CHAPTER 2

SPOKEN DIALOGUE SYSTEMS
Thoughts and ideas are meaningful and can be realized when they can communicated to others and spoken language is one of the human being’s main characteristic by which he can communicate his ideas to other and collaboratively realize them in practice other than in the written form of language. The speed and ease with which we can speak is comparatively more than any other form of interaction like key presses or mouse movements. But in real life when we speak we unknowingly embed lot of information in the form of pitch, intensity, temporal variation other than the linguistic word lattices which build our utterances during a dialogue with other human. And humans being an intelligent entities decode not only the linguistic information but also the auditory information and uses the previous dialogue context to generate a response or decide about the next move in the dialogue process. The ability of understanding and producing more or less coherent answers to spoken utterances implicitly defines different degrees of intelligence which most humans demonstrate in all situations. But intelligence has a closely relationship with learning.

Intelligent systems are agents which are capable of acting rationally towards any change in environment based on the valid information it has. Machines like computers can be intelligent if they are able to act rationally towards any change in environment by applying the knowledge it carries in its knowledge-base. Since humans use mainly spoken language and natural language for interaction with other human, machine to be human-like have to also use the spoken language as its means of input and output for interaction. There are many issues in the design of a spoken dialogue system. and two key assumptions are almost always taken.

i. Dialogues with exactly two participants are considered.

ii. All interactions between the system and the user are in the form of turns. A turn in a dialogue is a period in which one of the participants has a chance to say something to the other participant.

Under these assumptions, the dialogue will follow a cycle, known as the dialogue cycle. One of the participants says something, this is interpreted by the listener, who makes a decision about how to respond and the response is conveyed back to the original participant. This same process of listening, understanding, deciding and responding then occurs on the other side and the cycle repeats. Hence any dialogue system requires a number of components: one that can understand the user’s speech, one that makes decisions and one that produces the system’s speech.
2. SPOKEN DIALOGUE SYSTEMS

2.1 History

Machines producing speech is not new. Since 17th century mathematicians and logicians who designed the first computational machines had the thought that machines could speak which was clear from the Rene Descartes declaration from “Discourse on the method” that if machines bearing the image of our bodies and capable of imitating our actions, we may easily conceive a machine which emits vocables and even acts in response to change in its organs so as to reply what is said in its presence. Then Lenord Euler in 1761 said that, “It would be a considerable invention indeed that of a machine able to mimic speech with its sounds and articulations. I think its is not impossible”. In 1779 Christain Kratzenstein designed a machine which was able to produce vowel sound and was based on the human vocal tract. In 1791, Wolfgang Von Kempelen built the machine also known as first talking machine which was not only able to produce sounds but also words and even short sentences in Latin, Italian and French languages. At the end of 1878 Alexander Graham Bell invented telephone which was based on his inspiration to invent a machine that could transcribe spoken words into text which he could not complete.

While AT&T Bell Laboratories developed a primitive device that could recognize speech in the 1940s, researchers knew that the widespread use of speech recognition and understanding would depend on the ability to accurately and consistently perceive subtle and complex verbal input. In 1939 Homer Dudley of Bell Labs invented a controlled speech synthesizer but it required highly trained technicians to use it. In 1942 a toy dog which responded to its name was produced by Elmwood Button Company that created a landmark in the field of speech recognition. Since then lot of research in the parallel and inter-disciplinary fields have contributed to this area of speech analysis and synthesis in building machines which could respond to speech signal as a mean of interaction. In 1952, Bell Labs developed a system which recognized spoken digits transmitted by a phone with an accuracy of 98 % with speaker adaptation [Davis et al., 1952]. In 1959, a speaker independent system able to recognize vowels with an accuracy of 93% was developed by Forgie and Forgie at MIT. A system capable of matching spoken utterance to a list of 50 words with an accuracy of 83 % along with a confidence score to indicate the recognition result was developed by Ben Gold of MIT in 1966. In 1956, Noam Chomsky developed many theories about linguistics and
2. SPOKEN DIALOGUE SYSTEMS

computational grammars and built the foundations of Natural Language Processing. In 1950 Claude E. Shannon, also known as father of information theory used the concept of artificial intelligence to develop a chess playing software by a machine. Thus, in the 1960s, researchers turned their focus towards a series of smaller goals that would aid in developing the larger speech recognition system. As a first step, developers created a device that would use discrete speech, verbal stimuli punctuated by small pauses. In 1966 Joseph Weizenbaum from MIT developed ELIZA, the first artificial intelligence program which simulated human conversation and passed Turing Test to prove the intelligence quotient. In 1968, Arthur C. Clarke and Stanley Kubrick created HAL9000 a computer that could hold a conversation, think and adapt its behavior.

However, in the 1970s, continuous speech recognition, which does not require the user to pause between words, began. Also lot of research was funded in the speech understanding program. It aimed at analyzing, storing and understanding continuous speech by the computer systems. This led to lot of research groups at many leading universities like MIT, Stanford, Carnegie Mellon University and other research institutions like Microsoft and IBM to ponder on the different issues of speech understanding. More focus in these days was on Automatic Speech recognition, where the recognition error rates high because of the smaller vocabularies. It was because of the statistical and empirical pattern matching frameworks based on Hidden Markov Models used by James, Janet Baker and Fedreick Jelinek who actually got a break though in the area of statistical pattern matching framework based on Hidden Markov Models to speech recognition [Jelinek, 1976]. In the same years, due to technology improvement especially memory and Processesing power of the computers, the structure of human discourse was the main theme for investigation by researchers for making human computer interfaces more friendly [Rabiner and Schafer, 1978]. In 1974 Barbara Grosz studied the structure of dialogues in collaborative tasks [Grosz, 1974]. Speech Recognition Systems have become so advanced and mainstream that business and health care professionals are turning to speech recognition solutions for everything from providing telephone support to writing medical reports. Technological advances have made speech based systems and devices more functional and user friendly, with most contemporary products performing tasks with over 90 percent accuracy. In 1986, Barbara Grosz and Candace Sidner developed the theory of centering [Grosz and Sidner, 1986] that aimed to formalize the way a human follows the focus of a conversation. James
Allen applied statistical pattern matching techniques usually applied in speech recognition to semantic parsing of natural language [Allen, 1987]. In 1990’s hybrid methods combining Artificial neural networks and HMM’s were successfully used in large speech recognition systems. [Bourlard and Morgan, 1994] In 1996, the development of a complete spoken dialogue systems (SDS) which including automatic speech recognition, Natural Language understanding, dialogue management and speech synthesis started to emerge.

Today, the latest generation of speech technology delivers conceptual search. This approach utilizes advanced mathematics and complex algorithms to derive meaning from speech. Conceptual search addresses the shortcomings of previous speech technology models and provides the most accurate way of recognizing and finding speech because it understands what is being said. It can distinguish between homophones, heteronyms, as well as find and group things by concept. It can also find related information based on meaning and has lower computational need than some of the earlier generations of speech recognition technology. According to the industry, Satisfying the needs of consumers and businesses by simplifying customer interaction, increasing efficiency, and reducing operating costs, speech based software is used in a wide range of applications. Indeed, recent advances in spoken dialogue systems are creating a dynamic environment, since this technology appeals to anyone who needs or wants a hands-free approach to computing tasks. As the merger of large vocabularies and continuous recognition continues, look for more and more research is taking place toward speech based systems and researchers are developing new gadgets with this technology. Today, the latest generation of speech technology delivers conceptual search. This approach utilizes advanced mathematics and complex algorithms to derive meaning from speech. Conceptual search addresses the shortcomings of previous speech technology models and provides the most accurate way of recognizing and finding speech because it understands what is being said. It can distinguish between homophones, heteronyms, as well as find and group things by concept. It can also find related information based on meaning and has lower computational need than some of the earlier generations of speech recognition technology. According to the industry, Satisfying the needs of consumers and businesses by simplifying customer interaction, increasing efficiency, and reducing operating costs, speech based software is used in a wide range of applications. Indeed, recent advances in spoken dialogue systems are creating a
dynamic environment, since this technology appeals to anyone who needs or wants a hands-free approach to computing tasks. As the merger of large vocabularies and continuous recognition continues, look for more and more research is taking place toward speech based systems and researchers are developing new gadgets with this technology.

2.2 What is a Human - Computer Dialogue ?

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 agent is a computer and the other is human, the dialogue is known as Human- Computer Dialogue. Also When the system initiates the dialogue and always prompts the user to select an utterance from fixed menus it is known as system initiative dialogue system. When the human and the machine makes a more natural dialogue where the system attempts to determine the intentions of the user from the unrestricted utterances, the dialogue system is known as mixed initiative dialogue system. But when other mean of communication like facial expressions, prosodic information etc. other than speech is used in the interaction its known as multi-modal dialogue. When the human machine dialogue is dedicated to the realization of a particular task or set of tasks, the dialogue system is known as task oriented dialogue system. When the agent in a dialogue is a spoken dialogue system, the user and the system exchanges a series or utterances where each spoken utterance is the acoustic realization of the intentions and concepts embedded in the form of word lattice that one of the agents wants to communicate to the other. Human-to-computer interaction is an 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.

2.3 Levels in a Speech based Interaction

Information conveyed by speech can be analyzed at several levels. In the field of Natural Language Processing, seven levels are commonly admitted in order to describe speech-based communication [Boite et al., 2000]. These levels can be classified into high and low levels of description, the lower level starts from the physical sound signal. This
distinction between high and low levels is applicable to all types of communications as there is always a possibility to distinguish the physical stimuli and the interpretation.

(1) **The Acoustic Level**

Speech is a sequence of sounds which may also be defined as a variation of the air pressure created by the vocal tract. The acoustic level concerns the signal and as such represents the lowest level of speech communication. The study of the acoustic signal includes the study of any of its representation as the electrical output of a microphone (analog or digital), wave forms, frequency analysis (Fourier transforms, spectrograms) etc. Useful information can be obtained from the analysis of the acoustic signal such as the pitch (fundamental frequency), the energy and the spectrum. In general, it is the only level considered by speech coding techniques. As the human vocal tract is a physical instrument, it is subject to a certain inertia and thus, it cannot assume sudden state modifications. This results in an important property it can be considered as a pseudo-stationary signal.

(2) **The Phonetic Level**

It is a low level description where the main focus is on the production of particular sounds by the articulatory system. The phonetics studies how humans voluntary contracts muscles in order to dispose obstacles like tongue, lips, teeth and other organs in the aim of pronouncing a specific sound.

(3) **The Prosodic Level**

The main task at this level is the analysis of a limited number of distinct sounds allowed in a particular language (phonemes), the rhythm with which they are produced in a sequence, the musicality applied to this sequence (prosody) and he accentuated part within the sequence. This level is considered to be transitory between low and high levels as it concerns physically observable features of the signal but those specific traits are voluntary produced by the speaker in the aim of including meaningful clues into the speech signal. Prosody is used to detect the sentiments and emotions in the speech signals like tutoring applications [Litman and Forbes, 2003].
2. SPOKEN DIALOGUE SYSTEMS

(4) **The Lexical Level**
Also known as morphological level, the main focus at this level is on all the valid phoneme sequences that produce words included in the lexicon of a particular language where each phonemes are the finite number of different sounds in a specified language. This level forms the first stage for the high level where word elementary sub-units are studied which convey sense.

(5) **The syntactic Level**
Words constitute a valid sentence in a language only when the word chain follows the set of rules also known as the grammar of the language or syntax of the language. There are different rule sets to describe the syntax. The main function of the grammar is to make a word function in a sentence so that the sentence follows the syntactic structure. Computational grammars which are different from the linguistic grammars have been developed in the early ages of natural language understanding [Chomsky, 1965].

(6) **The Sematic Level**
At this level, the main focus is to determine the context independent meaning that the words in the sentence mean and how those meanings combine to form the information that the sentence try to convey. Although an utterance may be syntactically correct but it might not provide the coherent information for which it was framed. So this level studies how to extract the meaning/sense from the utterances.

(7) **The Pragmatic Level**
Pragmatics is the study of grouping all the context dependent information in a dialogue. Most often the utterances implicitly refer to the underlying information also known as ground information. which is expected to be known by the participants of the conversation either based on the environmental conditions, the beliefs that the participants hold, their background, common knowledge that they hold. Pragmatic level is divided into three sublevels [Allen et al., 1987].

**Pure pragmatic level**
the study of the different meanings that can convey a single sentence uttered in different contexts.
2. SPOKEN DIALOGUE SYSTEMS

**Discourse Level**

Concerns how the directly preceding sentence affects the interpretation of the next sentence. The study at this level helps in disambiguation and anaphora resolution.

**World Knowledge Level**

Also known as ground knowledge includes all the information people know about the world and what an participant in conversation knows about the other participants belief and goals.

### 2.4 General Dialogue System

In human to human conversation, the conversant tries to integrate all the information from the senses based on the knowledge.

### 2.5 Spoken Dialogue System

Spoken dialogue interaction has been suggested by researchers and practitioners as a promising alternative way of communication between humans and machines [Zue et al., 2000]. A compelling motivation is the fact that conversational speech is the most natural, efficient, and flexible means of communication among human beings. Because of the complexity of human-human interaction, human-machine conversations need to be much simpler. Talking to a machine requires a spoken dialogue system. These systems may be alternatively referred to in the literature as “conversational agents”, “Spoken language systems” or “conversational interfaces” [Jurafsky and Martin, 2008], [McTear, 2004] Huang et al., 2001. Such systems should be able to understand what a person says, take an appropriate action, and then provide a response. Ideally, spoken dialogue systems should yield successful, efficient and natural conversations within a given domain. However, 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 which is normally to access and manage information. These systems are an example of an open ended, goal oriented, real time interactions between humans and computers. A Multi Modal Spoken Dialogue systems is one which uses many modes of input like speech, Graphic User
Figure 2.1: Architecture of a general Dialogue System - The figure shows different processes involved in a human to human interaction.
Interface and computer vision e.g. In case of Telephonic technical support for product and services, In-car music control for music navigation, Tutoring, Language learning, Mobile search interface, Computer based assistance technology especially in Elder care, Automated receptionist. Voice enabled interfaces are now becoming common and most of us have used such interface while dialing the number of a contact using his speech tag, using voice recognition software for typing our documents in a word Processing software, browsing the internet using voice enabled internet browsing software which accepts our voice commands and hear the emails in the inbox along with their contents. In general the classification of spoken dialogue systems depends on the application and its complexity and are becoming ubiquitous due to their rapid improvement in performance and decrease in cost. The spoken dialog systems receive speech inputs from the user, and the system responds with the required action and the information. For example, a user might use a spoken dialog system to reserve a flight over the phone, to direct a robot to guide him to a specific room, or to control in-car devices such as a music player or a navigator. Since the early 1990s, many spoken dialog systems have been developed in the commercial domain to support a variety of applications in telephone-based services. For example, early spoken dialog systems functioned in restricted domains such as telephone-based call routing systems (HMIHY) [Gorin et al. 1997], weather information systems (JUPITER) [Zue et al. 2000], and travel planning (DARPA communicator) [Walker et al. 2001]. More recently developed systems are used in incar navigation, entertainment, and communications [Minker et al. 2004; Lemon et al. 2006; Weng et al. 2006]. For example, the EU project TALK2 focused on the development of new technologies for adaptive dialog systems using speech, graphics, or a combination of the two in the car. More recently, multi-domain dialog systems have been employed in real life situations [Allen et al. 2000; Larsson and Ericsson 2002; Lemon et al. 2002; Pakucs 2003; Komatani et al. 2006]. Such multi-domain dialog systems are now able to provide services for telematics, smart home, or intelligent robots. These systems have gradually become capable of supporting multiple tasks and of accessing information from a broad variety of sources and services.
Figure 2.2: Architecture of a Spoken Dialogue System - The figure shows different modules involved in a human to computer interaction.
2.6 Characteristics of spoken dialogue system

2.6.1 Turn-taking

A dialogue consists of many turns, where in every turn one participant speaks. Who should speak in the next turn is determined by using the Turn-Taking rules at the end of each turn. These rules apply at a Transition-Relevance place (TRP). Transition relevance places where the structure of the language allows speaker shift to occur. Here is a simplified version of the turn-taking rules, grouped into a single three-part rule. At each TRP of each turn:

1. If during this turn the current speaker has selected A as the next speaker then A must speak next.

2. If the current speaker does not select the next speaker, any other speaker may take the next turn.

3. If no one else takes the next turn, the current speaker may take the next turn.

2.6.2 Grounding

Grounding is a collective act performed by the speaker and the hearer wherein both must constantly establish common ground. By common ground we mean the set of things that are mutually believed by both speakers. The need to achieve common ground means that the hearer must ground or acknowledge the speaker’s utterances, or else make it clear that there was a problem in reaching common ground. For example

U: I want to fly from Boston.
S: Mm hmm. (Acknowledgement)

The word mm-hmm here is a continuer, also often called a backchannel or an acknowledgement token. A continuer is a short utterance which acknowledges the previous utterance in some way, often cueing the other speaker to continue talking (Jefferson, 1984; Schegloff, 1982; Yngve, 1970). Clark and Schaefer (1989) discussed five main grounding methods, ordered from weakest to strongest:
• **Continued attention:** B shows she is continuing to attend and therefore remains satisfied with A’s presentation. Relevant next contribution: B starts in on the next relevant contribution.

• **Acknowledgement:** B nods or says a continuer like “uh-huh”, “yeah” an assessment like “that’s great”.

• **Demonstration:** B demonstrates all or part of what she has understood A to mean, for example by reformulating A’s utterance or by collaborative completion of A’s utterance.

• **Display:** B displays verbatim all or part of A’s presentation.

Some Grounding examples:

U: ...I need to travel in June.
S: And, what day in May did you want to travel? (Display)
U: I want to be in New Delhi in time for the Asian games.
S: <reject>. Sorry, I didn’t get that. Please say the month and day you would like to leave.(Rejection)
U: I want to go on October fifteenth.

### 2.6.3 Conversational implication

The interpretation of an utterance relies on more than just the literal meaning of the sentences. For example:

S: ...And, what day in May did you want to travel?
U: OK uh I need to be there for a meeting that’s from the 12th to the 15th.

Here the user does not in fact answer the question. The user merely states that he has a meeting at a certain time. In this case the speaker seems to expect the hearer to draw certain inferences; in other words the speaker is communicating more information than seems to be present in the uttered words. Grice proposed that what enables hearers to draw these inferences is that conversation is guided by a set of maxims, general heuristics which play a guiding role in the interpretation of conversational utterances. He proposed the following four maxims:
2. SPOKEN DIALOGUE SYSTEMS

- Maxim of Quantity: Be exactly as informative as is required:
  a. Make your contribution as informative as is required
  b. Do not make your contribution more informative than required.

- Maxim of Quality: Try to make your contribution one that is true:
  a. Do not say what you believe to be false.
  b. Do not say that for which you lack adequate evidence.

- Maxim of relevance: Be relevant

- Maxim of Manner: Be clear, brief and orderly

2.7 Components of a Spoken Dialogue System

The general spoken dialogue system integrates four main components to process the speech signal from the user in presence of environmental noise and the system can generate the output which can be either visualized on the screen or synthesized by a text to speech synthesis module or a pre-recorded audio. This process works iteratively to complete the dialogue process wherein the intended purpose of the user is achieved. The components involved in the dialogue process are :-

2.7.1 Speech Recognition

The users makes a verbal response which is usually speech signals with noises which are recognized by an automatic speech recognition (ASR) subsystem which transforms the speech waveform into a sequence of parameter vectors which are then converted into a sequence of word (text). Most of the speech recognition methods uses Hidden Markov Model (HMM) to estimate the most probable sequence of words from a given speech signals. This component is built using many available toolkits ATK/HTK [Young et al., 2000] and SPHINX packages [Walker et al., 2004]. The performance of the speech recognition engine will depend on the difficulty of the task and on the amount of in-domain training data. The error rates are higher in the limited-domain dialogue systems and the user speaks freely using words which are out of the bounds of the list. [Raux et al., 2006] describes the Let’s Go! bus information system, which has a sentence average word error rate of 64%. The word error rate of the ITSPoke an
intelligent tutorial system is 34.3% [Litman and Silliman, 2004].

Most current spoken dialogue systems use only the most likely hypothesis of the user’s speech. State-of-the-art recognisers can however, output a list of hypotheses along with associated confidence scores. This list is called an N-best list, where N denotes the number of hypotheses. The confidence scores give an indication of the likelihood that the recogniser attaches to each word sequence. Ideally these confidence scores will give the posterior probability of the word sequence given the audio input [Jiang, 2005]. In some cases the recogniser may also return a word-lattice to represent the set of possible hypotheses. Such word lattices may be converted into N-best lists by first converting the lattice to a confusion network and then using dynamic programming to find the minimum sum of word-level posteriors [Evermann and Woodland, 2000].

2.7.2 Natural Language Understanding

This unit analyses the textual form for the set of hypothesis of the user utterance to understand the meaning of these words with the main aim to determine what the user wants to achieve by saying the words e.g morphological analysis, part-of-speech tagging, and shallow parsing. The NLU module maps the pre-processed utterance to a meaning representation (e.g., semantic frame) from which the dialogue act, user goal, and named entities are extracted by semantic parser. e.g, whether the user says “I’d like to know the doctor in the orthopaedics,” or “who is the orthopaedician on duty” the desired outcome is the same. The user is asking for about the doctor on duty in orthopaedics department. The fact that the first utterance is a statement and the second is a question is irrelevant. This distinction between the exact semantics of an utterance and it’s purpose was first made explicit in the definition of a speech act, which is a representation of this underlying action [Austin, 1962], [Searle, 1969]. In the example above, the speech act for both utterances would be “request”.

The speech acts has been extended in the case of dialogue to include actions relating to turn-taking, social conventions and grounding [Traum, 1999]. The resulting concept is called a dialogue act tag. Dialogue act tags also allow actions such as “confirm” and “affirm” for confirmations and affirmations. e.g, a “confirm” action might be used to represent “Did you say you wanted to see an orthopaedician” and an “affirm” act might be used to represent “Yes!” In the traditional definitions of both speech and dialogue acts, the semantic information is completely separated from the act. A
simplified form of semantic information is clearly an important input to the dialogue system. In the case of the user asking for the doctor on duty the information should be represented to indicate what is being requested. It is necessary to represent the dialogue act as “request(doctor)”. Similarly, the confirmation case above the dialogue act may be represented as “confirm( doctor=orthopaedic )”. Mostly the Dialogue acts are therefore represented as the combination of the dialogue act type followed by a (possibly empty) sequence of dialogue act items.

\[ \text{dialog\_act\_type}(a = x, b = y, \ldots) \]

The \text{dialog\_act\_type} denotes the type of dialogue act while the act items, \( a = x, b = y, \ldots \) will be either attribute-value pairs such as doctor=orthopaedic or simply an attribute name or value e.g request(addr), meaning “What is the address?” and inform(well), meaning “I am well”. With the concept of dialogue acts in hand, the task of understanding the user becomes one of deciphering dialogue acts. This is known as semantic decoding. In general one could imagine doing this on the basis of several sensory inputs. The prosodic information such as pitch or intensity of a user’s utterance might give some indication as to the dialogue act type.

There are a wide range of techniques available for semantic decoding. Hand-crafted techniques which include template matching and grammar based methods. Data-driven approaches include the Hidden Vector State model [He and Young, 2006], machine translation techniques [Wong and Mooney, 2007], Combinatory Categorial Grammars [Zettlemoyer and Collins, 2007], Support Vector Machines [Mairesse et al., 2009] and Weighted Finite State Transducers [Jurcicek et al., 2009]. Most semantic decoders will assign exactly one dialogue act for each possible word sequence obtained from the speech recogniser. In the case of ambiguities, however, the semantic decoder may choose to output a list of the most probable outputs along with associated confidence scores. Since the speech recogniser is producing an N-best list of word sequences, some method must be found for combining the confidence scores from the speech recogniser with those of the semantic decoder.

2.7.3 Dialogue Management

After the utterances are semantically decoded, the system must choose an appropriate response from a set of alternatives based on some strategy. The component which
makes these decisions is called the dialogue manager. The response chosen by the
system is encoded as a dialogue act and is known as the system action or system act.
The chosen response is selected from a set of possible actions, $a \in A$ and will depend
on the input that the system receives from the semantic decoder. This input is called
the observation, labelled $o \in O$, since it encodes everything that the system observes
about the user. Choosing the best action requires more knowledge than simply the
last observation. The dialogue manager coordinates the activity of all components,
controls the dialogue flow, and communicates with external applications. The dialogue
manager should play many roles which include discourse analysis, knowledge database
query, and system action prediction based on the discourse context and dialogue history
which plays an important role. The dialogue manager takes this into consideration
by maintaining an internal representation of the full observation sequence. This is
called the dialogue state, system state or belief state and is denoted by $b \in B$. The
current belief state will depend on a belief state transition function which is a mapping
$\delta : A \times O \times B \rightarrow B$ which takes a given belief state and updates it for each new
observation and system action.

The component of the dialogue manager which defines its behaviour is the dialogue
policy or dialogue strategy ($\pi$). The policy determines what the system should do in
each belief state. In general, the policy will define a probability distribution over which
actions might be taken. If $\pi(A)$ denotes the set of these distributions then the dialogue
policy will be a mapping from belief states to this set, $\pi : B \rightarrow \pi(A)$. Clearly the
actions, belief states and observations are all indexed by the turn number. When it is
important to note the time step being considered, they are denoted $a_t, b_t$ and $o_t$. While
the system is in state $b_t$ it will choose action $a_t$ according to the distribution determined
by the policy, $\pi(b_t)$. The system then observes observation $o_{t+1}$ and transitions to a
new system belief state $b_{t+1}$. When exact point in time is insignificant, the $t$ is omitted
and a prime symbol is used to denote the next time step (e.g. $o = o_{t+1}$).

### 2.7.4 Speech Synthesis System (TTS)

The system responses have to be finally conveyed to the user. The system dialogue
acts are first converted to natural language with a list of content items from a part of
the the knowledge base that keeps track of all the information generated through the
dialogue history, which is queried and/or updated by the natural language generator
and finally passed to Speech Synthesis system which conveys the message as audio. The simplest approach for natural language generation is to use templates. As an example, a template might transform “inform(doctor=x)” into “The doctor is x”, where “x” may be replaced by any name of the doctor which will be queried from the database. Templates have proven to be relatively effective for natural language generation, since the number of system dialogue acts is reasonably tractable. More complex approaches have also been developed.[Mairesse and Walker, 2007]. The most common approach used for the text to speech synthesis is the unit selection approach, which splices segments of speech from a database to generate sound for a given word sequence and other method is based on Hidden Markov’s model.

The process iterates until one of the conversant (user or machine) terminates the dialogue.

2.8 User Simulation

It is essential to test a Spoken Dialogue System with user dialogues but it is a difficult and time consuming exercise to generate the possible dialogues and then test the dialogue manager with human users. Simulated environments provide one way of speeding up the development process by providing a more efficient testing mechanism.

![Figure 2.3: Dialogue Act showing User simulator instead of User - A Graphical representation showing the user simulator and error simulator instead of the user in the dialogue act.](image)
A simulated environment generates situations that the dialogue system designer will not have thought about and the system can be refined to handle them. Dialogue managers that are built using techniques from machine learning can learn automatically what actions to take and for these systems simulated environment is particularly important as the system can be boot-strapped by learning from interactions with the simulator. Further refinements obtained from real interactions with human users make the dialogue manager act like humans do in various situations. A user simulator generates dialogue acts given the past dialogue history, as if it were human. This is passed through an error simulator which generates appropriate confusions and confidence scores. The user simulator operates on the dialogue act level as shown in figure 2.3 which graphically represents the user simulator instead of the human user in the dialogue act.

There are also simulation environments which have been built to operate at a word-level [Jung et al., 2009], [Schatzmann et al., 2007]. In this case, the simulated dialogue act is used to generate a word-level form, which is passed to the error-simulator to produce word-level confusions. This is then passed to the semantic decoding component of the spoken dialogue component as in the case of the human machine conversation.

The data-driven simulation techniques which are available and used for user simulation are as follows:

- Bigram models.
- Goal-based models.
- Conditional random fields (CRF Models).
- Hidden agenda models.

A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies is given in [Schatzmann et al., 2006].

### 2.9 Dialogue Manager Design Paradigms

In Spoken Dialogue System design, the most challenging task is to build an effective dialogue manager which can deal with the uncertainty as well follow policies which can optimally learn from the dialogue history automatically. There are four different paradigms for building the dialogue manager as given below.
2.9.1 Hand Crafted Approach

The dialogue manager at the most basic level can be defined by the concepts of belief state, state transitions and policy. In the hand crafted dialogue management framework, the system designer directly define all of these components. As dialogues become more complex, the number of states, transitions, and policy decisions becomes very large so researchers have developed various techniques to facilitate the design process.

![Diagram showing four major paradigms in spoken dialogue management: Hand Crafted Approaches (HDC), Markov Decision Process models (MDP), Bayesian Network approaches (BN) and Partially Observable MDP approaches (POMDP) with respect to how they handle uncertainty and policy optimisation.]

Figure 2.4: Four major paradigms in spoken dialogue management - The figure shows four major paradigms in spoken dialogue management viz. hand-crafted approaches (HDC), Markov Decision Process models (MDP), Bayesian Network approaches (BN) and Partially Observable MDP approaches (POMDP) with respect to how they handle uncertainty and policy optimisation.

The simplest approach to represent the dialogue manager is with the help of a graph or flowchart, sometimes called the callflow [Pieraccini and Huerta, 2005]. Nodes in the graph represent belief states of the dialogue and define which action should be taken, while transitions are determined by the observations received. 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. Another approach is frame-based dialogue manager, also known as a form-filling dialogue manager [Goddeau et al., 1996]. Frame-based approaches assume a set of concepts that the user can talk about, called slots, which take on values from a pre-defined set. The current set of filled slots is included as part of the state, along with some hand-crafted information about how
certain the system is about each slot value. Dialogue management proceeds by using a pre specified action for each set of known slots. The ability for users to speak about any slot at any time allows for much more freedom in the dialogue, which is typically perceived as more natural for users. The frame based approach is most often used in information seeking dialogues, where a user is seeking information subject to a set of constraints. e.g health information system, which might have slots for the symptoms, time of start of the problem. The system would ask for slot-values until it decided that enough have been given, at which point it would offer information about a relevant doctor. A problem with frame-based approaches is that some of the slots will not be relevant for particular dialogues. Another issue is that dialogues are often composed of smaller sub-dialogues, and the progress in a dialogue often follows a particular agenda, which cannot easily be modelled using frames. This has led to the development of hierarchical and agenda-based dialogue managers. These approaches allow the definition of sub-dialogues which depend on the values of slots at higher levels. When the domain of discourse moves away from information-seeking tasks, then the agenda-based and frame-based approaches sometimes struggle to adapt.

Dialogue systems which are based on shared plans and not only what the user wants such as collaborative dialogues e.g. language learning, where both participants must work together to achieve a task are said to use plan-based dialogue managers [Rich and Sidner, 1998]. Another framework for dialogue management is the information state model [Bos et al., 2003] where in the dialogue acts correspond to dialogue moves and are used to update an information state subject to certain preconditions. The information state represents the accumulation of everything that has happened in the dialogue, and is used by the dialogue manager to choose its next action according to an update strategy such as logic programming [Fodor and Huerta, 2006].

All the above framework’s help in structuring the dialogue manager but these approaches don’t help in analyzing how to handle the uncertainty which is inherent in the dialogue process.

2.9.2 Sequential Decision Process 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 with states which represent un-
certainty as well [Bohus and Rudnicky, 2005]. But this model lacks the principled
definition for these states of uncertainty, so the alternative was sequential decision
process models which were natural and augmented a well researched framework based
on Markov Decision Processes (MDP’s) [Puterman, 1994], to aid the spoken dialogue
system to reliably identify the underlying environment state.

A spoken Dialogue System based on the 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 states determine the probability distribution over next possible states
and as such are modelled as a stochastic process.

2.9.2.1 Markov Decision Processes framework (MDPF)

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. In this thesis it is assumed that the system
can be in a finite number of states and the agent (Spoken Dialogue System) can choose
from a finite set of actions. Let $S = s_0, s_1, s_2, ..., s_N$ be a finite set of states. Each state
at discrete time $t \in T$ 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.

$$Pr(S^{t+1}|S^0, S^1, ..., S^t) = Pr(S^{t+1}|S^t)$$

(2.1)

The Spoken Dialogue System at each state execute one of the available action ($a$) from
a set of actions ($A$) which affects the state transition probabilities. Thus each action
$a \in A$ is fully transcribed by a $|S|X|S|$ state transition matrix whose entry in $i^{th}$ row
and $j^{th}$ column is the probability that the system will move from state $s_i$ to the state
$s_j$ if the action $a$ gets executed.

$$P^a_{ij} = Pr(S^{t+1} = s_j|S^t = s_i, A^t = a)$$

(2.2)
The effect of the actions $A$ on the system states $S$ is given by *transition function* ($T$) where $T : S \times A \rightarrow \Delta(S)$ which associates a probability distribution over the possible successor states. and $\delta(S)$ represents the set of probability distribution over $S$. Thus for each state $s$, $s'$ and $a \in A$ the function $T$ determines the probability of a transition from state $s$ to state $s'$ after executing action $a$.

$$T(s, a, s') = Pr(S^{t+1} = s' | S^t = s, A^t = a) \quad (2.3)$$

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 $R : S \times A \rightarrow \mathbb{R}$. The casual relationship between MDP states, actions and rewards is shown in the figure 2.5.

![Figure 2.5: Casual Relationships between MDP states, actions, rewards - $R^t$ is the reward received at time $t$](image)

The Markov Decision Process was first suggested as a dialogue model by [Levin and Pieraccini, 1997]. The system proposed used bi-gram language model for training and
optimized the reward using standard algorithms. [Walker, 2011] proposed an MDP based system (PARADISE framework) by using regression on the known features of the dialogue as a means to determine the reward the system should assign during each turn of the dialogue. This system along with various other dialogue systems have been successfully tested with the human users [Kearns et al., 2011]. But since the state transitions are mostly handcrafted by the system design spoken dialogue system based on MDP framework, the state set gets intractable when the complexity of the dialogue increases. For example the information state updates have been used in the MDP systems by adding the concept of rewards and Markov property (i.e. the system belief state depends only on its previous value and not on the history) to the standard information state model [Lemon et al., 2006]. There is some research work [Paek and Chickering, 2006] indicating that this may not necessarily be a valid assumption.

2.9.3 Partially Observable Markov Decision Process (POMDP) Framework

In order to act optimally, the spoken dialogue system must take all the previous history of observations and actions into account, rather than just the current state it is in. A POMDP is a generalization of MDPs in which system states are not fully observable. Partially observable Markov Decision Process (POMDP) were first suggested for dialogue by [Roy et al., 2000]. A POMDP framework is based on the underlying MDP extended with observation space $O$ and observation function $Z(.)$. In MDPs the dialogue system has the complete knowledge of the system states whereas in case of partially observable environments, observations are only probabilistically dependent on the underlying environment state. Also the same observations can be observed in different states which makes it difficult to determine the state of the system. Observation function $Z : S \times A \rightarrow \delta(O)$ specifies the relationship between the system states, actions and the observation space. Thus $Z(s', a, o')$ is the probability that observation $o'$ will be recorded after an agent performs action $a$ and moves to state $s'$. Thus

$$Z(s', a, o') = Pr(O^{t+1} = o'|S^{t+1} = s', A^{t} = a) \quad (2.4)$$

Formally POMDP is a tuple $< S, A, T, R, O, Z >$ where $S$ is the set of states, $A$ is the action space, $T(.)$ is the transition function, $R(.)$ the reward function, $O$ is the
observation space, and $Z(.)$ is the observation function. The casual relation between
the elements of the tuple are shown in the figure 2.6

Figure 2.6: Influence Diagram in a POMDP framework - Casual Relationships
between POMDP states, actions, rewards and observations

2.9.3.1 Process History

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, A^1), ...

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 ob-
servable history as it cannot fully observe the underlying world state. The SDS has the
prior belief about the world state which are summarized by the probability distribution
2. SPOKEN DIALOGUE SYSTEMS

$b_0$ 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.

2.9.3.2 Performance Measures

The system trajectories are ranked with the help of a Value function ($V: H \rightarrow \mathbb{R}$) which assigns a real number to each system history $h \in H$. A history $h$ will be preferred over $h'$ if $V(h) > V(h')$. In case of infinite horizon problems i.e. the problems where the decision stops after a finite number of steps the value function for a system trajectory $h$ of length $l$ is simply the sum of rewards attained at each stage [Bellman, 1954]

$$V(h) = \sum_{t=0}^{t=l} R(s^t, a^t)$$

(2.5)

In case of infinite horizon problems i.e. the problems where the system trajectory is unbounded, a discount factor $\gamma$ is introduced which states that the rewards received later get discounted which contribute less than current rewards. The value function for such total discounted reward function is given as [Bellman, 1954]

$$V(h) = \sum_{t=0}^{\infty} \gamma^t R(s^t, a^t)$$

(2.6)

2.9.3.3 Policy

On each turn in a spoken dialogue, the system has to decide and execute an optimal course of action in an uncertain environment contingent on the observable history. A policy $\pi: H_0 \rightarrow A$ is a rule that maps observable histories into actions. The main aim of the spoken dialogue system is to choose a policy which maximizes the objective function that is defined on the set of system trajectories($H_0$). Given a history

$$h' = \langle a^0, o^0 >, < a^1, o^1 >, ..., < a^{t-1}, o^t >$$

the action prescribed by the policy $\pi$ at time $t$ would be $a^t = \pi(h')$ where $a^0$ is the system’s initial action and $o^t$ is the latest observation.

The likelihood of particular system trajectory is controlled by inducing the probability distribution $Pr(h|\pi, b_0)$ over all possible sequence of states and actions by the
system for initial distribution \( b_0 \). The *Expected policy value* is the expected value of system trajectories induced by the policy \( \pi \) and is given by

\[
EV(\pi) = V^\pi = \sum_{h \in H} V(h)Pr(h|\pi, b_0)
\]  

(2.7)

The system’s goal is to find a policy \( \pi^* \in \Pi \) with the maximum expected value from the set \( \Pi \) of all possible policies. The policies are generally represented using tractable representations where in the observable histories are either represented as probability distributions over system states or grouped into a finite set of distinguishable classes using finite-suffix trees or Finite state controllers.

### 2.10 Conclusion

This chapter discusses the importance of spoken language in the design of the human computer interaction process. When a user interacts with a computer, the user unknowingly embeds lots of information at different levels of speech which if capitalized properly can help in the design of an efficient and effective dialogue system. The chapter also elaborated on different modules that are to be focussed on during the design of a spoken dialogue system. Testing of a spoken dialogue system with different dialogues is a challenging task, the system simulated user dialogue for testing purpose has shown remarkable results in the design of a spoken dialogue system. Dialogue manager forms the heart of the Spoken Dialogue system and the strategy it follows to reply to user query/utterance has to be based on some prior knowledge of the dialogue context. Various approaches in the design of the dialogue manager has been discussed in the last section.