As per the Fig. 4 the text box control in interface of AKPS system shows the query string “what are the courses provided by the Burdwan University” in text format. System splits the string and creates token (“Word in sentence”) and check whether first token is “Wh” word or not. The first token is “Wh” word and the “Interrogative Sentence” Method will be called.

3. Rules will be created as per the above algorithm at runtime. Rules will be stored in Rules Array. Again AKPS checks Query string length with the each rule length from Rules Array. If any rule length is matched then insert each token in Sentence Table and insert each word from matched rule in Rules Table in Database.

4. In next step the parsing method will be called. Parsing method will return either 0 or 1 as per the algorithm.

5. Select Row from Rules Table and Sentence Table in MS Access Database. As per the step 7 in algorithm it creates two arrays. First array holds the value for the grammar table and other holds the column name of grammar table in MS Access database. Now arrays are inserting into the Grammar Table as per the step 7 in algorithm.

6. Select Noun, Aux, Verb, and Preposition from Grammar Table and do the step 9 of the main algorithm. The Semantic Table is formed with entities and their relationship after completion of the STEP 9 of the algorithm.

7. conceptual form (E-R Diagram) of Database is formed by the semantic table and it is shown in Fig. 5.

5 Applications of the Model

The proposed Model of AKPS is based on the e-University Information Services. The Application of this model can be used for any other organizational e-services after some extension and modification. The model of AKPS system has a flexible framework based on SR engine. The speech recognition may be used in many types of e-services viz. e-Tourism services, e-hospitality services, and e-banking services in the Web and for mobile applications offering SR.

6 Conclusion and Future Work

The proposed interface of AKPS system is well equipped with the speech recognition technique which enhances the performance of AKPS system in many organizations. The Interface is flexible to the user for their day to day, repetitive queries. The AKPS system can handle a large number of queries in automated
mode and interface gives the opportunity to the user to enter their queries in their own voice. The SR technique uses the grammar table to detect the words which are spoken by the user. The AKPS system processes the queries and creates the conceptual form (E-R Diagram) of database. But AKPS system can only retrieve or extract the text based knowledge data. If knowledge data is an image, then the AKPS system is not been able to retrieve or extract the image data from the knowledge data base. The Future work lies on querying image based knowledge data as well as text base.

References


