2

SPOKEN DIALOGUE SYSTEM

CONTENTS

OBJECTIVE OF THE CHAPTER

The objective of this chapter is to study the basics about spoken dialogue system. In this chapter we explain the characteristics, components, challenges, classification, evaluation and some examples of spoken dialogue system. As in our work, we focused on recognition and system testing module so evaluation and error handling mechanism of spoken dialogue system is also included in this chapter.

2.1 INTRODUCTION

Spoken dialogue system (SDS) is a computer program designed to communicate between human and computer through speech. It is an automated tool or a computer agent that interacts with humans through expertise spoken utterances. There are exclusive methods of interaction among human and system like text, gestures, symbols, speech or mixture of these which is done through dialogue turns [13]. Among all of this speech is the most natural manner of verbal communication. The main goal of a spoken dialogue system is to provide information by conversing with a human-being in a natural way. To take input, spoken dialogue systems used small set of spoken words like yes and no, digits such as 0-9 or combination of both, DTMF etc. After processing the input, the output may be spoken or displayed as text on a screen, and may be accompanied by a visual output in the form of tables or images. Some of the main characteristics of spoken dialogue system are [13]:

a. The system should identify the user's purpose, initially what the user is trying to look for.

b. There should be provision for both the parties to control the dialogue flow during their conversation, while they discuss and explore the solution in mixed initiative interaction mode.

c. Depending on the conversation history and the current context, interactions are contextually interpreted.

d. The system should understand the user's goal and appropriate path to reach the satisfied solution.

e. There should also be an ability to carry out sub-dialogues, in order to achieve sub-goals.

f. There should be a facility to pass control from one sub-dialogue to another.

g. There should be an ability to vary the dialogue initiative modes, from system initiative to user initiative.

h. Use of a user model to expect user's utterances and act aptly.

i. Ability to give direction to user towards task completion.

2.2 SPOKEN DIALOGUE SYSTEM ARCHITECTURE

Spoken dialogue systems are very complex to setup as it requires a good number of technologies to process the human language, which is a very complex task. Spoken dialogue systems (SDS) often take the form of complex software architectures that consist of a wide range of interconnected components. These components are dedicated to various tasks related to speech processing, understanding, reasoning and decision-making. The architecture of spoken dialogue system is broadly categorized into sequential architecture and centralized architecture. In sequential architecture, each individual module communicates directly with the other module forming a pipeline. In centralized architecture, a central module or central communication manger is present, which connects all the modules together. All modules interact with each other through this communication manager. A common spoken dialogue system consists of six components as follows [13-16]:

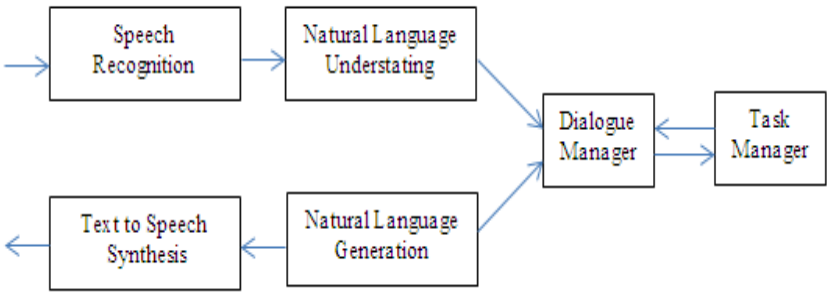

Figure 2.1: Architecture of Spoken Dialogue System

2.2.1 AUTOMATIC SPEECH RECOGNITION: Automatic speech recognition system maps the spoken utterances (a speech signal) into sequences of words. For this purpose, the incoming speech signal is first segmented into regions containing speech and silence. A simple approach to voice activity detection is to identify regions with energy level lower than some set threshold as silence and the rest as speech. This is not a trivial task as silence is not an accurate measure of whether a speaker has finished speaking. Also, background noise in the environment of system use can

further make it hard to tell apart speech from silence. More sophisticated approaches can also take spectral features into account to discriminate between speech and noise. Once a speech segment is obtained a corresponding sequence of words is searched [13].

2.2.2 NATURAL LANGUAGE UNDERSTANDING: The task of the natural language understanding (NLU) component is to parse the speech recognition result and generate a semantic representation. Such components may be classified according to the parsing technique used and the semantic representations that are generated. It produces a semantic representation of words from the strings, using syntactic and semantic analysis. There are several methods which can be used in this analysis, namely semantic grammar, probabilistic semantic grammar, semantic HMM, template based semantic etc. As the name suggests this unit tries to understand what user want to tell. It converts the sequence of words into a semantic representation that can be used by the dialogue manager. This component involves use of morphology, syntax and semantics. Morphology is the study of the structure and content of word forms. After identifying the keywords and forming a meaning it provides the result to the dialogue manager [14].

2.2.3 NATURAL LANGUAGE GENERATION: It generates in each dialogue state, an appropriate expression. It mainly deals with what to say addressed by the content planner module and how to say, addressed by the language generation module [13].

2.2.4 TEXT-TO-SPEECH SYNTHESIS: This module converts the generated words into audio. A speech synthesizer or text-to-speech (TTS) system turns text into speech. Usually it will speak just by supplying it with plain text. This can be done by developing speech models, modeling how speech sounds based on transcribed recordings (Acapela, 2012a) [15]. As further described by software Synthesizer Company Acapela (2012b), the most important qualities of a speech synthesizer are naturalness and intelligibility. Naturalness measures how natural the speech synthesizer sounds, how close to a real human being it is. Intelligibility on the other hand measures how well the user understands what the speech synthesizer has said. A good speech synthesizer must be both natural and intelligible [16].

Figure 2.2: A detailed architecture of Spoken Dialogue System

2.2.5 DIALOGUE MANAGER: It takes the output from the NLU module and passes it to the task manager and vice-versa. It also controls the structure of the dialogue. A dialogue manager interprets the user's input in the context of the dialogue, keeping track of what has been said and deciding what to say next. The dialogue manager simply controls the behavior of the dialogue system. There are several approaches to dialogue management. Some are well-tested, some are newer. The Dialogue Manager manages all aspects of the dialogue [17]. It takes a semantic representation of the user's text, figure out how text fits in the overall context and creates a semantic representation of the system response. It performs many tasks like maintaining the history of dialogue, adopting certain dialogue strategies, dealing with deformed and unrecognized text, retrieving the contents stored in files or database, deciding the best response for users, managing initiative and system responses,

handling issues of pragmatics, performing discourse analysis, it also performs grounding etc. For these tasks, dialogue manager has many components. These are dialogue model, user model, knowledge base, discourse manager, reference resolver and grounding module

2.2.6 TASK MANAGER: The task manager assists other components in recognition of user's intention and in execution of problem solving steps with respect to the task at hand. It answers queries about objects and their role. Intention recognition services are usually used by the Interpretation Manager (IM). The task manger also provides a generic interface to task-specific agents that do the execution.

2.3 CLASSIFICATION OF SPOKEN DIAOLUGE SYSTEM

On the basis of method used to control dialogue, a dialogue system can be classified in three categories [14]:

2.3.1 FINITE STATE OR GRAPH BASED SYSTEMS: In this type of system the user is taken through a dialogue consisting of a sequence of predetermined steps or stages. The flow of dialogue is specified as a set of dialogue states. Following is the example:

System: Please Say your destination?
User: Kokrajhar
System: Is it Kokrajhar
User: Yes
System: Which date you want to travel
User: 15-02-2017
System: Did you say 15-02-2012
User: Yes

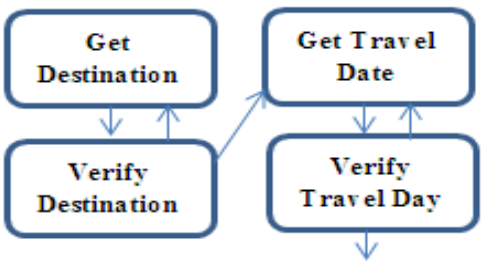

Figure 2.3: **Example of Finite State System**

Advantages

- Simple to construct
- The required vocabulary and grammar for each state can be determined in advance.

Disadvantages

- Dialogues are not natural
- Do not allow over-informative answers
- Inhibits the user's ability to ask questions and take initiative.

2.3.2 FRAME BASED SYSTEMS: Frame Based systems use template filling from user response. In this system user is asked questions that enable the system to fill slots in a template in order to perform tasks. The flow of dialogue is not predetermined but depends upon the content of user input and the information the user has to elicit. In the following example, we can see there are two different dialogues in first the dialogue goes like finite state system. In second dialogue user provide over information in response to a question, but the system fills its slots from the user's input and asks for the remaining information. This is how frame based systems works.

System: What is your destination?
User: Kokrajhar
System: What day you want to travel
User: Monday

System: What is your Destination?
User: Kokrajhar on Monday, 05 February 2014 around 9 in the morning
System: I have following connection….

Advantages

- Allow more natural dialogues
- User can provide over informative answers

Disadvantages

- These systems can't handle complex dialogues
- Range of application is limited to the systems that elicit information from users and act on the basis on the same

2.3.3 AGENT BASED SYSTEMS: These systems allow complex communication among the system, the user and the application in order to solve some problem or task. The interaction is viewed as interaction between two agents, each of which is capable of reasoning about its own actions and beliefs. The dialogue model takes the preceding context into account. The dialogue evolves dynamically as a sequence of related steps that build on top of each other.

Advantages

- Allow natural language in complex domain
- User friendly, like talking to human

Disadvantages

- These systems are hard to build
- The agent itself are usually very complex

2.4 EVALUATION METHODOLOGY

The goal with any evaluation is always to answer a set of questions, and it is the questions asked which determine how an evaluation should be performed. There are multiple of such evaluations that are appropriate for different goals identified by Hirschman and Thompson in 1997.

ADEQUACY EVALUATION: This evaluation type finds the goodness of the system or the component of a system. This evaluation process also identifies what is required to do, relative to the tasks and users at hand. For performing this evaluation process, the system should know how adequacy evaluation requires considerable knowledge of by whom, how and where the system is going to be used [18].

DIAGNOSTIC EVALUATION: This evaluation processes how well the system, or a component of a system, works relative to a closed set of input variables. In speech components where coverage is important, a common development methodology employs a large test suite of exemplary input **[18].**

PERFORMANCE EVALUATION: The goal is to get a measurement of the performance of a system, or a component of a system, in one or more specific areas. One usually defines what one is interested in evaluating, which specific property of system which reflects the criterion, and how to assess the value returned for a given measure [18].

2.5 ERROR HANDLING IN SPOKEN DIALOGUE SYSTEM

Since the performance in spoken language technologies such as ASR and SLU have been improved, spoken dialog systems can be developed now for many different application domains. Nevertheless, there are major problems for practical spoken dialog systems. One of them which must be considered by the dialogue manager is the error propagation from ASR and SLU modules. In general, errors in spoken dialog systems are prevalent due to errors in speech recognition and language understanding. The user's input may be unclear or incomplete because some or all of the words are incorrectly recognized or even though all the words are correctly recognized, the SLU module does not capture all the correct meanings due to data sparseness or ambiguity. These errors can cause the dialog system to misunderstand a user and in turn lead to an inappropriate response. To avoid these errors, a basic solution is to improve the accuracy and robustness of the recognition and understanding processes. However, it has been impossible to develop perfect ASR and SLU modules because of noisy environments and unexpected inputs. Therefore, error handling is also an active research topic in the dialogue management problems to improve the performance of the spoken dialog systems against ASR and SLU errors [19]. Examples of error Recovery strategies are shown in **Table2.1**

Table 2.1: Error Recovery Strategies

Error Recovery Types	**Example**
Explicit Confirmation	User: I want to know my final semester result. System: So, do you want to know your final semester result. Please answer yes or no User : yes

Implicit Confirmation	User: I want to know my final semester result. System: What is your roll number? User : 12345
Static Rephrasal	User: I want to know my final semester result System: Sorry , Please repeat User: I want to know my final semester result.
Dynamic Rephrasal	User: I want to know my final semester result. System: Sorry. Would you please rephrase using expression I'd like to know my [specify semester] result. User : I'd like to know my 8th semester result

2.6 CHALLENGES IN SPOKEN DIALOGUE SYSTEM

To build dialogue system developers faces many difficulties. These are mainly due to the lack of computer's understanding of natural language. This problem gives rise to many challenges for developers e.g. problem of Anaphora Resolution, Inferences, Ellipsis, Pragmatics, Reference resolutions and Clarifications, Inter sentential Ellipsis etc. [14]. Besides these language problems, other challenges are to design system prompts, grounding, detection of conflicts and plan recognition etc. In spoken dialogue systems, the problem related to utterance of the user, occur like ill formed utterances. These are the some of the challenges that developers have to take care of at designing time [20]. Following are some key problems for building a spoken dialogue system.

a. Lack of expertise: Development and research of spoken dialogue systems currently require technical expertise across several subject areas. Because of these requirements, development and research is currently limited to a few specialized laboratories.

b. High development costs: The development process is lengthy and expensive, often requiring months or even years to produce a spoken language application. Data collection for training recognizers and for building language and dialogue models is costly and often must be done via wizard-of Oz simulation, with humans attempting to mimic the performance of a spoken language systems.

c. Lack of portability: Current spoken language systems technology is not very portable, i.e. the technology cannot support adequately the development of new applications with acceptable performance without significant engineering for each new task.

The combination of these problems severely hinders application development and limits the role that spoken dialogue technology can play in key areas such as research and education.

2.7 EXAMPLES OF SPOKEN DIALOGUE SYSTEM

JUPITER: JUPITAR [21] is a weather information system developed at MIT Laboratory. Jupiter can answer questions about general weather forecasts, as well as information on temperature, wind speed, humidity, sunrise time, and advisories. Jupiter can also tell which cities it knows about in a particular region. It has been built on top of their spoken dialogue platform called GALAXY, which they have also used for other domain areas. Jupiter is a conversational system that provides up-to-date weather information over the phone. Jupiter knows about 500+ cities worldwide of which 350 are within the US and gets its data from four different Web-based sources.

CSLU Toolkit: The CSLU Toolkit [22] was created to provide the basic framework and tools for people to build, investigate and use interactive language systems. These systems incorporate leading-edge speech recognition, natural language understanding, speech synthesis and facial animation. It is a tool to enable exploration, learning, and research into speech and human-computer interaction. They have made a free toolkit for creating spoken language interfaces. CSLU toolkit provides the basic framework and tools for people to build, investigate and use interactive language systems.

TRIPS: The Rochester Interactive Planning System (TRIPS) are the latest in a series of prototype collaborative planning assistants developed at the University of Rochester's Department of Computer Science. The goal of the project is an intelligent planning assistant that interacts with its human manager using a combination of natural language and graphical displays. The system understands the interaction as a dialogue between it and the human. The dialogue provides the context for interpreting

human utterances and actions, and provides the structure, for deciding what to do in response. With the human in the loop, they and the system together can solve harder problems faster than either one could solve alone [23].

SUMMARY

This chapter gives a brief overview of spoken dialogue systems, with special emphasis on speech recognition and dialogue management. We start by reviewing some key concepts that are particularly relevant for our work: overview, its importance and characteristics of spoken dialogue system etc. A proper understanding of these aspects is indeed a prerequisite for the design of spoken dialogue system in Bodo language. The basic architecture of spoken dialogue system typically comprise multiple processing components, from speech recognition to understanding, dialogue management, output generation and speech synthesis. We briefly describe the role of each component and their positions in the global processing pipeline. Graph based, frame based and agent based are three different types of spoken dialogue system. The advantage and disadvantage of these types are stated in this chapter. Last but not least, the final section of this gives the light on its evaluation methods and error handling mechanisms. Lack of expertise, high development costs and lack of portability are some of the key challenges in building spoken dialogue system. JUPITER, CSLU toolkit, TRIPS is some of examples of widely used spoken dialogue systems.