A Survey on Question-Answering System

Veronica Naosekpam
Dept of Computer Science Engineering & IT
Assam Don Bosco University, Guwahati, Assam – 781017

Abstract- *Question-Answering system is one of the many domain in the field of Natural Language Processing. Many researches have been going on in this field. In this paper, we explore some of the trending techniques currently used some of the many question-answering systems and review the merits and demerits of each system. Also, some of the architectures used in these systems is being discussed in details. For building a question-answering system, various techniques used are question pairing, part-of-speech tagging, information retrieval, n-gram, etc. The main aim of each of these systems is to find the most probable answer to each question inputted with higher accuracy and prediction. In addition to this, the paper also discuss conversion of natural language question into Structures Query Language (SQL) for information extraction..*

Keywords- information retrieval, n-gram, selection, syntax, semantic, query, parsing.(keywords)

I. INTRODUCTION

Since long time, there has been systems that perform Question-Answering task (Green et al,1963). By 1960s, there were systems implementing the paradigm of the modern question answering system- IR(Information Retrieval) based question answering system and knowledge based question answering system. Recently, systems have been developed to handle large database of question. Most of the system uses TREC (Text Retrieval Conference) corpus to perform the task of answering to the question posted by the user. Different techniques were used to answering the question such as Question/Answer type classification[1], information retrieval, selection of the most probable answer. Database is one of the major source of information where information is stored in a relational model and information can be retrieved using Structured Query Language(SQL). But SQL queries cannot be understood by a user who does not have technical background. Hence, Artificial Intelligence(AI) and Linguistic can be combined together to develop programs for information production in a natural language. NLIDB (Natural Language Interface Database) [2] systems are those that translate the natural language sentences into database query which consists of number of steps for the conversion of the sample question to an SQL query. Thus, it helps to optimize the search result with a more accurate result. The power of surface text pattern for an open domain question answering system [3] help to develop method for learning pattern automatically. On TREC-10 corpus and tagged corpus built using the web, the TREC-10 question set is used to find the answer. The LASSO Question/Answering system [4] consists of three modules-Question Processing Module, Paragraph Indexing Module and the Answer Processing Module. The first module defines keywords which are passed to the next module. For ambiguous question type, the concept of 'focus' was introduced which is word or a sequence of words that disambiguates what the question is looking for. The development of the question-answering system is a complex process that includes the application of a variety of NLP techniques and also the data mining algorithms such as deep and recursive neural networks. One approach is the development of algorithm based on searching the relevant answers in the knowledge base [5]. In the AskMSR Question-Answering System[6], there is dependency of]n data redundancy. And it also predicts the percentage of giving an incorrect answer rather than giving a wrong answer. Some of the linguistic resources used are part-of-speech tagging, parsing, named entity, semantic relation, dictionaries, WordNet etc. Here, the source is the Web which serves as a gigantic data source. Since the data source is the web, this system seems to give more accurate result than the others system discussed in this paper

II. DIFFERENT APPROACHES OF THE QUESTION ANSWERING SYSTEMS

The Question-Answering system using natural language processing with NLIDB(Natural Language Interface for Database) approach uses as user input the question written in natural language and using NLIDB , it is converted into an SQL(Structured Query Language) for further processing of the answer to the question. Finally the SQL query question is passed to the knowledge based for information retrieval. The user question is pre-process and the pre-processed question is passed to the lexical analyzer which convert the question into SQL query. The problem is visualized in Figure1 below:

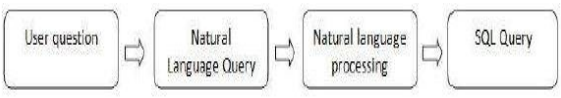

Fig 1

The preprocessing of the user question includes tokenizing, lower case conversion, escape word remover, removing ambiguous attributes. Figure 2 shows the system implementation.

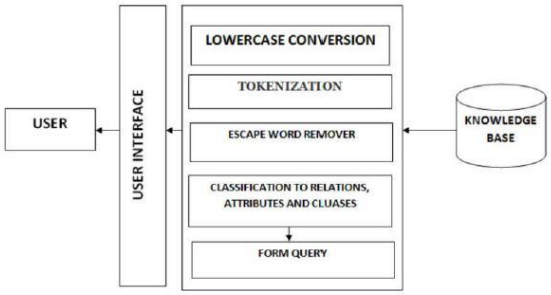

Figure 2

Production rules are primarily pre-described set of rules and behaviour. Hence the executed production rule converts the user statement into SQL query and then is fired on database. Production rule consist of two parts i.e. a precondition and action.

The purposed of this work is to handle the challenges in Natural language processing and make it more robust and flexible for all kinds of queries respective to its domain. However, it suffers from certain drawbacks since it finds it difficult to handle difficult and challenging questions with more accuracy.

The IBM's Question-Answering System consists of four sub components:- Question/Answer Type Classification, Query Expansion/ Information Retrieval, Named Entity Marking and Answer Selection. The system architecture is shown in Figure 3.

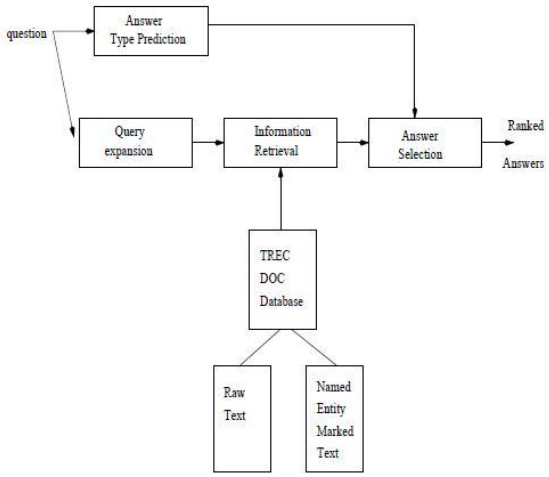

Fig 3

In this system, the question is input whose answer is classifies as one of the named entity classes. Also, the question is passed to the information retrieval (IR) engine for query expansion and document retrieval. This engine looks at the database and selects the best possible answer to the question using the Maximum entropy modeling (Della Pietra et al. 1995). The features of this model were n-grams of words (that is to be adjacent) and bag of words where position is not important. The information retrieval module search the database for matching passages of text, containing relevant information. It uses two passes. The first pass searches an encyclopedia database. The highest scoring passages were the used to create the expanded queries applied to the TREC passages. This system uses MUC marked up classes for named entity annotation. In the answer selection step, each retrieved passage is split into sentences, the distances such as matching words, mis-matching words, dispersion etc are computed and they are weighted a score and the Mean Reciprocal Rank (MRR) of the highest ranking passage containing the answer is calculated. The answer with the highest MRR is accepted as the answer. This approach is better than the previous mentioned work since it utilizes the maximum entropy features.

Parallely, a learning surface text pattern for a question answering system[3] introduces the concept where the system automatically learns regular expression from the web for given types of question. This approach uses machine learning technique of bootstrapping to build a large tagged corpus which initially contains only a few set of QA pairs. The assumption made is that each sentence is made up of simple word sequence and repeated word orderings as evidence for useful answer phrase. Suffix trees are used for taking out substrings of optimal length. Lastly, the pattern learnt by the system is test on new unseen questions from the TREC-10 set and the result is evaluated. Its uses a pattern learning algorithm and an algorithm for calculation the precision of each pattern. However, it suffers from certain drawbacks since no external knowledge has been added to the patterns. Also, the pattern cannot handle long distance dependencies. There is a need to integrate output of the web and the TREC corpus.

A better approach is the LASSO Q/A system, which uses a combination of syntactic and semantics techniques. The architecture of LASSO is shown below in Figure 4.

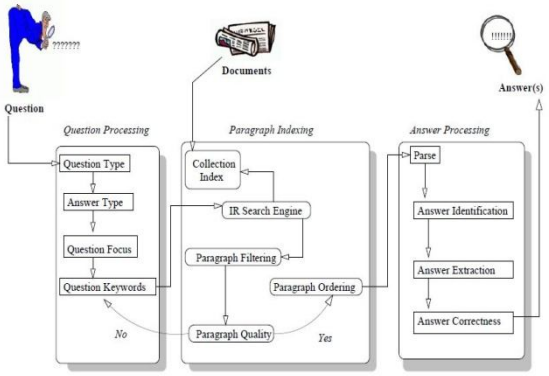

Fig 4

The Question processing module automatically find the question type, answer type and the question focus and transform the question into queries. Further, this model also indentifies the keywords which are then passed to the Paragraph indexing module. The extraction of keywords uses a set of ordered heuristics such as *Keyword-Heuristic 1* which states that "Whenever quoted expressions recognized in a question, all non-stop words of the quotation became keywords", etc. The IR Engine for LASSO is related to the Zprise IRsearch engine . The Boolean indexing is opted since it increases the recall at the expense of precision. Steps involved in index creation are normalization of the SGML tags, eliminate extraneous characters, identify the words within each documents, stem the words using Porter Stemming[7], calculate the local and global weights and create and inverted dictionary file. The next step is the paragraph filtering and the ordering of the paragraphs using the *same _word_sequence_score*, largest *Distance_score* and smallest *Missing_keyword_score*. The Answer Processing Module extracts the answer from the paragraph containing the keywords. The performance evaluation for this approach is based on *accuracy*. And the accuracy NIST score for short answer and long answer is found out to be 55.5% and 64.5% which is quite a good improvement. However, the processing time per question is approximately 61sec which of the total is dominated by the answer extraction time.

We know that it is very necessary to generate base of knowledge to develop a QA-system. The knowledge base can be local (books, scientific papers placed in local repository) and based on web technologies. First it is necessary to allocate so-called knowledge from a source of potentially useful information in the assembled list. In this Question-Answering system approach,

by knowledge here we mean a Text phrase that includes brief and specific information about any fact.
The search for the answer to the test question is proposed to realize using the following Algorithm:

1. Let we have some question Q and a limited set of n_A answers $A=\{a_j, j=1 \text{ to } n_A\}$ to choose the right one.

2. For each question Q, n_A phrases are formed: it is a combination of the question and the j -th answer from a set A .

3. Now it is necessary to evaluate the truth degree of each pair based on the available base of knowledge among all pairs for (Q,A_j) for j-1 to n_A.

4. Let we have n_K knowledge and n_A variants of answers for each question Q. The matching level (a measure of inclusion) for a pair (Q,aj) and some knowledge K is proposed to be present with the formula below:

$$d((Q,a),K)=[d'(Q,K)+ d'(a_j,K)]\cdot (d'(Q,K)> 0)\cdot (d'(a_j,K)> 0) .$$

where $d'(Q,K)$ and $d'(a_j, K)$, K is the measure of the inclusion of question $Q$and answer a_j into the knowledge K .

5. The answer which is considered as correct corresponds to the maximum value of a inclusion measure among the whole knowledge from the base of knowledge .

The result of the quality of the algorithm using three textbooks and a sample as base of knowledge is found out to be 55.7%.

The final and the most efficient approach for the Question-Answering system is the AskMSR which differs from other system in its dependency on data redundancy and prediction the likeliness of giving incorrect answer. The system architecture is given below in Figure 5.

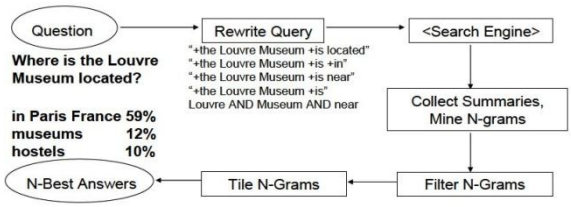

Fig 5

The first step here is the query reformulation where a question a is given, the system generates a number of weighted rewrite strings. Example, "*When was the aeroplane invented?"* is rewritten as *"The aeroplane was invented"*. Each rewrite is formulated as a search engine query and sent to the search engine where page summaries are collected and analysed. Text is processed in accordance with the patterns specified by the rewrites. Unigrams, bigrams and trigrams are

extracted and subsequently scored according to the weight of the query rewrite that retrieved it. Next, each n-gram is filtered and assigned one of the seven question type such as *what-question, who-question* , etc. After this step, the last step is the application of the N-gram tiling algorithm which merges similar answers together. The algorithm proceeds greedily from top-scoring candidate. In this approach, the percentage error of the unknown problem is just 5%. One typical feature of this system is that it built a decision tree to ry to predict whether the system will answer correctly or not based on a set of features extracted from the question string.

III. CONCLUSION AND FUTURE WORK

This paper summarized the various approaches used for Question-Answering System in Natural Language Processing. Some of the tool such as "Learning Surface Text Pattern" can be used for multilingual QA. The different techniques used here can be extended beyond short QA. In order to address question of higher classes, we need real time knowledge acquisition and classification from various domains. Further work can utilize the maximum entropy features in the answer selection process which will lead to the system completely trainable from examples. Goal of the SQL approach is to increase the accuracy and making the system robust. Using of complex queries using HAVING and GROUPBY will further enhance the filtering of the answers.

IV. ACKNOWLEDGEMENT

I consider this as a privilege to express a few words of gratitude to all those who guided and inspired me for the successful completion of this work, especially Mr. Arup Baruah, Assistant Professor, Department of Computer Science, Assam Don Bosco University, Assam.

REFERENCES

[1] IBM's Statistical Question Answering System, Abraham Ittycheriah, Martin Franz, Wei-Jung Zhu, Adwait Ratnaparkhi from IBM and Richard J.Mammone, Dept. of Electrical Engineering, Rutgers University,Piscataway, NJ,USA.

[2] Question-answering system using NLP with NLIDB Approach, Prof. Pooja Malhotra, Yash Kapadia, Krishna Saboo and Ankita Sharda , K.J. Somaiya College of Engineering ,International Journal of Current Research Vol. 9, Issue, 09, pp.57575-57577, September, 2017.

[3] Learning Surface Text Patterns for a Question Answering System, Deepak Ravichandran and Eduard Hovy, Information Sciences Institute ,University of Southern California, roceeding of the 40th annual meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 41-47.

[4] The Structure and Performance of an Open Domain Question-Answering System, Dan Moldovan, Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, Vasile Rus, Department of Computer Science and Engineering, Southern Methodist University, Dalas Texas.

[5] Question-answering system ,A A Stupina, E A Zhukov, S N Ezhemanskaya, M V Karaseva, L N Korpacheva ,Siberian State Aerospace University,Krasnoyarsk, 660037, Russia, IOP Conf. Series: Materials Science and Engineering 155 (2016) 012024 doi:10.1088/1757-899X/155/1/012024.

[6] An Analysis of the AskMSR Question-Answering System Eric Brill, Susan Dumais and Michele Banko, Microsoft Research ,One Microsoft Way Redmond,{brill,sdumais,mbanko}@microsoft.com, Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, July 2002, pp. 257-264.

[7] Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition. Daniel Jurafsky & James H. Martin.

[8] http://www.packtpub.com/mapt/book/big_data_and_business_intelligence/

[9]http://www.facweb.iitkgp.ernet.in/~sudeshna/courses/nlp07/lec24-QA.pdf