Research on Interactive dialogues Question Answering System

Kai Pun, Pokman Leong, Hoifong Mak, Fai Wong, Yiping Li
Faculty of Science and Technology of University of Macao (Macao S.A.R)
{da11240, da11349, da11362, derekfw, ypli}@umac.mo

Abstract

Question Answering is a type of information retrieval. Then, Speech Question Answering is a kind of information retrieval which requires complex natural language processing techniques together with the speech recognition techniques. Most probably, QA system is used at web site by giving certain questions as the input and the web server will try to retrieve the answer from the database. Besides, there is another type of informational retrieve system called telephony IVR (Interactive Voice Recognition) system which is very commonly used. Most of the question answering system which is probably used is somehow a unidirectional communication with the user, that is, the system is not able enough to have interaction with the users, however, there is a trend of developing a bidirectional question answering system with some interactive dialogues as the importance of artificial intelligence is running up. Here, our research is following this trend by using speech. Here, our focus will be on researching this field and let the East Asian Games (EAG) be the domain of our research. That is, the spoken question will be recognized by our speech engine and our system will extract a correct answer, or gives some other communicative dialogues if the question is out of our domain, in response to the questions spoken by the user. There are two main parts in our research: one is QA system and the other one is the speech recognition development.

Introduction

This paper describes a study which is on the field of QA system using the technique of speech recognition. In the part of QA system, we put our focus on the analysis of the structure and the components of the implementation of the system, and also what the principle is as well as what and how the techniques will be used in it. For the other part, which is the part of development of speech recognition, we will analyze the speech engine and discuss some of the difficulties and restrictions we have met, for example, the ambiguity like uttering, different words but with same pronunciation, low accuracy when dealing with large amount of words and so on.

In fact, question answering system can be divided into two specific parts of closed-domain and open-domain[1]. The difference between them is that the previous deals with questions under specific domain question answering and therefore for the knowledge about the questions can also be specific and whereas for the open-domain question answering deals with almost all kinds of questions and therefore the data from which the answer extracted will be much larger. In our research, we have chosen to use the closed-domain specific question answering system and our focus is in the field of East Asian Games (EAG) which is held in Macau in the coming year. By using this, we can increase the accuracy of the speech recognition and decrease the ambiguity since the amount of the words in the database used is greatly reduced.

The paper is outlined as the following: the next section is focused on the analysis of the QA system Architecture with speech recognition engine. After this, we will discuss how we make use of the IBM speech engine to work with our system. Then, we will describe how we analysis the question given, how to deal with the matching and selecting the answer from the database. After all, we will talk about our difficulties and obstacle met and also our future work with this system. Finally, we will give the conclusion of the paper.

Conventional Question Answering System (Unidirectional QA System)

Unidirectional QA System means that users only give questions and the system will retrieve answer from the knowledge. Just like the question answering system which is most probably used on the web, most of them required users to input a question by text and answer will be retrieved. For another type of unidirectional QA System with speech, like IVR (Interactive Voice Recognition) system, this type of system uses the pre-recorded message to guide the user to get reach of the answer by letting them press keypad to submit their choice accordingly. Lack of communication is the most important shortcoming of such kind of QA System. Most probably, what the user needs to do to deal with this kind of QA system is to enter the question he/she would like to ask. Since the system is not able to communicate with the user, it will just retrieve the answer from the knowledgebase. However, the question entered may not be within the knowledgebase of the system or it may not be clear enough for the system to retrieve the answer from

the knowledgebase correctly or successfully. These inabilities may cause low efficiency when retrieving the answer since it may take more time in processing wrong question or low accuracy in for the answer retrieved. Here, we would like to have the research in improving these shortcomings of conventional system.

Interactive Dialogues QA System

For us, the human being, we usually actively confirm with the user that the question he/she has entered is correct or not, or we may try to get some keywords from the question he/she given in order to organize another question which is somehow related to the question input by the user and try confirm with the user with that question. Such kind of communication of course required many dialogues with the user, because of this, there are several problems in developing the system.

Issues that to be addressed in developing IDQA System

Before introducing the architecture of the IDAQ System, we must first identify the problems which we must meet during the development of the system.

Speech Recognition Accuracy – The accuracy of speech recognition is not quite satisfied. Without filtering, the accuracy can be only 20% empirically. Although we have design a set of grammar to deal with the problem of low accuracy, this problem can only be lightened but could not be solved completely. There are actually lots of programming work to deal with this problem, like if the question could not be fully recognized, the system must be able to interact with the user. The system should be able to interact with the user in order to get more information so as to retrieve the answer for the user.

Human speech – Human has numerous ways to ask the same question with the same meaning. The system should understand most possible ways in asking a question. For example, people have at least 10 ways in asking "How to go somewhere". For a polite person, many **courteous words** may inside the question, for example, "please", "I would like to…", "thank you", 'Excuse me". In human communications, those words are needed and are approved; however, for machine-based system, those words are somehow **obstacles**. Speech QA System needs to accept and filter out those words in order to minimize the effect in recognition process.

By solving the above problems, unambiguous question can be got by the system and it can follows an interactive approach, which is discussed below, to achieve the goal of interacting with dialogues with the user in the closed-domain QA System

Architecture of the system

After referring to the earliest QA System Algorithm presented by Simmons at 1973[2] and many of the publications in TREC QA System[3], our QA System has been divided into five modules as shown in figure 1:

i) Speech Controlling and Synthesizing Module
ii) Question Analyzing Module
iii) Recognition/Matching Module
iv) Knowledgebase Module
v) Answer Constructing/Selecting Module

At the beginning of our paper, we have mentioned that the QA System will be developed with the techniques of the speech recognition in order to let the user can input their question with their speech. Here, the first module has token the role of getting the question input by the user with the speech recognition engine. The engine we have chosen to use is IBM Speech Engine due to their 40-years profound research and development in that area. Besides, IBM has provided a complete API for developer so that we do not need to spend so much time on the low-level control of the speech engine and recognition process. After getting the input question, we need to pass that question to the Question Analyzing Module to analyze the question. If it is successfully analyzed, it will be passed to the Recognition/Matching Module to extract the answer; if not, or corresponding answer can not be extracted, response will be given by the system through the speech engine and therefore communicate with the user so as to achieve the goal of bidirectional communication between the user and the engine. Answer will be extracted from the Knowledgebase Module, just like database, and then pass the data to the Answer Construction/Selecting Module to construct the answer for output.

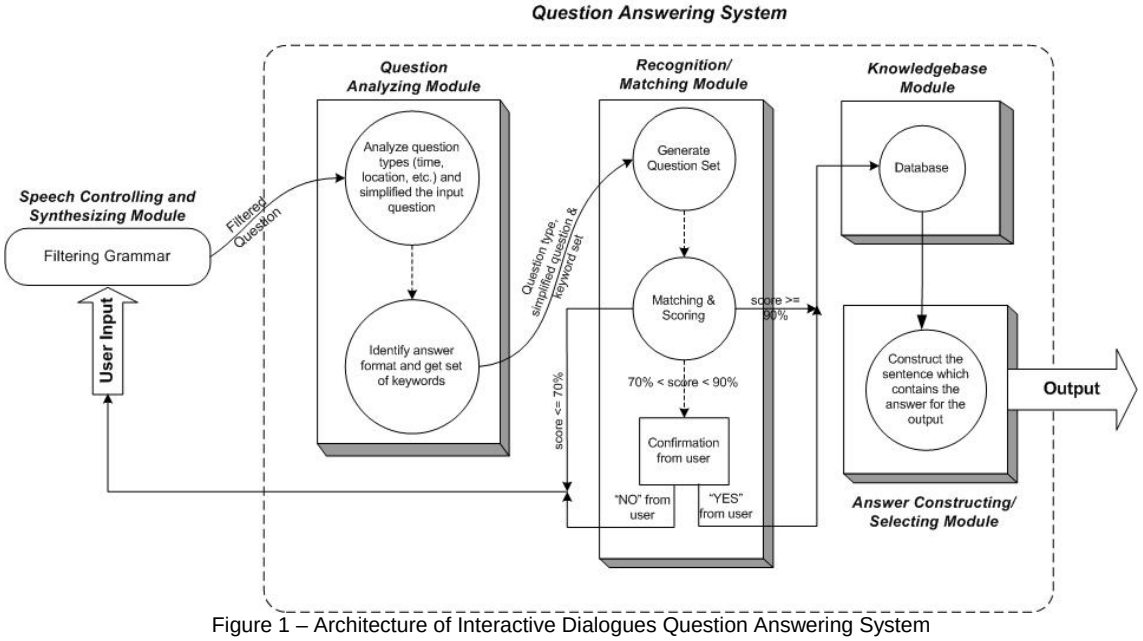

Figure 1 – Architecture of Interactive Dialogues Question Answering System

Questions filtering by IBM Speech Engine

Nowadays, the overall accuracy of speech recognition is not good enough to recognize all the speech from all different kinds of people. Some of the speech engines, for example, IBM, Microsoft, need a process called "Training" in order to increase the accuracy. If this QA system is single-user oriented, the training process can enhance the accuracy of the speech recognized. By using this process, user needs training the speech recognition engine by reading some passages so as to let it get familiar with different ways of speech of the user. However, as we can see, this training process will just improve the accuracy of that user. For IDQA system, which is a system for multiple users, this training process may be meaningless and unreasonable since we could not ask the users to train the engine in advance. For this reason, we decide to use the IBM Speech Engine without training.

In order to know the rate of accuracy of the IBM Speech Engine, we have done some testing and we found that the accuracy without training can be very bad. IBM Speech runtime can recognize more than 200,000 words including professional and special nouns. For short sentences, the accuracy is acceptable. For longer sentences, or sentences include proper nouns or homonyms words(feel, fill, field), the result is poor, in some cases, the recognized words can be even out of what we can expected, like, "Macao" is recognized as "Moscow" very often.

Due to the reasons we have discussed above, some restrictions about what can be said must be raised, i.e. users can only ask the questions within certain field (domain of knowledgebase). For example, if the domain is inside the campus, the proper nouns can be "library", "lab", "plaza", "toilet" and many other else, other irrelevant proper nouns or homonyms are omitted.

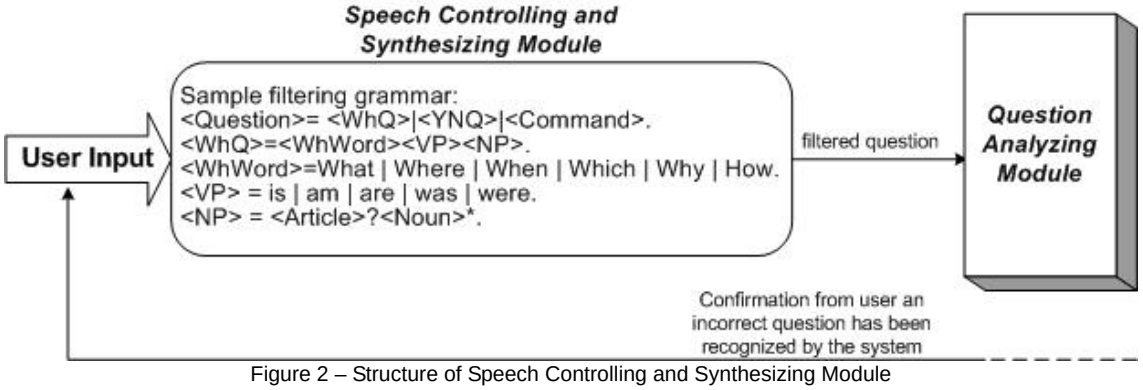

Figure 2 – Structure of Speech Controlling and Synthesizing Module

We can achieve the method we have discussed above by designing the grammar provided by the IBM speech engine, the Speech Recognition Control Language[4]by ourselves. SRCL Grammars formally define the set of allowable phrases that can be recognized by the speech engine, the language is in EBNF, such as:

<Question>= <WhQ>|<YNQ>|<Command>.
<WhQ>=<WhWord><VP><NP>.
<WhWord>=What | Where | When | Which | Why | How.
<VP> = is | am | are | was | were.
<NP> = <Article>?<Noun>*.

Here, "*" has the meaning 0 or more. "?" has the meaning "with or without". The <Noun> is the nouns of our knowledgebase. For example, in recognizing the question "Where is Macao Stadium", this question is in our specified grammar. This simple grammar is an example; the real grammar is much more complex. These phrases can be spoken continuously, without pausing between words. The grammar provides a language model for the speech engine, constraining the valid set of words to be considered, and increasing recognition accuracy while minimizing computational requirements. In our Q/A System, this is a possible way to filter out unpredictable sentences. Figure 2 shows how the speech engine works with the other modules in the system.

Analysis Routine in QA System

In order to make the real-world users find the QA System is useful, several standards must be met[5]. They are timelines, that is, the answer to a question must be provided in real-time; accuracy is extremely important because wrong answer is even worse than no answer; moreover, the answer is usable for the enquirer or not for there may be some cases that different answers are needed for that same questions for different domains; whether the complete answer is provided to the enquirer or not; and also the answer provided must be relevant within a specific context. Therefore, in order to fulfill these standards, a correct analysis of the question must be given before all. By referring to some passages and papers[6][7], we have concluded the following algorithm in analysis the problem.

i) coarse analyze the input question to identify its type (types include time, location, who, amount, method, size, number, what, command and determination) and simplify the question; if the system is unable to get the question type, communication with the user again so as to get some keyword
ii) detailed analysis according to the type of question to identify the answer format, which is predefined, provided to the user; moreover, a set of keywords of the input question will be got
iii) a set of questions will be generated according to the predefined question format according to the question type and the set of keywords got from the previous procedure. Then, matching the input question and the set of questions will be done by scoring
iv) and answer will be generated according to the data from the database and the format which is identified previously

Here, we first give the analysis procedure of dealing with normal question

After the speech engine recognized a sentence, it will be passed to the Question Analyzing Module to identify the type. The question-answering system categorized questions into several types according to wh-word: (1) **time** questions ("when"), (2) **location** questions ("where"), (3) **who** questions ("who" or "whose"), (4) **what** questions ("what" or "which," used alone or as question determiners), (5) **amount** questions ("how many", "how much" and so on), (6) **method** questions("how to", "how can"), (7) **size** questions ("how" followed by an adjective), (8) **number** questions ("what is number/amount"), (9) **command** ("list", "show" and so on), and (10) **determination** questions (start with verb-to-be, like "is", "are"). If the type can not be identified, our system will communicate with the user again like "Could you please repeat your question". Then the procedure will start all over again. In addition, with the question type, we can identify what the answer format is, like "The competition will start at <time>" for time questions or "The sport complex is located at <location>" for location questions and so on. Besides identifying the type, we will also simplify the questions like "Could you please tell me where is…". Since the phrase "Could you please tell me" is not important for our system, we will simply such kind of question by cutting this type of phrase away. Furthermore, set of keywords will be got from the input question. After this, the question type and the simplified question will be passed to Recognition/Matching Module.

There will be a predefined question format corresponding to the question type. With the set of keywords, we can generate set of questions according to the question format in the Recognition/Matching Module.

Now, we can match the simplified input question with the set of questions we have just generated. Score will be given for each matching and set of scores will be given corresponding to the each question in the question set. The score set, keyword set and the answer format will be pass to the next two modules. If the highest score can not reach certain level, for example, 50%, this implies that the question recognized by speech engine maybe can not be answered due to the lack of knowledge, or it may be not belonging to our field or the question recognized is understandable enough. In such case, our system will communicate with the use with keywords got in order to get another more understandable question and the procedure will start over again. In other case, if the highest score can reach the level of 70% to 80%, we can confirm the question with the questions of highest several scores. If we can get the answer of "yes", then move to the next module; if not, request the user to input the question again.

Here, if we can reach the Answer Constructing/Selecting Module, that means the generated question that has been passed to this module must be confirmed by the user or the score taken must be greater than or equal to the 90%. Now, we can extract the answer from the database according to the question type and the keywords got. After extracting the answer, the system will generate a sentence according to the answer format, for example, like "The sport complex is at <location>". "<location>" means the answer extracted from the database. Finally, this sentence will be "spoken" by the speech engine. This is the overall procedure for the analysis routine.

Coarse and ambiguous utterance Analysis
The analysis of ambiguous utterance for computer can be very complex. Since the accuracy of speech input is not that high, wrong input may be generated very often. Empirically, human often think and pause during asking questions, then some irrelevant words what we have just mentioned will appear. So, the system should filtrate out those words and make sure it got the complete question but not a partial one.

What difficulty about answering questions is that the fact that before a question can be answered, it must be first understood. One level of the interpretation process is the classification of questions. This classification should be determined by well defined principles. As mentioned, wrong answers are worse than no answer.

If the system does not understand the question entirely, the best answer is "Sorry, please ask again". How about the system understands the question partially but not very well? We have used the method just like human being, that is, our system will try to repeat the question which has already been modified to the user in order to get the confirmation. Then, just as the procedure above, we can either proceed to the next module or start the procedure all over again.

Difficulties

After getting an unambiguous question, there are still several bottleneck difficulties in the research and implementation of Speech QA system.

Taxonomy – Questions are difficult to be classified; then, if the question cannot be classified, low efficiency and low usability may occur. For a question with no type, it may need to scan through all candidates in the knowledgebase and match against with all of them in order to evaluate the most possible answer. This problem occurs when the question contains "How" or "What".

Constructing complete answers – Even completeness in QA system is very important, the difficulties in answer complete questions are a bottleneck. Knowledgebase are not presented as passages or documents but in different database tables. Most of the time, answers are **dynamically** constructed from different tables.

Matching and Scoring the questions – The efficiency and correctness of catching keyword are both low. Setting up an efficient algorithm for algorithm for matching is a very tough job. Till now, our algorithm for catching keywords is not good enough. This is also the most difficult part we need to solve in the research and implementation of IDQA.

We need to solve the bottlenecks discussed above one by one in the future. Particularly, the above bottlenecks are human natures. Human can easily solve them but computer cannot. So, solving the above problems is indeed improving humanity in computer science, an area of Artificial Intelligence.

Prototype of Interactive Dialogues Question Answering System

The main interface of the prototype of IDQA is shown in figure 3. Here we can use our microphone as the input media give query input after we press the button. The question we asked is shown in the textbox just above the button in this prototype for reference. Here, PakPak is welcoming us as a guide!

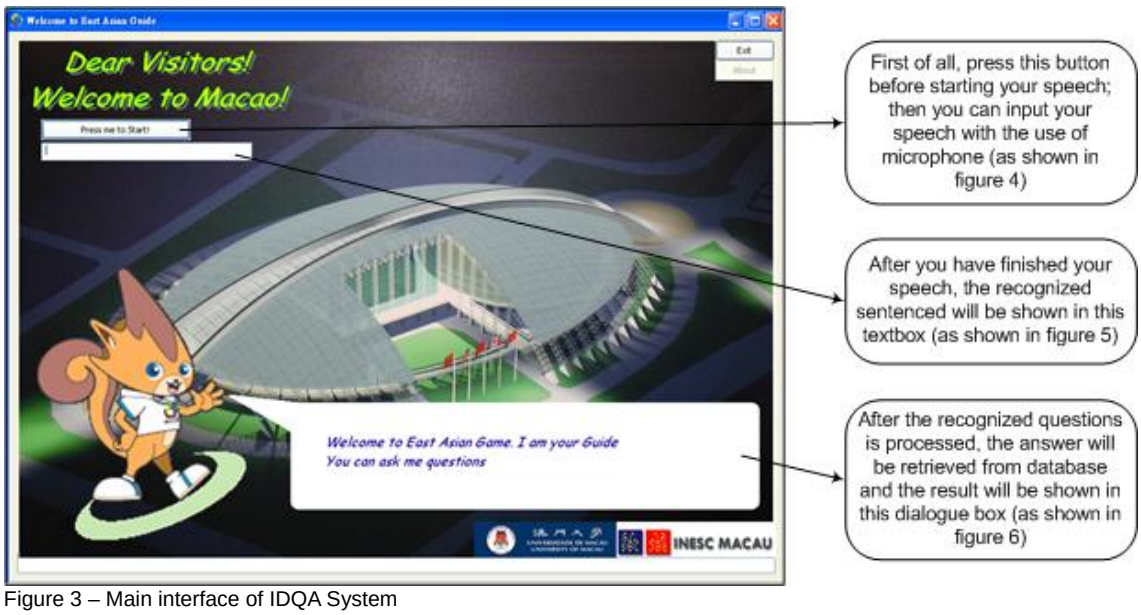

Figure 3 – Main interface of IDQA System

In figure 4, the user is speaking to the system and asking for a question "Where is Macao Stadium". After the question is recognized, it is temporarily displayed in the textbox for simple reference.


Figure 4 – The user is speaking to the system

Figure 5 – The question is recognized and output to the textbox

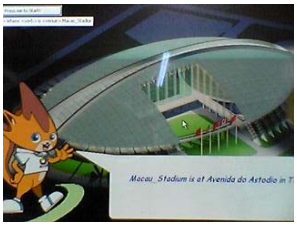
Figure 6 – The answer is retrieved from the database and output to the textbox

After the recognized question has been processed by analysis routine, the answer is retrieved and displayed in the dialogue box as the output. Here, the text "Macao Stadium is in Avenida do Astodio" is displayed as the output.

Conclusion and Future work

Speech QA System is an interpretation of Human-Computer Interaction. Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them. Human use their speech instead of typing into the keyword and the computer, making computing more natural, like a friend but not a machine. This paper has described a preliminary prototype Speech Question Answering System. At this moment, our system is still not fully-functioned; however, we have already developed some of the basic functions although the functions are not very well-developed. In the coming work, we need to enhance the technique in analyzing the input question; therefore, we need in improve the accuracy in getting the keywords. If the keywords could not be caught correctly, our system would be meaningless to some at extend. Moreover, since the grammar which we are using now is just only good enough to accept the most common ways of question asking, we need to improve the ability of accepting them by enhancing the design of the grammar. Furthermore, an algorithm of matching and scoring the questions must be designed for the seek of getting a correct and unambiguous questions. The algorithm we are using now is only matching keywords in the question and the score is given according to the number of keywords matched. Later on, we are going to improve this by applying Dynamic Programming (DP) with different

keyword should has different score. Finally, we are going output the answer with the speech instead of text which we are using so as to achieve the goal of having dialogue with speech.

References

[1] Question Answering, "*Question Answering (QA) is a type of information retrieval*", http://en.wikipedia.org/wiki/Question_answering.

[2] Simmons, R.F., "*Semantic Networks: computation and use for understanding English sentences*", San Francisco, 1973.

[3] Enrique Alfonseca, Marco De Boni, Jose-Luis Jara-Valencia, Suresh Manandhar, *"A prototype Question Answering system using syntactic and semantic information for answer retrieval"*, in Proceedings of the Tenth Text Retrieval Conference (TREC-10), Gaithersburg, US, 2002.

[4] IBM(R) ViaVoice™ SDK - SMAPI Developer's Guide, pages 17-18.

[5] John Burger, Claire Cardie, Vinay Chaudhri, Robert Gaizauskas, Sanda Harabagiu, David Israel, Christian Jacquemin, Chin-Yew Lin, Steve Maiorano, George Miller, Dan Moldovan, Bill Ogden, John Prager, Ellen Riloff, Amit Singhal, Rohini Shrihari, Tomek Strzalkowski, Ellen Voorhees, Ralph Weishedel, *"Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A)"*, 2001.

[6] Kenneth C. Litkowski, "*Syntactic Clues and Lexical Resources in Question-Answering"*, Ninth Text REtrieval Conference(TREC-9). Gaithersburg, MD. November 13-16, 2000.

[7] Ellen M. Voorhees, *"Overview of the TREC 2003 Question Answering track"*, In TREC 2003 Proceedings, 2004.