IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 3, Issue 4, Aug-Sept, 2015
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

Can A Question Answering System Predict? - A Survey

Susan Aisoo Thomas[1], Lekshmy P Chandran [2], Bindu.M.S[3]

[1,2]Computer Science Dept., College of Engineering, Kidangoor, Kottayam, India
[3]Computer Science, STAS, Pullarikkunnu, Kottayam, India

Abstract— Early days, information was available only in textbooks. It was a hectic job to find out any required details from these books. But now everything is at our fingertips- just a click will give you a lots of information on any topic. In a Question Answering system, the user submits a question and waits for the answer as the response. If the system is capable of predicting the user's future interest as the next question, its performance will improve greatly. This paper is a discussion on various Question Answering (QA) Systems which can predict the future requirement.QA Systems uses query clustering algorithms, association rules and query expansion for question prediction.

Keywords-QA System, question prediction, answer retrieval, evaluation

I. INTRODUCTION

Artificial intelligence (AI) is the intelligence exhibited by machines or software. It is also the name of the academic field of study which studies how to create computers and computer software that are capable of intelligent behavior. The central problems (or goals) of AI research include reasoning, knowledge, planning, learning, natural language processing (communication), perception and the ability to move and manipulate objects.

Natural Language Processing (NLP) is a branch of Artificial Intelligence (AI) that deals with analyzing, understanding and generating natural languages that humans use.NLP gives machines the ability to read and understand the human language.

Question answering, an important field of NLP, enables users to ask questions in natural language and get precise answers instead of long list of documents usually returned by search engines.

In a Question Answering system, the user submits a question and waits for the answer as the response. If the system is capable of predicting the user's future interest as the next question, its performance will improve greatly. This paper is a survey on such QA systems for various languages and various domains.

This paper is divided into 5 modules. Module II explains approaches to QA Systems, module III describes QA System architecture, module IV is the literature survey and module V concludes the paper.

II. APPROACHES TO QA SYSTEMS

A QA system can be classified based on various factors such as domain of QA, language used for input query and the retrieved answer, types of question asked, kind of retrieved answer, levels of linguistics applied to the documents in the corpus and answer resources. Accordingly, QA systems falls into Open Domain vs. Closed Domain, Monolingual vs. Multilingual, Factoid vs. Non-factoid, Document vs. Answer Retrieval, Deep vs. Shallow, and Database vs. FAQ vs. Web QA.

Open domain Question Answering System is an area of Natural Language Processing research, aimed at providing human users with a convenient and natural interface for accessing information. It deals with questions about nearly everything [9]. These systems usually have much more data available from which to extract the answer. ASKJEEVES is the most well-known open domain QA System.

Closed domain Question Answering Systems deal with questions under a specific domain (for example, medicine or automotive maintenance), and can be seen as an easier task because NLP systems can exploit domain-specific knowledge frequently formalized in ontologies. Closed domain refers to a situation where only limited types of questions are accepted. In a closed domain QA, correct answers to a question may often be found in only very few documents since the system does not have large retrieval set. Green's BASEBALL system is a restricted domain QA System that only answers questions about the US baseball league over a period of one year.

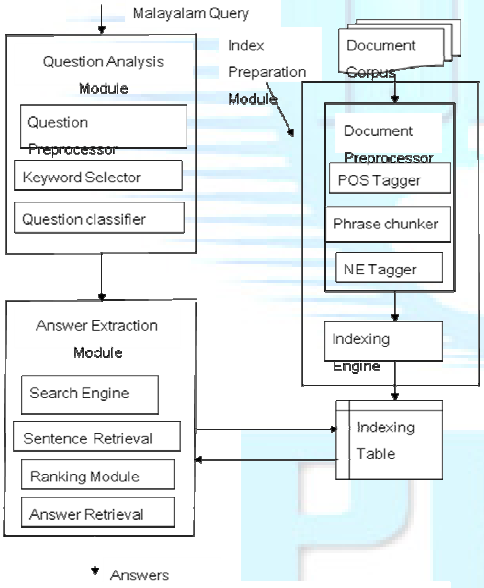
In a QA System, questions and answers are given in natural languages. Hence QA Systems can be characterized

by the source (question) and the target (answer) languages. Based on these languages QA Systems (QAS) are named as monolingual, multilingual or cross-lingual systems

Question types can also be used to categorize QAS. Different question types may require different strategies to deal with them. There are three question types– Factoid, List and Description.

Another classification is Shallow or Deep Systems, based on the level of processing applied to the questions and documents. Some Shallow QAS use keyword based techniques to locate interesting passages and sentences from the retrieved documents based on the answer type. Ranking is then done based on syntactic features such as word order, location or similarity to query. But question reformulation is not sufficient for Deep QA; more sophisticated syntactic, semantic, and contextual processing must be performed to extract or construct the answer.

III. QA SYSTEM ARCHITECTURE



System Architecture Design

Indexing Module processes all the documents in the corpus and prepares a Named Entity (NE)-based index. When a query is entered by the user, the Question Analysis Module analyses the query sentence and determines its type. Also it identifies keywords and other significant words. Keywords are useful in finding the entity type of the expected answer and significant words help in determining the final answer. These details are passed to the Answer Extraction Module. Answer Extraction Module uses this information to search through the index and extracts the most relevant answer from the corpus.

A. Document Preprocessor

Document preprocessing stage preprocesses the documents with the help of the tools Compound Word Splitter, Part-of-Speech (POS) Tagger, Phrase Chunker and NE Tagger.

1) Compound Word Splitter

Compound word is a word composed of two or more words either in the closed form, hyphenated form or in an open form Owing to this compounding or agglutinative nature 80-85% words in Malayalam documents are compound words. Hence a compound word splitter is essential to split these words into their components to deduce the word meaning.

2) POS Tagger

POS Tagging, also called grammatical tagging, is a principal issue in Natural Language Processing. The purpose of this task is to assign part-of-speech or other lexical class markers to each and every word in a document. Since most of the words in a Malayalam document are compound words decomposition of these words into their constituents is extremely necessary for finalizing their POS tag. Sometimes more than one morphological analysis and hence more than one POS may occur for a single word. A correct resolution of this kind of ambiguity for each occurrence of the word is crucial in QA Systems. A large percentage of words also show ambiguity regarding lexical category.

3) Phrase Chunker

This module in the preprocessing stage contains a clause identifier, a clause separator and a phrase tagger. Clause identifier identifies and separates the clauses from the sentences using a rule base. Then the phrase separator separates the phrases from each of these clauses and phrase tagger attaches appropriate phrase tags to them.

4) NE Tagger

Named Entities are words or word sequences which usually cannot be found in common dictionaries and yet encapsulate important information that can be useful for the semantic interpretation of texts. A Malayalam NE Tagger takes a POS tagged document and produces NE tagged document. NE Tagger uses NE marker, NE identifier, NE classifier, and NE Disambiguator for properly determining the NE tags.

IV. LITERATURE SURVEY

Authors of paper [1] present a strategy that aims to extract and rank predicted answers from the web based on the eigenvalues of a specially designed matrix. This matrix models the strength of the syntactic relations between words by means of the frequency of their relative positions in sentences extracted from web snippets. In this system the user enters via some input device an NL query which is further passed to a search engine. The search engine returns a ranked list of document links together with a snippet for each document. The best N–snippets are passed to the answer prediction component. This component extracts from all snippets the best predicted answer strings. A predicted answer string is a substring extracted from the snippets for which a high semantic similarity to the question has been determined. Unlike Pseudo Relevance Feedback (PRF), the predicted answers are ranked and the M–best are submitted to the answer extraction component. It further splits the predicted answer strings into smaller units which might correspond to exact answer strings. The answer extraction component uses the NL user question in order to determine the Expected Answer Type (EAT). Since the goal is to be as language independent as possible and focus is to evaluate the quality of the answer prediction strategy, this step resembles any traditional system based on pattern matching and lexical databases. They assess the rank of predicted answers by extracting answer candidates for three different kinds of questions. Due to the low dependence upon a particular language, they also apply this strategy to questions from four different languages: English, German, Spanish, and Portuguese. The system is evaluated with three different types of questions from four languages, obtaining a combined MRR=0.52 for the respective subset of the CLEF–2004 data set.

Rosy Madaan, A.K. Sharma, and Ashutosh Dixit presents a paper which predicts users' future questions based on the current interaction records of the user with the system [2]. Their current interactions with the system show what they are interested in. These interaction records are maintained in the form of questions log from which the user sessions are extracted. Based on the user sessions, the system predicts the next question for which the user may become interested in near future. A sample questions log is selected for the purpose of performing experiments. The model of association rule mining is applied to predict the future question of the user. In this proposed system the user enters his question on the interface of the QA system. The user asks questions from the QA system by entering a question on the QA Interface. This question is classified by the question classifier module. The module classifies the questions according to their question type and then converts the rest of the question into query. For the query a search is done by the searcher module. Along with performing search for the answer(s) to the questions, these users' questions are stored in the questions log along with the IP address from which the question has been asked, the Question ID (QID), date and time of questioning. The interactions between the user and the system are maintained in a file termed as question log.

In [3] authors have presented a method to extract user Interests from search query logs. The global representation is composed of a semantic taxonomy of query log terms together with a function that evaluates the semantic distance between the query terms. The distance takes into account a new property related to the abstraction level of terms. In addition to that, they proposed a fast algorithm that enables to extract the user's topic of interest in form of clusters of query terms. Such precious information is an input for several applications. The whole system constitutes a two step process. The first step aims to produce a semantically enhanced global reference for the whole log while the second step considers this global reference as a platform on which the user interests are identified. They defined the global reference as a data structure organizing the query terms in a taxonomy equipped with a semantic distance function. They presented an approach to construct this taxonomy over the terms used in keyword search logs by means of the WordNet lexical database of English terms. The construction of the taxonomy is based on two principal aspects. The first is the hierarchical relation between the terms established by the hypernymy and generalization/specialization relations ("is a"). This kind of relation enables to sort the terms into different levels of abstractions and organize them in a tree structure. The second aspect is the semantic distance between two terms connected in a IS-A relation; it represents the weight of the Edges of the tree obtained previously. Here, authors introduce a new weighting function that takes into account the abstraction level of terms. In fact, two terms in the bottom of the hierarchy are considered to be closer than two terms situated at the top of the hierarchy. Furthermore, the distance function is generalized to take into account every couple of terms in addition to those that are directly in a IS-A relation. After building the query terms taxonomy, a clustering algorithm is applied, which extracts users' interests in the form of clusters. The algorithm is based on a threshold that enables to control its precision and to adapt the results according to the target application. This approach was evaluated using real-world Web logs from the AOL search engine.

Web search engines became one of the most popular services available on the Web. Despite the recent advances on the technology of the search engines there are still many situations where the user is contemplated with non-relevant answers. One of the great challenges faced by search engines is the difficulty in uncovering an exact description of the user need, since users usually submit very short and imprecise query. A popular solution to help the users in the task of specifying their information needs is to use relevance feedback techniques. These techniques improve the interactivity of the system by allowing users to inform about the relevance of answers given to their initial query. The feedback information is used to refine the initial query and get a better specification of the user needs. Work in [4] presents a method for automatic generate suggestions of related queries submitted toWeb search engines. The method extracts information from the log of past submitted queries to search engines using algorithms for mining association rules. Experimental results were performed on a log containing more than 2.3 million queries submitted to a commercial searching engine giving correct suggestions in 90.5% of the top 5 suggestions presented for common queries extracted from a real log.

Zang and Nasraoui [5] presents a simple and intuitive method for mining search engine query logs to get fast query recommendations on a large scale industrial strength search engine. In order to get a more comprehensive solution, they combined two methods together. To evaluate the hybrid method, they used one hundred day worth query logs from SINA' search engine to do off-line mining. Then analyzed three independent editors evaluations on a query test set. Based on their judgment, the method was found to be effective for finding related queries, despite its simplicity.

Paper [6] is concerned with actively predicting search intent from user browsing behavior data. In recent years, great attention has been paid to predicting user search intent.

However, the prediction was mostly passive because it was performed only after users submitted their queries to search engines. It is not considered why users issued these queries, and what triggered their information needs. According to their study, many information needs of users were actually triggered by what they have browsed. That is, after reading a page, if a user found something interesting or unclear, he/she might have the intent to obtain further information and accordingly formulate a search query. Actively predicting such search intent can benefit both search engines and their users. In this paper, authors proposed a series of technologies to fulfil this task. First, extract all the queries that users issued after reading a given page from user browsing behaviour data. Second, learn a model to effectively rank these queries according to their likelihoods of being triggered by the page. Third, since search intents can be quite diverse even if triggered by the same page, an optimization algorithm is proposed to diversify the ranked list of queries obtained in the second step, and then suggest the list to users. The system is tested on large-scale user browsing behaviour data obtained from a commercial search engine. The experimental results have shown that this approach can predict meaningful queries for a given page, and the search performance for these queries can be significantly improved by using the triggering page as contextual information.

Users' search tasks vary a lot from a simple known-item search to very complex exploratory search. In known-item search, a user has a well-defined information need and can generally formulate an effective query and thus the current search engines often work very well. In exploratory search, however, the information need is often complex and vague, and the goal of search is mainly to gather and study information about some topic. Thus a user generally does not know well about the information to be found in exploratory search (which is the reason why the user needs to initiate the search in the first place). As a result, it is often difficult for a user to formulate effective queries in exploratory search, and the user has to reformulate queries many times in a trial-and-error manner. Querying alone is often insufficient to support exploratory search well due to the difficulty in formulating good queries. When a user is unable to formulate effective queries, browsing would be intuitively very useful because it enables a user to navigate into relevant information without formulating a query. Unfortunately, with the current search engines, browsing is mostly through following static hyperlinks. This is very restrictive and would not allow a user to go very far in the information space. In paper [7], authors propose to leverage search logs to allow a user to browse beyond hyperlinks with a multi-resolution topic map constructed based on search logs. Specifically, they treat search logs as “footprints” left by previous users in the information space and build a multi-resolution topic map to semantically capture and organize them in multiple granularities. Such a topic map can support a user to zoom in, zoom out, and navigate horizontally over the information space, and thus provide flexible and effective browsing capabilities for end users. To test the effectiveness of the proposed methods of supporting browsing, authors used real search logs and a commercial search engine to implement their proposed methods. The experimental results show that the proposed topic map is effective to support browsing beyond hyperlinks.

D. Gupta, A.Puniya, and K.K.Bhatia [8] proposed a predication model for the oncoming queries on the web and a neural network approach was used for training which provides relevant result in less time. This model helps to predict the oncoming queries for a particular domain so that searching of documents becomes more efficient in terms of time complexity. Training and testing the artificial neural network for user queries based on frequency and PMI (Point wise mutual Information) value. Finally 25 queries from all the domains have been selected for testing and training the neural network. The queries now have been triggered at Google search browser and frequency of these queries has been kept in the database for further processing. All the queries are further broken into the individual keywords and the frequency of individual keywords count has also been taken in account. The PMI (Point Wise mutual Information) of each query has been calculated, which basically defines the maximum probability of the event. In the neural network each queries identified by their PMI value. In the neural network the PMI values are taken as inputs among twenty five queries. Twenty queries have been used for training the neural network and rest of all queries has been taken as testing inputs to test the efficiency of the neural network. Due to the usage of back propagation algorithm (supervised learning algorithm) the target values has been required. The target values have been tagged as 0,1,2,3 on the basis of PMI values of training dataset (maximum the PMI value tagged). Once the neural network [27 and 28] has been trained by the given training data, it can be used further for mining the large data sets.

This last step shows oncoming query for next user. This proposed model has prediction factor which is easier to search to next query for the user.

The crawler [08, 09, 11 and 25] maintains a list of unvisited URLs called the frontier. The list is initialized with seed URLs which may be provided by a user or another program. Crawler crawls all web pages stored in the repository. Indexer indexes all the keywords stored in the local repository. The user sends a query through user interface and query processor processes the query and identifies the domain name. Find the PMI value of each query. The neural network is the tool which learns using some rules and conditions invented by incoming queries and work for oncoming queries. In the present work neural network is being used for prediction of the oncoming data on the bases of incoming queries.

Paper [9] predicts users' future queries and URL clicks based on their current access behaviors and global users' query logs. Authors explore various features from queries and clicked URLs in the users' current search sessions, select similar intents from query logs, and use them for prediction. Because of an intent shift problem in search sessions, this paper discusses which actions have more effects on the prediction, what representations are more suitable to represent users' intents, how the intent similarity is measured, and how the retrieved similar intents affect the prediction. MSN Search Query Log excerpt (RFP 2006 dataset) is taken as an experimental corpus. Three methods and the back-off models are presented.

G.Dupret and M.Mendoza presented a method to help a user redefine a query based on past users experience, namely the click-through data as recorded by a search engine. It takes into account the co-occurrence of documents in individual query sessions. It is also particularly simple to implement [10]. Many identical queries can represent different user information needs. Depending on the topic the user has in mind, he will tend to select a particular sub-group of documents. Consequently, the set of selections in a session reflects a sub-topic of the original query. The authors tried to assess the existing correlations between the documents selected during sessions of a same query, create clusters and identify queries relevant to each cluster. Candidate queries for the set of documents selected during a session are searched, and if a significant proportion of sessions points to a given query, it is recommended.

Three methods are identified for evaluating query recommendations in the Literature. The first consists in asking a group of volunteers to test the system. Results are then compiled and conclusion drawn in favor of one method or another. The second method consists in modifying an existing search engine and in proposing the query recommendations to regular users. Finally, the authors opted to analyze their results in detail to evaluate whether they make sense.

All available literature is aimed at predicting the next query of the user and none of the above discussed approaches work for the prediction of questions that a user may pose next to the QA system. So, for building an efficient QA system, it becomes necessary to look for the approach that aims at predicting the next question that the user may pose to the QA system.

IV. CONCLUSION

When a user poses a question to the QA, the system responds with the answer to the question. A search session of a user involves a set of questions that have been asked by the user. Along with responding the user with answer to his question, the system also predicts the next question in which the user is interested in. This survey paper describes the different question answering approaches and different types

of question answering system. We also discussed about the QA systems that have prediction capability.

ACKNOWLEDGMENT

We are extremely thankful to the help, co-operation and support provided by Mrs. Anitha R, Head of the Computer Science Department, College of Engineering, and Kidangoor.

I also express my sincere gratitude to Project Coordinator Mr.Anoop Varkey, Assistant professor College of Engineering, kidangoor.

I also thank my Guide Lakshmy P Chandran, Assistant Professor College Of Engineering, Kidangoor.

REFERENCES

[1]Alejandro Figueroa, and Guernt Neumann, "Language Independent Answer Prediction from the Web", LNAI 4139, pp 423-434, 2006, Springer-Verlog

[2] Rosy Madaan, A.K. Sharma, and Ashutosh Dixit," A Data Mining Approach to Predict Users' Next Question in QA System", IEEE, pp211-215

[3] L. Limam, D. Coquil, H. Kosch, and L. Brunie, "Extracting User Interests from Search Query Logs: A Clustering Approach," DEXA '10 Proceedings of the 2010 Workshops on Database and Expert Systems Applications, 2010.

[4] B. M. Fonseca, P. B. Golgher, E. S. de Moura,, and N. Ziviani, "Using Association Rules to Discover Search Engines Related Queries," in Proceedings of the First Conference on Latin American Web Congress, pp. 66–71, 2003.

[5] Z. Zhang, and O. Nasraoui, "Mining Search Engine Query Logs for Query Recommendation," in Proceedings of the 15th international conference on World Wide Web, pp. 1039–1040, 2006.

[6] Z. Cheng, B. Gao, and T. Liu, "Actively Predicting Diverse Search Intent from User Browsing Behaviors," in Proceedings of the 19th international conference on World wide web, pp. 221–230, 2010.

[7] X. Wang, B. Tan, A. Shakery, and C. Zhai, "Beyond Hyperlinks: Organizing Information Footprints in Search Logs to Support Effective Browsing," in Proceeding of the 18th ACM conference on Information and knowledge management, pp. 1237–1246, 2009.

[8] D. Gupta, A. Puniya, and K.K. Bhatia, "Prediction of the Query of the Search Engine using Backpropogation Algorithm," IJCSE, 2011.

[9] K.H Lin, "Predicting Next Search Actions with Search Engine Query Logs," Web Intelligence and Intelligent Agent Technology (WI-IAT), IEEE/WIC/ACM International Conference on (Volume: 1), 2011.

[10] G. Dupret , and M. Mendoza, "Recommending Better Queries Based on Click-Through Data," LNCS, Springer, 2005.

[11] R. Mudgal, R. Madaan, A.K. Sharma, and A. Dixit, "A Novel Architecture for Question Classification Based Indexing Scheme for Efficient Question Aswering," International Journal of Computer Engineering & Applications (IJCEA), ISSN: 2321-3469, Volume-2, Issue-2, June 2013.