ISSN:2320-0790

An International Journal of Advanced Computer Technology

A COMPARATIVE STUDY OF VARIOUS APPROACHES FOR DIALOGUE MANAGEMENT

Manzoor Ahmed[1], Romana Riyaz[2] and Saduf Afzal[3]

Department of Computer Sciences,
University of Kashmir Srinagar, Kashmir 190006, India

manzoor@kashmiruniversity.net[1], samar_riyaz@yahoo.com[2], sadaf.afazl123@gmail.com[3]

Abstract: This paper has been developed to draw a comparison between various available approaches for dialogue management. Currently there is much interest in building interactive human-computer interfaces which involve spoken input and output. Spoken dialogue system usually combines speech recognition with natural language understanding, language generation and dialogue management. Dialog systems are created for domain specific applications, so that a high demand for a flexible dialog system framework arises. There have been several approaches to dialog management. In this paper we present three different approaches to the dialog management.

Keywords: *Markov decision process, Templates, Dialogue management, partially ordered markov decision.*

1. INTRODUCTION

Dialogue may be defined as an interaction / a spoken or written conversation exchange between two agents based on a sequential turn taking with an aim of achieving some goal. When one of the agents is a computer and the other is human, the dialogue is known as Human- Computer Dialogue. Human-to-computer interaction is a form of natural language Processing task between human and the computer where the elements of human language, be it spoken or written, are formalized so that a computer can perform value-adding tasks based on that interaction.

Spoken dialogue systems allow users to interact with computer-based applications such as databases and expert systems by using natural languages. The origins of spoken dialogue systems can be traced back to Artificial Intelligence research in the 1950s concerned with developing conversational interfaces. However, it is only within the last decade or so, with major advances in speech technology, that large-scale working systems have been developed and, in some cases, introduced into commercial environments. As a result many major telecommunications and software companies have become aware of the potential for spoken dialogue technology to provide solutions in newly developing areas such as computer-telephony integration. Voice portals, which provide a speech-based interface between a telephone user and Web-based services, are the most recent application of spoken dialogue technology

In recent time there has been large increase in the number of spoken dialogue systems that have been developed. Advances in key technologies behind spoken dialog systems (automated speech recognition, natural language understanding, dialogue management, language generation and synthesis) have allowed researchers to build systems in a variety Of domains. The system implementation mainly depends on the use of grammars and dialogue management. Information access domains have received a lot attention especially due to the relatively simple structure of the dialogues in these

domains. An increasing number of companies are automating tasks that were previously performed by human operators in call center applications (e.g. checking credit card balances, making travel reservations, checking baggage status etc.). Spoken dialogue interaction has been suggested by researchers and practitioners as a promising alternative way of communication between humans and machines. A compelling motivation is the fact that conversational speech is the most natural, efficient, and flexible means of communication among human beings. Since there is lot of complexity in human-human interactions so, talking to a machine requires a spoken dialogue system.

In order to develop spoken dialogue system one needs to have large corpus of data for system refinement and evaluation, building such systems is still a challenge for science and engineering. Thus a spoken dialogue system may be defined as an intelligent agent that interacts with humans using spoken language in order to perform some task a spoken dialog system enables a human user to access information and services that are available on the computer using a spoken language as the medium of interaction. In future visions of the interaction technology, a talking computer is portrayed as all knowing, often human like, with the ability to provide all types of useful information, recognize gestures and emotions, react to the problems and other wide range of situations. Commercially available systems are able to automate a variety of customer services thus making human users free from the mundane tasks that are repetitive and thus can be easily automated, and for which a spoken dialogue is also a natural mode of communication.

2. ARCHITECTURE OF SPOKEN DIALOGUE SYSTEM

A common spoken dialogue system consists of following components:

- Automatic Speech Recognition component (ASR)
- Natural Language Understanding component (NLU)
- Natural Language Generation component (NLG)
- Text-To-Speech synthesis component (TTS)
- Dialogue manager

I. Automatic Speech Recognition

The user's speech input consists of a string of acoustical signals that are converted into a sequence of words by the speech recognition module. To achieve this goal, most **speech recognition** modules use statistical methods – such as Hidden Markov Models (HMMs). First, they generate a word lattice of the *n-best* word solution sequences, with simple models to compute approximate likelihoods in real-time. Then, more accurate likelihoods are computed and compared with a limited number of hypotheses to determine the most likely word sequence the language model for speech recognition is a network (regular) grammar, and it allows each speech interval to be an arbitrary number of *phrases*. A phrase is a sequence of words, which is to be defined in a domain-dependent way. Sentences can be decomposed into a couple of phrases. The reason we use a repetition of phrases instead of a sentence grammar for the language model is that the speech recognition module of a robust spoken dialogue system sometimes has to recognize spontaneously spoken utterances, which include self-repairs and repetition. When the speech recognition module finds a phrase boundary, it sends the category of the phrase to the language understanding module, and this information is used in the parsing process. It is possible to hold multiple language models and use any one of them when recognizing a speech interval. The language models are switched according to the requests from the language understanding module. In this way, the speech recognition success rate is increased by using the context of the dialogue.

II. Natural Language Understanding

The Natural language understanding module takes the word sequence delivered by the speech recognizer and analyzes it syntactically, pragmatically and semantically. The aim is to determine the intended meaning of the word sequence. This process is called **parsing**. Usually the parsing process uses a grammar which describes how words in an utterance are combined.

III. Dialogue Manager

The decoded information is then sent to the **dialogue manager**. This unit plays a central role in operating the dialog – a difficult task. Very frequently the recognized words delivered by prior modules are fragmented and incorrectly modeled. Background

noise, human noises such as sneezes, coughs, unequal emphasis, word accents, inarticulate and incomplete pronunciations, word or syllable recurrences and different acoustical realizations of the phonemes can significantly affect speech recognition performance.
A spoken dialog can be judged to be successful if it provides correct information and comprehends it correctly. The dialogue flow is controlled by the dialogue management module. This module has to determine whether sufficient information has been elicited from the user in order to enable communication with an external application, to engage in communication with the external application to retrieve the required information, and to communicate that information back to the user. The dialogue management module is also responsible for detecting and repairing breakdowns in the dialogue through verifications, confirmations and corrections. Dialog management systems can be categorized in terms of type of control offered and how the control is managed. Dialogues control may be system-led, user-led or mixed initiative. In a system led dialogue the system asks a sequence of questions to elicit the required parameters of the tasks from the user. A user led dialogue is controlled by the user who asks system questions in order to obtain the information. In a mixed initiative dialogue is shared. The user can ask questions at any time, but the system can also take control to elicit the required information. Dialogue manager is often viewed in terms of two sub components: dialog control, which deals with the flow of control in the dialogue and dialogue context modeling, which is concerned with contextual information used by dialogue manager to interpret user's input and inform the decisions of dialogue control component. Different approaches to the dialogue management problem are (1) Graph-based dialog management (2) Frame- based dialog management 3) Statistical approach.

A. Graph-based dialog management

One of the simplest approaches is to represent the dialogue manager as a graph or flow-chart, sometimes called the call-flow [9].This method is also known as "finite state based "dialogue control since the states of the dialogue graph can be traversed using finite state automaton. The nodes of graph represent system questions, the transitions between the nodes represent answers to questions, and graph specifies all legal dialogue. Alternatively each stage in dialogue can be viewed as a state in which some dialogue action is performed. Sub dialogues can be also be used within the basic network to support a more modular design approach. Most commercially available spoken dialogue systems use this form of dialogue control strategy. A finite state model can also be used for structured tasks such as obtaining weather forecasts, football scores, ordering items from a catalogue, or making simple bank transactions. For example

System*: What is your destination?*
User*: SRINAGAR*
System*: Was that SRINAGAR?*
User*: Yes*
System*: What day do you want to travel?*
User*: Friday*
System*: Was that Sunday*
User*: No*
System*: What day do you want to travel?*

This approach generally uses finite-state automata which often involve handcrafted rules. Graphs have been widely used to design dialogue systems and there are several toolkits, such as the centre for spoken language understanding (CSLU) toolkit, and various other commercial toolkits that allow designers to construct a dialogue by dragging, dropping, and connecting icons representing dialogue objects on screen. The toolkit then compiles the graph into a language such as Voice *Xml*, which can be interpreted and run as dialogue application. The management of dialogue control is straight forward in systems based on graphs, since the transitions to the next node are predetermined and are based on a small set of conditions, such as whether the user's utterance was understood with a sufficient degree of confidence. Use of graphs is suitable for fairly simple interactions where dialogue flow can be predetermined. In this way the semantics of the system is clear and intuitive. A major advantage of the finite state model is its simplicity. Moreover, as the user's responses are restricted, fewer technological demands are put on the system components, particularly the speech recognizer. The lack of flexibility and naturalness can be considered as a trade-off against these technological demands. For these reasons most currently available commercial systems use some finite-state dialogue modeling.once the dialogue becomes more complex, the number of nodes and transitions show a significant increase and it is no longer possible to build the graph manually. This approach has proven to be very effective when the system's prompts elicit highly restricted responses from the user. On the

other hand, the call-flow model typically struggles when users take the initiative and direct the dialogue themselves. Graph based dialogue is appropriate for well structured tasks, in which there is predetermined set of questions to be asked. Some of the examples of finite state based systems are The automatic book service, directory assistance, questionnaires, and travel inquiries, provided the dialogue is constrained to a basic, system-led series of questions to elicit a number of well-defined responses.moreover, since the space of user responses to each prompt can be predicted, the speech recognition and language understanding can be restricted to what the user is expected to say following each prompt, thus enhancing the performance of the ASR and NLU components. Response generation is also simpler since in conformations the system can reflect back the information provided by the user in the previous utterance and gather together the information retrieved from the database into a predefined template. Dialogues developed using the graph based method support a style of interaction known as system- directed(or system initiative) dialogue. In this type of interaction system prompts the user for one or more items of information. The system might confirm that it has understood each item as it is elicited, or may leave conformation until some or all items have been collected. Once all the information has been gathered, the system performs some task, such as retrieving the required data, and then outputs this information to the user. The user has no control on the dialogue flow and is mainly constrained to responding to the system's prompt, although there are often options for go back or start over that allow the user to take some degree of control over the interaction. Finite state dialogue models are generally rigid and inflexible. These characteristic would not have been a problem if the interaction with the user is controlled by the system and restricted to a well ordered sequence of questions. However, because the dialogue paths are specified in advance, there is no way of managing deviations from these paths. Problems arise if the user needs to correct an item or introduce new information or topics that was not foreseen

At the time of the design of the dialogue Adding natural language facilities, while providing the user with greater flexibility in what they can say, can add to these problems.thus system directed dialogue would not be suitable for more complex tasks such as planning a holiday that involve negotiation of the constraints that had not been predicted in advance by the system designer and included in the dialogue graph.for example in a simple travel inquiry system, a system might ask the questions in the following order: *destination* > *origin* > *date* > *time*. However, when answering the system's question concerning destination the user might reply with a destination as well as the departure time (or indeed other combinations of the four required parameters). A finite-state based system would simply progress through its set of predetermined questions, ignoring to process the additional information that was provided by the user with the destination and then asking an irrelevant question concerning the departure time. The solution to this problem would be to include a dialogue model so that the system 'knows' what it has already elicited as well as what has still to be asked. The system could then continuously loop through the dialogue model until all the required information has been elicited. In this way the problem of irrelevant questions would also be avoided to some extent. However, the problem is that as soon as the number of items grows, the number of transitions to cater for each required dialogue path grows to unmanageable proportions.

B. Frame based dialog management

In a frame (or template) based system the user is asked questions that enable the system to fill slots in a template in order to perform a task. In this type of system the dialogue flow is not pre-determined but depends on the content of the user's input and the information that the system has to elicit in the course of dialog. Frame offers a better way of providing flexibility to the dialogue control. Frame based systems function similar to that of production systems, taking a particular action based on the current state of dialogue. The questions and other prompts that the system should ask can be listed along with the conditions that have to be true for a particular question or prompt to be relevant. Some form of natural language input is required by frame-based systems to permit the user to respond more flexibly to the system's prompts. Natural language is also required to correct errors of recognition or understanding by the system. Generally, however, it is sufficient for the system to be able to recognize the main concepts in the user's utterance.

The frame-based approach has several advantages over the finite-state based approach, for the user as well as the developer. As far as the user is concerned, there is greater flexibility in terms that it allows the user to fill in the slots in different orders and different combinations. The ability to use natural language and

the use of multiple slots filling enables the system to process the user's over-informative answers and corrections, as in
S1: where are you flying to?
U1: Srinagar, departing at 9.
This type of input is referred to as "over answering" where the user provides more information than is requested in the system prompt. Over answering is not possible in a graph based dialogue system because each system prompt is linked to a range of responses that in turn lead to the next system prompt. Thus in a graph based system, a response such as S1 would expect a word or phrase stating a destination as a response. If the user produced a response such as U1 this additional information would either be ignored or mis-recognized.Infact the repercussions could be even worse, as the system would subsequently ask the next question in the graph that is "at what time do you want to depart?"This would be inappropriate as the user has already provided this information. Interpretation of the user's input in the frame based system is much more complex when compared with the graph based systems, as the user's response can include various permutations of the required information e.g. in response to the question "where are you flying to?" the user may respond with an utterance that may contain the following combinations such as:
Destination
Destination+date
Destination+time
Destination+time+date

In such a case more complex grammars would be required in which all the items that are relevant to a dialogue are extracted from the user's utterance to fill the slots in a frame. Semantic grammars' are often used for this purpose because they focus on the semantic content of the input rather than its syntactic structure. They are also more robust to the types of ungrammatical input and recognition errors that occur in spontaneous speech. In addition to permitting a wider range of user inputs, the use of frames provides for a more flexible dialogue flow since the dialogue can be driven by the information needed at a given state in the dialogue and actions to fill slots can be executed in various sequences this flexibility is desirable for applications that permit more flexible slot filling and in which it is desirable to allow the user's more freedom in how they input information to the system. However frame based systems require a more elaborate dialogue control algorithm to determine what would be the system's next question based on the analysis of the user's previous utterance in conjunction with a template of slots to be filled and a number of priorities for control of the dialogue . a frame-based approach has the disadvantage like that of any production system with a large number of rules and contexts in that it is difficult to predict which rule (or question) is likely to fire in a particular context. A considerable amount of experimentation may be required to ensure that the system does not prompt an inappropriate question under some circumstances that had not been foreseen at design time.

C. Statistical Approach

Spoken dialogue systems have to deal with uncertainty which is inherent during the process of speech recognition and natural language interpretation. A common approach is to augment the hand crafted belief states which represent uncertainty as well [10].But most of the models lacks the principled definition for these states of uncertainty. So an alternative model Sequential decision process model which are natural and augmented a well researched framework based on Markov decision process[11], to aid the spoken dialogue system reliably identify the underlying environment state. Sequential decision process framework interacts synchronously with the external environment i.e. the user with the main goal of maximizing its reward by taking appropriate actions. These actions and history of the environment state determine the probability distribution over next possible states and such are modeled as a stochastic process.

1) ***Markov decision process*:** A complete dialogue system can be modeled as a Markov decision process (MDP) in which each dialogue exchange results in a state transition from S to S`. It is a formal model of fully-observable sequential decision processes which is an extension of Markov chains with a set of decisions/actions and a state based reward structure. In this process for each state a decision has to be made regarding the action to be taken in that state to increase some predefined measure of performance. The action affects not only the transition probabilities but the rewards as well. A state describes the environment at a particular instant of time. It is assumed that the system can be in a finite number of states and the agent (SDS) can choose from

a finite set of actions. Let S =S0, S1, S2…SN be a finite set of a states. Each state at discrete time t ∈ T is viewed as a random variable S t whose domain is the state

Space S as the process is stochastic. The past history in the form of system states is irrelevant in predicting the future so the state must contain enough information to predict the next state for the process to be Markovian

$$P_r(S_t + 1 \mid S\,0\,, S\,1\,, \ldots S_t\,) = P_r(S_t + 1 \mid S_t\,)$$

The Spoken Dialogue System at each state executes one of the available actions (a) from a set of actions (A) which affects the state transition probabilities. Thus each action a ∈ A is fully transcribed by a | S | X | S | state transition matrix whose entry in ith row and jth column is the probability that the system will move from states i to the states j if the action a gets executed.

$$P\ a_{ij} = P_r(S_t{+}1 = s_j \mid S_t = s_i\ ,\ A_t{=}\ \mathrm{a})$$

The effect of the actions A on the system states S is given by transition function (T) where

T: SXA → Δ(S) which associates a probability distribution over the possible successor states and δ(S) represents the set of probability distribution over S. Thus for each state s, s0 and a ∈ A the function T determines the probability of a transition from state s to state s0 after executing action a.

$$\mathrm{T}\ (\mathrm{s},\ \mathrm{a}, s_0) = \mathrm{P\ r}\ (S_{t+1} = s_0 \mid S_t = s_i\ ,\ A_T = \mathrm{a})$$

The spoken dialogue system assigns a reward (or cost if the value is negative) for being in a state s and executing action a using a reward function

R: SXA → R0

Even with the benefits of simulations for efficiently generating training data, the state space must still be cut down significantly if optimization is to be tracable.The effect of this is inevitably to discard history information and thereby make the process non-markovian.It has been found that using eligibility traces can compensate somewhat for this effect[j loch and S Singh, 1988].Directly observed discrete MDPs will always be a rather limited compromise solution to the problem of dialogue modeling. This is especially true in real system s where recognition error rates can be high and the need to explicitly model the system's uncertainty in the user's beliefs, desires and intentions cannot be ignored.

2) ***Partially observable Markov decision Process (POMDP) Framework*:** The principle problem with the approach described earlier is that the complete dialogue state is never in practice observable. More accurate modeling demands that a minimum the hidden nature of the user state is acknowledge. Also due to recognition error the system state is also uncertain. To handle this the dialogue is modeled by a partially observable Markov decision process(POMDP).A POMDP is a generalization of MDP'S in which system states are not fully observable.POMPD framework is based on the underlying MDP extended with observation space o and observation function z(.).

In a POMPD, system actions are taken on the basis of the belief state rather than the dialogue state. In POMPD framework, the user dialogue acts are regarded as observation arising from a noise measurement process ,where the true underlying signal is the user output .In MDP's the dialogue system has a complete knowledge of the system states whereas in the case of partially observable environments, observations are only probabilistically dependent on the underlying environment state. Also the same observation can be observed in different states which make it difficult to determine the state of the system observation function Z: S x A -> δ (O) specifies the relation between the system states, actions and observation space.

In terms of practical application of POMPD's dialogue, relatively little has been done. In the earlier attempts towards the POMPD framework used a simplified vector representation of belief state

consisting of the most likely state and the entropy of the belief state [14] later a variable grid approximation to calculate value function to calculate value functions and the result suggested that this is significantly better than entropy approximation [13] this work was based on simulation using Bayesian networks.

In a POMDP the complete system history from start till time t is represented by a triplet i.e. by the system state, the observation and the action taken e.g. (S_0 ,O_0 , A_0), (S_1 , O_1 ,$1A_1$), ..., (S_t , O_t , A_t)
The history is the record of everything that has happened during the execution of the process. In partially observable environment, the system bases its decision on the observable history as it cannot fully observe the underlying world state. The SDS has the prior belief about the world state which is summarized by the probability distribution b_0 over the system states and the system starts by executing an action a 0 based on the distribution b_0 . The set of all observable histories or trajectories are represented as H_0 .Representing and structuring hH_0 in different ways has led to different POMDP solutions and Policy execution algorithms.

IV. Natural Language Generation

The natural language generation module constructs natural language outputs from non-linguistic inputs. The goal of this module can be viewed as the inverse of that of natural language understanding module (NLU) in that NLG maps from meaning to text, while NLU maps from text to meaning.

V. Text To Speech Component

The text from the NLG will be forwarded to the text to speech module (TTS). The text can be either read out by a speech synthesis unit or, pre-recorded; natural language utterances can be played. Thus, the information processed by the system gets back to the user in an acoustical form. It also notifies the language generation module that
Speech output has finished so that the language generation module can take into account the timing
Of the end of system utterance.

TTS plays a very important role in a SDS. The voice that the TTS produces directly interferes with the only sense that the SDS makes active – the hearing. On the quality of the synthesized speech stands or falls the complete impression from the SDS. The synthesized speech is not just an output from the TTS module, but it stands for all the processes that are realized in all of the modules since the user's input. Synthesis of poor quality considerably diminishes the probability of the SDS to be used again. The quality of speech synthesis is measured by two factors: its intelligibility and naturalness. For the proper and satisfying functioning of the SDS a TTS producing an intelligible speech is doing enough. However, the trends in speech synthesis of the last two decades head towards suppressing the robotic-style speech. The main challenge in TTS within a SDS is then the trade-off between the intelligibility and the naturalness.

3. CONCLUSION

The research in the field of spoken dialogue systems has been increasing tremendously due to growing demand and interest for improved human-machine interaction. Large number of spoken dialogue systems has been successfully developed and diverse type of approaches for dialogue management has been used. Frame based approach provides a greater flexibility to the user and efficiently processes even over-informative inputs of users while Graph based approach states of the dialogue graph can be traversed using finite state automaton .But graph approach becomes complex when the complexity of the input increases. Finally the Statistical Approach deals with the inherent uncertainty in the speech recognition and also takes into account the past states and dialogue history.

4. REFERENCE

[1] Daniel Jurafsky, James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition,
[2] Ralf Engel Natural Language Understanding
[3] S. Russell, P. Norvig Artificial Intelligence, A Modern Approach
[4] Andreea Niculescu Laerholzstr. Implementation and Evaluation of a Spoken Dialog System for Exam Result Enquiries
[5] Gregor Rozinaj, Andrej Páleník, Jozef Čepko, Jozef Juhár, Anton Čižmár, Milan Rusko,

Roman Jarina ”*The First Spoken Slovak Dialogue System Using Corpus Based TTS”*

[6] Mihai Rotaru. Applications of Discourse Structure for Spoken Dialogue Systems

[7] Mikio Nakano , Noboru Miyazaki, Norihito Yasuda, Akira Sugiyama,Jun-ichi Hirasawa, Kohji Dohsaka, Kiyoaki Aikawa.WIT: A Toolkit for Building Robust and Real-Time Spoken Dialogue Systems NTT Corporation

[8] Ali Rahmani.Spoken Dialog Systems Seminar: Advances in Human Computer Interaction University of Saarland (2008)

[9] Micheal F McTear,modeling spoken dialogues with state transition diagrams

[10] Bohus, D. and Rudnicky, A. Constructing accurate beliefs in spoken dialog systems. In Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on, pages 272–277. IEEE, 2005.

[11] Puterman, M. Markov decision processes: Discrete stochastic dynamic programming. John Wiley& Sons, Inc., 1994.

[12] J Loch and S Singh *“Using eligibility traces to ind the best memory less policy in partially observable markov decision process*”. In 15th *in conf on machine learnin*g (ICML-98), 1998.

[13] B Zhang, Q Cai, J Mao, E Chang, and B Guo”*spoken dialogue management as planning and acting under uncertainty”*. In*Eeurospeech,*Aalborg,Denmark,2001.

[14] N Roy, J Pineau, and S Thrun, “Spoken dialogue management using probabilistic reasoning,” in Proceedings of the ACL 2000, 2000.