Spoken Dialogue Applications: Research Directions and Commercial Deployment

2

This chapter consists of three main parts. In the first part different types of spoken dialogue application will be described. The second part of the chapter will provide an overview of past and current research directions in spoken dialogue technology. The final part will discuss the commercial potential of spoken dialogue technology.

Spoken Dialogue Applications

Spoken dialogue systems can be used for many different purposes. Many applications have involved the retrieval of information, in particular, travel information such as flight schedules and train timetables. Other applications have addressed the provision of services, such as call management, and transactions, such as making reservations or managing a bank account. A more advanced type of application is collaborative problem solving, for example, developing a plan to evacuate a disaster area or giving advice on a financial investment. Spoken dialogue systems can also be deployed in educational contexts as an instructional aid. Finally, there are applications within the area of games and entertainment.

Information Retrieval, Services and Transactions

The information age has brought the promise of vast amounts of information that are potentially accessible to anyone at anytime and anywhere. The Internet has made this information available to anyone with a PC and Internet access. However, many people do not have ready access to PCs and it has been estimated that, although there are a quarter of a billion PCs in the world, there are more than 1.3 billion telephones (Larson, 2002). Telephones have the potential to provide a more universal interface to information and, given recent developments in the integration of the telephone and the Internet, a speech interface to this information brings the promise of a natural, efficient and ubiquitous access method. For this reason it is not surprising that most spoken dialogue applications involve some sort of information retrieval. Typical examples include travel schedules, traffic reports, weather forecasts, stock prices, lists of movies at a local cinema, directory enquiries and package tracking.

Information retrieval is a relatively constrained type of application in which the system elicits one or more required parameters from the user and then executes a query to find a response – usually by retrieving information from a database. Such an application is analogous to a form-filling dialogue on a Web page to obtain information or order goods. Implementing a spoken information retrieval application can be achieved using fairly simple methods, in which the system prompts the user with a predefined sequence of queries and accepts narrowly constrained responses. In this way the vocabulary and language structures can be restricted to facilitate speech recognition and language understanding, while dialogue control can be implemented using simple and well-understood techniques.

The following example is taken from a dialogue with the Philips Automatic Train Timetable Information System (available at: +49 241 604020) which provides information over the telephone about train connections between 1200 German cities. The system conducts a natural dialogue in German with the user, who can speak to the system using fluent, unrestricted speech. The text of this dialogue is translated from the original German (Aust et al., 1995).

2.1
1 System: Good morning. This is the automatic timetable information service. How can I help you?
2 User: I need a connection from Hamburg to Munich.
3 System: When would you like to go from Hamburg to Munich?
4 User: Next Friday at eight o'clock.
5 System: So you want to travel on Sunday the 20th of November at eight o'clock a.m.?
6 User: No, on Friday.
7 System: So you want to travel on Friday the 18th of November at eight o'clock a.m.?
8 User: Yes.
9 System: There is the following connection: With IntercityExpress 793, departure from Hamburg at 8:02, arrival in Munich at 14:05. Do you want me to repeat the connection?

The dialogue begins with an open-ended prompt to which the user can respond with some or all of the information that the system requires to consult the database for a suitable train. If more values are required, the system requests these, as in utterance 3, and then confirms the values, as in utterance 5. Note that the user is able to correct the system's attempted confirmation (utterance 6). Details of how this and similar systems are implemented are described in Chapter 5.

Problem-Solving Applications

By their very nature problem-solving applications require more complex dialogue as they require the system and user to act together to solve a problem.

The following example illustrates collaborative problem solving in the Circuit-Fix-It Shop system (Smith and Hipp, 1994, p. 61). This system helps users to fix an electronic circuit. Communication between system and user is motivated by the fact that neither participant is fully equipped to solve the problem alone. The system has all the necessary knowledge about the task but does not have the required sensory abilities to monitor the state of the world nor to perform actions in the world. These are the concern of the human user who also has some knowledge of the task, which can range from that of a novice to that of an expert, depending on the individual user.

2.2
1 System: Is there a wire between connector 84 and connector 99?
2 User: It is not there.
3 System: Add a wire between connector 84 and connector 99.
4 User: I need help.
5 System: Obtain a wire.
6 User: Okay.
7 System: Connect metal of one free end of the wire to connector 84.
8 User: How?
9 System: Bend back connector 84.
10 User: Done.
11 System: Insert free end of wire into connector 84.
12 User: The wire is connecting.

At the beginning of the dialogue, the system does not know whether there is a wire between connector 84 and connector 99. As this is a missing element in the part of the problem to be solved, the system produces utterance 1 to ask the user. The user confirms that the wire is missing. From this the system can infer that the user knows the location of the connectors and these facts are added to the user model – a store of information that the system believes the user knows.

To complete the current subtask, the system instructs the user to add a wire between the connectors. As the user does not know how to do this, the system has to adopt a subgoal to advise the user on how to accomplish this task. This subgoal consists of the following actions:

1. locate connector 84;
2. locate connector 99;
3. obtain a wire;
4. connect one end of the wire to 84; and
5. connect the other end of the wire to 99.

As the user model contains the information that the user can locate connectors 84 and 99, instructions for the first two actions are not required and so the system proceeds with instructions for the third action, which is confirmed in utterance 6, and for the fourth action. Here the user requires further instructions, which are given in utterance 9, with the action confirmed by the user in utterance 10. At this point the user asserts that the wire between 84 and 99 is connecting, so that the fifth instruction to connect the second end to 99 is not required.

In the Circuit-Fix-It Shop system the dialogue evolves dynamically, depending on the current state of the problem being solved, as well as on the system's estimate of what the user needs to be told. As the state of the problem changes constantly, as well as the state of the user's knowledge, the system needs to maintain a record of its current information state about the problem and the user and to update this information dynamically.

Educational Applications

Spoken dialogue interfaces can be used in educational applications to provide a more natural mode of communication between students and computer-based learning materials. One particularly interesting example involves the use of the Center for Spoken Language Understanding (CSLU) toolkit to assist profoundly deaf children to speak. The CSLU toolkit, which was developed by the CSLU at the Oregon Graduate Institute, includes a graphical authoring environment to support the development of interactive speech applications (cslu.cse.ogi.edu/toolkit/). The latest release of the toolkit, version 2.0, also contains an animation engine CUAnimate, donated by the Center for Spoken Language Research (CSLR) of the University of Boulder, Colorado (cslr.colorado.edu/). Previous versions of the toolkit used an animated three-dimension talking head (Baldi), developed at the Perceptual Science Laboratory at the University of California, Santa Cruz (UCSC) (mambo.ucsc.edu/).

The CSLU toolkit's graphical authoring tool enables a wide range of learning and language training applications to be developed. Baldi has been used at the Tucker–Maxon Oral School in Portland, Oregon, to help deaf children to learn how to form their words and practise pronunciation (Cole et al., 1999; Connors et al., 1999). Baldi's lips, tongue and jaw movements are a near-perfect copy of human speech movements. The children mimic Baldi and then find out if their responses are correct through Baldi's feedback. In addition to deaf children, the toolkit is being used to develop applications for children with autism, who have problems with verbal communication. A variety of other instructional aids such as vocabulary tutors and interactive reading tutors have also been developed, and the latest release of the toolkit also includes the CSLU Vocabulary Editor and Tutor. Chapter 8 contains a series of tutorials for the development of educational applications using the CSLU toolkit.

Conversational interfaces are also being used in conjunction with Intelligent Tutoring Systems (ITSs). ITSs are similar to problem-solving applications as they involve a dialogue between the system and the learner, who is trying to solve a problem. However, in an ITS the purpose of the interaction is to enable the learner to learn about the problem, so that important components of the architecture will include a tutoring strategy that determines the system's behaviours and a learner model that represents the learner's current state of knowledge.

Some recent ITSs support mixed-initiative conversational dialogues with the learner, in which the learner types in answers in English and the system conducts a dialogue in which solutions are developed. Graesser et al. (2001) describe a number of such systems that they have been developing, including AutoTutor, a conversational agent with a talking head, that helps college students learn about computer literacy. The talking head uses synthesised speech, intonation, facial expressions, nods and gestures to communicate with the learner, who types in his or her contributions. The learner's answers can be lengthy, exhibiting deep reasoning and thus requiring sophisticated natural language processing in order to interpret them correctly. Usually a lengthy multiturn dialogue evolves during the course of answering a deep reasoning question. The dialogue properties of advanced systems such as this will be discussed in greater detail in Chapter 12.

Most dialogue-based ITSs involve text-based interactions. In a recent paper, Litman (2002) has proposed adding a spoken language interface to an existing text-based ITS. The initial stages will explore the issues involved in replacing the current input and output modalities with speech and, in particular, with investigating the additional problems that arise with speech recognition errors. At this level speech would function as a potentially more convenient mode of input and output compared with text-based interaction. However, in the longer term the pedagogical effectiveness of a speech interface will be explored, by making use of information that is only available in speech, such as prosodic features that can indicate emotional states such as annoyance, confusion, boredom and certainty. The plan is to use this additional information to enable the system to adapt its tutoring strategies to match the learner's perceived emotional and cognitive state.

Games and Entertainment

Spoken dialogue technology has tremendous potential in computer games and entertainment. The simplest applications involve the replacement of the mouse, keyboard and joystick by voice commands. "Hey You, Pikachu!" from Nintendo is a good example (www.pikachu.com). The game consists mainly of the player taking Pikachu to different places and getting him to carry out actions for which the commands that can be recognised are presented on screen. Another example is Game Commander from Sontage Interactive (www.gamecommander.com). Game Commander is a voice control application for games. Game Commander allows players to control many games with verbal commands instead of, or in conjunction with, keyboard, joystick and mouse controls. For example, instead of remembering that `Alt+Shift+F8` is the command for lock missiles, you can just say "Lock Missiles". Recently, Scansoft has released a Games Software Development Kit for PlayStation2 that enables integration of speech recognition functions into games and "edutainment" software (www.scansoft.com/games).

Spoken dialogue technology is being combined with computer games technologies in a European research project NICE (Natural Interactive Communication for Edutainment) (www.niceproject.com). NICE is developing a prototype system for children and adolescents that will allow them to have conversations with the fairy-tale author Hans Christian Andersen and to play games with animated characters. Communication will involve spoken conversation combined with two-dimensional input gestures in a three-dimensional dynamic graphics virtual world.

It has been estimated that there is a huge market for advanced edutainment systems that could act as companions to groups such as the elderly, as well as providing useful assistance such as providing help in medical emergencies. There are already some examples of such systems in the form of "chatterbots" – a type of conversing computer. The term "chatterbot" was coined by Michael Maudlin, founder of the Lycos search engine (Maudlin, 1994). A chatterbot is a computer program that accepts verbal input from the user and outputs a verbal response. Generally, the input and output take the form of typed natural language phrases or sentences, although some chatterbots are now also able to handle spoken input and output.

Chatterbots would appear to be most successful when they do not need to simulate an intelligent, cooperative conversational participant. Chatterbots in games do not need to make relevant responses – indeed, their odd behaviour can often be seen as part of the game. Nevertheless, the techniques used to produce chatterbots have also been used successfully in a number of more serious applications, for example, to provide on-line help. Ford Motor Company has an online chatterbot called Ernie who helps technicians at its network of dealerships to diagnose car problems and to order parts. Ernie is an example of a vRep, an automated agent developed by NativeMinds, that uses natural language dialogue to answer customers' questions (www.nativeminds.com). Similarly, IBM's Lotus software division employs a service chatterbot that can diagnose problems in a user's software and upload patches to the user's computer (Nickell, 2002). In these applications the success of the chatterbot depends on an extensive set of patterns that match the user's input within a restricted domain to trigger an appropriate system output. The technology underlying chatterbots and other systems that simulate conversation will be described in more detail below.

Research in Spoken Dialogue Technology

Research in spoken dialogue technology can be traced back to work on natural language processing and artificial intelligence (AI) in the 1960s. The earliest dialogue systems involved typed input of natural language phrases and sentences, and it was not until the late 1980s that the speech and natural language communities started to come together to develop spoken dialogue systems as they are known today.

Two main approaches can be distinguished in dialogue research. One approach has focussed on theoretically motivated models of dialogue based on research in natural language processing and artificial intelligence. The other approach, sometimes known as "simulated conversation" or "human–computer conversation", has used methods ranging from pattern matching to fairly complex data-driven techniques to simulate conversational interaction. The following sections present a brief historical overview of dialogue systems from the 1960s through to the present time.

Natural Language Dialogue Systems in the 1960s

A number of systems developed in the 1960s aimed to provide natural language interfaces to computers. Much of this work disregarded theories of human language processing from theoretical linguistics and cognitive psychology and used a variety of techniques to manipulate language to produce acceptable results.

BASEBALL was an early question–answering system that could answer questions about the dates, locations, teams and scores of baseball games (Green et al., 1963). The system could be queried with questions such as "Who beat the Yankees on July 4?" or "Did any team play at least once in each park in each month?" To illustrate the ad hoc nature of some of the language processing, the meaning of the word "who" in the system's dictionary was given as "Team = ?", a domain specific meaning that could not be generalised to other domains.

STUDENT was a program that was able to solve algebra story problems cast in natural language (Bobrow, 1968). For example,

2.3
Mary is twice as old as Ann was when Mary was as old as Ann is now. If Mary is 24 years old, how old is Ann?

The program converted the natural language input into a set of algebraic equations by breaking it into simple patterns and looking for words and phrases that could be replaced by arithmetic expressions and variables. STUDENT demonstrated some interesting features that foreshadowed later developments. One of these was that the system was able to deal with connected discourse and not simply with isolated sentences. A second feature was that the system used a store of general knowledge to assist with the solution of problems and it could expand its store of knowledge by eliciting further information from the user. Nevertheless, systems such as STUDENT and BASEBALL were generally based on techniques that could be applied successfully within the domain in question but were not easily generalisable to other domains. Moreover, the techniques in question were not based on any more theoretically motivated approaches to dialogue.

Linguistic and Artificial Intelligence Approaches to Dialogue in the 1970s and Early 1980s

The 1970s and early 1980s saw the emergence of more theoretically motivated approaches to dialogue, based on research in natural language processing and artificial intelligence.

SHRDLU

SHRDLU, developed at MIT in the early 1970s, was an attempt to model the processes of natural language understanding on a computer (Winograd, 1972). The system operated within a very limited domain, a tiny world containing a box, a table, blocks and pyramids that could be manipulated according to commands from the user. The system could also answer questions and could "learn" from the interactions. The following is an example of an interaction with SHRDLU:

2.4
1 User: Pick up a red block.
2 SHRDLU: OK.
3 User: Grasp the pyramid.
4 SHRDLU: I don't understand which pyramid you mean.
5 User: Find a block which is taller than the one you are holding and put it into the box.
6 SHRDLU: By "it" I assume you mean the block which is taller than the one I am holding.
7 User: OK. What does the box contain?
8 SHRDLU: The blue pyramid and the blue box.

SHRDLU has to process the user's input, which in some cases is fairly complex. However, the input has to be processed not only linguistically but also in relation to the simulated world in which the positions of objects change continually as a result of the user's commands. Objects can be referred to using pronouns and other referring expressions. In some cases the reference is ambiguous. For example, "grasp the pyramid" is ambiguous, as there are three pyramids in the scene, and "put it into the box" is ambiguous, as "it" could refer either to the block that SHRDLU was holding or to the larger block that SHRDLU was to find.

SHRDLU used a combination of syntactic, semantic and pragmatic analyses to interact within the blocks' world. The syntactic analysis used a comprehensive grammar of English that assigned a syntactic structure to the user's input by determining the parts of speech of each word and the permissible combinations of words as phrases in a sentence. The semantic analysis enabled SHRDLU to reject meaningless sentences based on semantic knowledge about the objects in the domain. For example, the question "Can the table pick up blocks?" was rejected because a table is an inanimate object and the verb "pick up" requires a subject that is animate. The pragmatic component kept track of the objects in the domain, for example, "Block1 supports Block2" and had procedures to represent actions that could be carried out. If there was an instruction to grasp an object, it would be necessary to check if the object was of a type that could be manipulated, if there was another object on top of the object to be grasped, if the robot was currently holding some other object, and so on. These procedures enabled SHRDLU to carry out actions involving several subactions and, more interestingly, to answer questions about its actions. For example, if asked "Why did you put object2 on the table?" SHRDLU could answer "To get rid of object2". If asked "Why did you get rid of object2", SHRDLU would reply "To grasp object1".

SHRDLU was able to combine its processing modules in an interesting way to resolve sentences that might otherwise be ambiguous. The following example illustrates:

2.5 Put the blue pyramid on the block on the box.

Using syntactic analysis alone, these words could be grouped in two different ways:

1 Put (the blue pyramid on the block) in the box.
2 Put the blue pyramid on (the block in the box).

In other words, either there is a blue pyramid on a block or there is a block in the box. SHRDLU would begin to analyse the sentence using its syntactic knowledge. To decide on the meaning of the sentence it would consult its semantic knowledge, for example, whether the sentence is meaningful in terms of objects that can be manipulated. At this stage there would still be two interpretations. However, the pragmatic component would then check the current state of the world to see if one interpretation made more sense in context. If there was a blue pyramid on a block, then the first interpretation would be accepted, otherwise the second interpretation would be investigated. This interaction between different sources of knowledge to interpret natural language sentences in context remains an important area for research in natural language processing.

Artificial Intelligence Approaches: Knowledge Structures and Inference

In addition to knowledge about objects and their attributes, as utilised in SHRDLU, natural language understanding systems require other knowledge structures in order to make sense of natural language text, such as knowledge about event sequences and knowledge about people, their beliefs, desires, motivations and plans. Schank (1975) developed a theory of language in the 1970s called Conceptual Dependency Theory, in which the emphasis was on the content of information rather than on its syntactic form. As the focus moved from the analysis of single sentences to larger structures such as stories, Schank and his colleagues at Yale developed knowledge structures to represent events, goals and plans that would support the interpretation of stories and similar discourse units.

Scripts were used to represent stereotypical sequences of events, such as going to a restaurant or travelling by bus. Schank argued that to understand a story, people (and computers) required knowledge beyond the information contained explicitly in the text. The following example, taken from an interaction with the program SAM (Script Applier Mechanism), illustrates a script for VIP visits (Cullingford, 1981):

2.6
Sunday morning Enver Hoxha, the Premier of Albania, and Mrs Hoxha arrived in Peking at the invitation of Communist China. The Albanian party was welcomed at Peking Airport by Foreign Minister Huang. Chairman Hua and Mr Hoxha discussed economic relations between China and Albania for three hours.

There are several points in this apparently simple story where script knowledge is required to make sense of the story. Words like "invitation" cause SAM to look in its database of scripts and, when it finds VIPVISIT, a number of relevant concepts are activated, such as arrival and mode of travel. Using this information SAM examines the second sentence and can conclude that, as the group has been welcomed at Peking Airport, they are likely to have arrived there and to have travelled by plane. The third sentence makes sense in the context of a VIP visit, as one of the expected events is an "official talks" episode, in this case a discussion about economic relations. Various inferences are made during the processing of the story. There is no mention of where the talks are held, so SAM assumes it was in the city where the Hoxha party arrived. Similarly, SAM can answer questions such as "Who went to China?" although the story does not say explicitly that anyone went to China, only that the Hoxhas arrived in Peking.

Research in scripts showed that understanding connected discourse involves more than analysing the syntactic structure of sentences and examining their literal meanings. Understanding involves finding causal links between events and making assumptions about events and other items that have not been explicitly mentioned. SAM used the notion of scripts, or stereotypical sequences of events, to perform this reasoning. Another program from the same group, PAM (Plan Applier Mechanism), used the notion of plans to make sense of events that, unlike scripts, had not previously been encountered (Wilensky, 1981). PAM encoded general information about how people achieve goals and about what sorts of goals they try to achieve. Another program QUALM was used in conjunction with SAM and PAM to answer questions (Lehnert, 1980). Finally, within this tradition of research, a program called POLITICS modelled political beliefs and the way in which different people can have different interpretations of the same event, illustrated with a conservative and a liberal view of particular events (Carbonell, 1981).

Natural Language Database Queries

Systems that could process natural language queries and translate them into a formal database query language were one of the earliest major successes in natural language processing. LUNAR, which could answer natural language queries to a database containing information about moon rocks, was one of the first such interfaces that appeared in the late 1960s (Woods et al., 1972). Following active research throughout the 1970s and 1980s, a number of systems have become available commercially, such as English Wizard and Access ELF (Androutsopoulos and Ritchie, 2000).

Strictly speaking a natural language database system does not engage in a dialogue with the user. Rather the user submits a query, usually in typed natural language, that is translated by the system into a query in a formal database language such as SQL. Most systems will simply generate an error message if they are unable to process the user's input and invite the user to submit another query. Some systems are able to process a series of queries that are related to one another and that use discourse phenomena such as anaphora and ellipsis, as in the following examples, quoted in (Androutsopoulos and Ritchie, 2000):

2.7
1 User: Who leads TPI?
2 System: E. Feron
3 User: Who reports to him? (example of anaphoric reference: "him" refers back to "E. Feron" in the previous utterance).

2.8
1 User: Does the highest paid female manager have any degrees from Harvard?
2 System: Yes, 1.
3 User: How about MIT? (elliptical question that is understood by replacing "Harvard" in the first question with "MIT").

Systems that employ discourse devices such as these support a user-driven dialogue in which the user can ask a series of questions and the system has to keep track of the people and objects mentioned in the dialogue in order to be able to process subsequent queries. Considerable research has been directed towards the issue of cooperative systems that provide some form of feedback if the user's query cannot be sufficiently processed. For example, problems may arise if the vocabulary of the dialogue does not map directly on to the vocabulary of the application, or if the query makes false assumptions concerning the actual contents of the database so that no straightforward response is possible. Kaplan (1983) addressed the issue of false assumptions, as illustrated in the following example:

2.9
1 User: How many students got As in Linguistics in 1981?
2 System: None.

The system's response is correct if the set of students that got "A" in linguistics is empty, but it would also be correct if there were no students taking linguistics in 1981. However, in the latter case, the system's response is misleading, as it does not correct the user's false assumptions.

Problems may also arise if the user has misconceptions about the world model represented in the database. Carberry (1986) discusses the query "Which apartments are for sale?" which (in an American real-estate context) is inappropriate, as apartments are rented, not sold, although apartment blocks may be sold, for example, to property developers. Resolving this problem involved discerning the user's goal that gave rise to the ill-formed query.

Dialogue systems that enable users to access the contents of a database over the telephone are usually designed in such a way as to assist the user in the construction of the query by eliciting the required parameters. In most commercially deployed systems each parameter is elicited separately and inserted subsequently into a query template. In this way, the issue of ill-formed queries is avoided.

Dialogue as Planning

Another approach involved the application of formalisms and inference mechanisms from research in planning (a subdiscipline of artificial intelligence) to the modelling of the plans and intentions of participants in dialogue. Consider the following simple example, taken from a system that plays the role of a railway employee answering the queries of a client who has one of two possible goals – to board a train or to meet a train (Allen, 1983; Allen and Perrault, 1980):

2.10
1 User: When does the train to Windsor leave?
2 System: 3.15 at Gate 7.

In this example the user asks for the departure time of the train. However, the system reasons that in order to board a train a person will also need to know the departure location, hence the mention of Gate 7. To provide helpful answers in this way the system has to interpret the user's question and then progress through a series of inferences to determine the user's goal and to identify any potential obstacles to the achievement of this goal. Various inference rules are involved, such as the Action–Effect rule, which can be stated informally as follows:

2.11
If an agent wants to achieve a goal, and some action has that goal as its effect, then the agent will probably want to carry out that action.

A chain of inferences leads from the literal interpretation of the user's question to the system's production of a helpful response. This work has continued throughout the 1980s and 1990s and has formed the basis for current research on conversational agency known as the Belief, Desire and Intention (BDI) approach (see Chapter 12).

Using Frames to Guide the Dialogue – The Travel Agent GUS

The Genial Understander System (GUS) combined sophisticated natural language understanding in terms of morphological and syntactic analysis with a frame-based approach to dialogue control. GUS acted as a travel agent that conducted a dialogue in written natural language with a client who wanted to make a simple return trip from Palo Alto to a single city in California. The following is an excerpt from a dialogue with GUS (Bobrow et al., 1977):

2.12
1 GUS: Where do you want to go to?
2 User: I want to go to San Diego on May 28.
3 GUS: What time do you want to leave?
4 User: I must be in San Diego before 10 a.m.
5 GUS: Would you like PSA flight 102 that arrives at 9.15?
6 User: What is the next flight?
7 GUS: Air California flight 310 that leaves at 8.30 a.m. and arrives at 10 a.m.
8 User: I'll take the first one.

GUS used a set of frames to guide the dialogue. For example, the following is part of a simplified version of a frame, consisting of a series of slots and fillers, for one leg of a trip:

Slots	Fillers
Fromplace	City
Toplace	City
Traveldate	Date
Departure	Time range
Arrival	Time range

During the course of the dialogue, slots are filled with values of the appropriate type. Each slot has procedures to generate questions to acquire information from the client. For example, the "Toplace" slot has the question "Where do you want to go to?" If the user supplies more information than is asked for by a question, as in 2 ("I want to go to San Diego on May 28"), then other slots such as the Traveldate will also be filled and it will not be necessary to ask a Traveldate question later in the dialogue. This is similar to the mechanisms used in many current spoken dialogue systems, as described in Chapters 5 and 10.

GUS also uses sophisticated techniques for resolving expressions that are incomplete and are thus potentially difficult to interpret. For example, the phase "the next one" in 6 is interpreted as referring to the flight following the one mentioned in 5. Another example (not cited in this extract) was a relative date expression "Friday in the evening" in response to the question "What date do you want to return?" This expression has to be interpreted as the Friday following the date of arrival and not the Friday following the date of the conversation. Many current systems involving date and time expressions use similar procedures for resolving relative expressions such as these.

Simulated Conversation

Simulated conversation is concerned with the development of systems that simulate conversational interaction. This approach uses methods such as pattern matching rather than models and theories derived from natural language processing and artificial intelligence. Nevertheless many of the systems developed within this approach are able to perform impressively as conversationalists.

Simulated conversational systems can be traced back to the "imitation game" proposed in 1950 by the British mathematician and code-breaker Alan Turing in his seminal paper, "Computing Machinery and Intelligence", in which he considered the question "Can machines think?" (Turing, 1950). In the imitation game a person interacts in natural language via a terminal and has to decide whether they are interacting with a person or a computer. If they are unable to tell, then the computer should be regarded as having intelligence. Although the validity of Turing's test of machine intelligence has come in for considerable criticism on methodological grounds, there has been a long tradition of efforts to develop computer programs that would pass the test. In 1991, Hugh Loebner initiated the annual Loebner Prize competition, offering a prize of $100,000 for the first program to pass an unrestricted Turing test. The prize was won in 2000 and 2001 by a program called A.L.I.C.E. written by Richard Wallace (www.alicebot.org).

ELIZA

The first, and probably the best known, system to simulate conversation was Weizenbaum's ELIZA program, published in 1966. ELIZA simulates a psychotherapist. The following is an example of a typical conversation with ELIZA (Weizenbaum, 1966):

2.13
1 Patient: Men are all alike.
2 ELIZA: In what way?
3 Patient: They're always bugging us about something or other.
4 ELIZA: Can you think of a specific example?
5 Patient: Well, my boyfriend made me come here.
6 ELIZA: Your boyfriend made you come here?
7 Patient: He says I'm depressed much of the time.
8 ELIZA: I am sorry to hear you are depressed.
9 Patient: It's true. I am unhappy.
10 ELIZA: Do you think coming here will help you not to be unhappy?
11 Patient: I need some help, that much seems certain.
12 ELIZA: What would it mean to you if you got some help?
13 Patient: Perhaps I could learn to get along with my mother.
14 ELIZA: Tell me more about your family.
15 Patient: My mother takes care of me.
16 ELIZA: Who else in your family takes care of you?
17 Patient: My father.
18 ELIZA: Your father.
19 Patient: You are like my father in some ways.
20 ELIZA: What resemblance do you see?
21 Patient: You are not very aggressive but I think you don't want me to notice that.
22 ELIZA: What makes you think I am not very aggressive?
23 Patient: You don't argue with me.
24 ELIZA: Why do you think I don't argue with you?
25 Patient: You are afraid of me.
26 ELIZA: Does it please you to believe I am afraid of you?
27 Patient: My father is afraid of everybody.
28 ELIZA: What else comes to your mind when you think of your father?
29 Patient: Bullies.
30 ELIZA: Does that have anything to do with the fact that your boyfriend made you come here?

While this conversation appears impressive and human-like, in reality ELIZA's conversational abilities were limited and were based on the use of a number of keywords and associated responses. For example, a sentence containing "always" could produce the response "Can you think of a specific example?", while a family word such as "mother" would trigger the response "tell me more about your family". A pronoun transformation routine, used in many similar programs, would transform an input pattern such as “You are not very aggressive” into “What makes you think I am not very aggressive?” The patient’s input sentences were stored and cycled back at a later point when no specific keyword could be found to trigger a response, giving the illusion of an intelligent conversational partner. Thus ELIZA’s final question in this extract, “Does that have anything to do with the fact that your boyfriend made you come here?”, goes back to something said earlier by the patient at a time when the patient’s input “bullies” does not match any keyword pattern.

PARRY

PARRY, a system developed by Colby (1975), was a contemporary of ELIZA. PARRY simulated a patient with paranoia and was demonstrated in a version of the Turing test in which psychiatrists were unable to distinguish utterances by PARRY from utterances spoken by real paranoids. PARRY had a large set of about 6000 patterns against which any input was matched. While ELIZA did not take the initiative in a conversation but reflected back what the user said, in the manner of a Rogerian psychotherapist, PARRY was able to keep a conversation going because it always had something to say, based on a number of stored resentments that modelled paranoid behaviour. Indeed, when PARRY did not understand the user’s input, it would redirect it according to its own interpretation. In this way PARRY simulated the sorts of unpredictable associations that a paranoid person might make. The following is an example of part of a conversation with PARRY:

2.14
1 Interviewer: Why are you in the hospital?
2 PARRY: I shouldn’t be here.
3 Interviewer: Who brought you here?
4 PARRY: The police.
5 Interviewer: What trouble did you have with the police?
6 PARRY: Cops don’t do their jobs.
7 Interviewer: What do they do instead?
8 PARRY: Cops arrest the wrong people.
9 Interviewer: They arrest the right ones sometimes.
10 PARRY: The law should get those Italian crooks.
11 Interviewer: Are you Italian?
12 PARRY: My background is British, but I was born in this country.
13 Interviewer: Do you know any Italian crooks?
14 PARRY: I try to avoid the underworld.
15 Interviewer: Are you afraid of the underworld?
16 PARRY: The mafia are the worst gangsters.
17 Interviewer: Are you connected with the mafia?
18 PARRY: You know, they know me.
19 Interviewer: How do they know you?
20 PARRY: They spy on me.
21 Interviewer: Why do they?
22 PARRY: They want to get false evidence to frame me.
23 Interviewer: What do they have against you?
24 PARRY: They are out to get me.

25 Interviewer: What for?
26 PARRY: I have incriminating evidence against the mafia.
27 Interviewer: Are you sure of all this?
28 PARRY: You don't believe me?

CONVERSE

CONVERSE (Batacharia et al., 1997) was the 1997 winner of the Loebner prize. The character of CONVERSE is Catherine, a 26-year-old female whose character details – her physical appearance, likes and dislikes, where she works, and so on – are stored in a database. CONVERSE is driven by scripts covering a range of 80 topics, including crime, racism, religion, mobile phones, "The Simpsons" and abortion. The system tries to maintain control of the conversation by asking a question at the end of a system utterance, but is also responsive to a question from the user. A weighting system determines whether the system-driven or system-responsive mode gains control. The following is an extract from the system's output from the Loebner competition, in which the system displayed strong views on the lesbian couple Bill Clinton had welcomed to the White House the night before the competition, and also on Clinton himself:

2.15
1 PROGRAM: Did you see that story on CNN last night about the lesbian couple who came out at a White House party on Sunday?
2 JUDGE: Yes, I did. I think it may all be a publicity stunt for Ellen.
3 PROGRAM: Ellen Degeneres was one of them – she was kissing her lover in front of Bill Clinton. Do you think he was very embarrassed?
4 JUDGE: I think he wants the gay vote. He will do anything to get it and keep it. He does not have a conservative base.
5 PROGRAM: Are you still a friend of Bill Clinton?
6 JUDGE: I am not sure if he has any real friends. He has achieved a lot and has left a lot of people behind.
7 PROGRAM: You really don't like the guy – well, at least that's definite – I thought most people were just indifferent or numb by now.

Comparing Spoken Dialogue Technology and Simulated Conversational Systems

Spoken dialogue systems differ from simulated conversational systems in the following ways:

1. They make use of more theoretically motivated techniques derived from speech technology, natural language processing, and artificial intelligence, as opposed to the pattern-matching techniques used in simulated conversational systems.

2. They are generally domain specific, focussing on transactional dialogues within a specific domain, such as flight enquiries, rather than modelling conversational interactions in general.

3. Simulated conversational systems use techniques and resources derived from empirical natural language processing, such as large databases of dialogue

patterns and thesaurus networks, as well as weighting systems that can bias the control of the dialogue and provide a range of flexible and unpredictable behaviours.

However, these distinctions are becoming increasingly blurred in large simulated conversational systems, such as CONVERSE, which already include knowledge sources such as scripts that represent conversational topics and a database that contains details on the system's personal characteristics. Proposed extensions to CONVERSE include the incorporation of a model of individual agent beliefs and intentions using techniques from artificial intelligence. Another feature to be developed is the use of statistical dialogue modelling and the machine learning of dialogue behaviours. These are features that are also being adopted in more advanced spoken dialogue systems (see Chapter 13).

Speech Technology

The natural language systems that have been described in the preceding sections have all involved typed input and output. This was due partly to the fact that speech technology had not developed sufficiently in the 1960s and 1970s to handle the complex natural language used in these systems. Another reason was that, while there was some interaction between the natural language and the artificial intelligence communities, there was almost no interaction between these communities and the speech technology community until the mid-1980s.

Research in speech recognition in the 1960s focussed on systems that were characterised by the following features:

1. Speaker-dependent recognition – the system had to be trained to recognise the speech of an individual user.
2. Discrete word recognition – the speaker had to pause between each word to enable the system to identify word boundaries.
3. Small vocabularies of less than 50 words.

A major research programme (Speech Understanding Research (SUR)), sponsored by the Advanced Research Projects Agency (ARPA) of the United States Department of Defense, ran from 1971 to 1976 with the aim of overcoming the limitations of the systems of the 1960s. The systems were required to recognise connected speech from several cooperative speakers using a vocabulary of 1000 or more words. One system, HARPY, from Carnegie Mellon University (CMU) met the programme's requirements, being able to recognise more than 1000 words with an error rate of 5%. More important, the HARPY system was one of the first to use the statistically based form of modelling that is used in almost all current commercial and research speech recognition systems.

Subsequent work in speech recognition has focussed on the development of robust statistical models and of systems capable of handling large vocabulary continuous speech, leading to current voice dictation products. Handling difficult speech data, such as speech over the telephone, speech in noisy environments, and the speech typical of naturally occurring conversation, has directed the interest of speech technologists towards spoken dialogue as a prime example of difficult data. As a result, recent research in spoken dialogue technology has brought together the earlier traditions in speech technology, natural language processing and artificial intelligence that developed largely independently of one another throughout the previous decades.

Recent Developments in Spoken Dialogue Technology

Research in spoken dialogue technology emerged around the late-1980s as a result of two major government funded projects: the DARPA Spoken Language Systems programme in the United States and the Esprit SUNDIAL programme in Europe. The DARPA programme was concerned with the domain of Air Travel Information Services (ATIS). A number of research laboratories throughout the United States were involved, with the main focus on the input technologies of speech recognition and spoken language understanding that were required to make a flight reservation using spoken communication with a computer over the telephone (DARPA, 1992; ARPA, 1994). There was no explicit focus on dialogue issues in the ATIS projects. As all of the project participants were required to use the same database, it was possible to compare the performance of different implementations, and regular evaluations were a major focus of the ATIS programme. The ATIS corpora, a collection of task-oriented dialogues in the ATIS domain which is available from the Linguistic Data Consortium (LDC), provide a resource for developers and evaluators of spoken dialogue systems (www.ldc.upenn.edu).

The Esprit SUNDIAL project, funded by the European Community, was concerned with flight and train schedules in English, French, German and Italian (Peckham, 1993). The goal of the project was to build real-time integrated dialogue systems capable of maintaining cooperative dialogues with users. In addition to research on continuous speech recognition and understanding, a major technological focus was spoken dialogue modelling, resulting in significant insights into dialogue management. The SUNDIAL research led to a number of subsequent European-funded projects in spoken dialogue modelling, such as RAILTEL (Lamel et al., 1995), VerbMobil (Wahlster, 1993), ARISE (den Os et al., 1999) and DISC (Bernsen and Dybkjær, 1997). One well-known commercial development arising out of the SUNDIAL research is the Philips Automatic Train Timetable Information System (Aust et al., 1995).

The DARPA Communicator programme is the most recent large-scale government-funded effort in spoken dialogue technology, involving a number of research laboratories and companies across the United States, and including several affiliated partner sites in Europe (http://fofoca.mitre.org). The aim of the programme is to develop the next generation of intelligent conversational interfaces to distributed information, using speech-only as well as multimodal modalities. The Communicator dialogue systems support complex conversational interaction, in which both user and system can initiate the interaction, change topic, and interrupt the other participant. The application domains include meeting coordination and travel planning, requiring access to multiple data sources. In these respects the Communicator projects represent an advance on earlier programmes such as ATIS and SUNDIAL, which focussed on single domain enquiries and permitted less flexible dialogue strategies (see also Chapter 12).

Alongside these major research programmes there are many individual projects involving spoken dialogue technology. In the United States these include: the Spoken Language Systems Group at MIT, the CSLU at Oregon Graduate Institute of Science and Technology, the Sphinx Group at CMU, the CSLR at the University of Colorado at Boulder, and the Conversational Interaction and Spoken Dialogue Research Group at the University of Rochester. Companies involved actively in spoken dialogue research in the United States include AT&T, Bell Laboratories, Microsoft, IBM and SRI. Within Europe there is a large number of research centres, including the Natural Interactive Systems Laboratory in Odense, Denmark, the LIMSI Spoken Language Processing Group at the Laboratory of Computer Science for Mechanical and Engineering Sciences, Paris, the Centre for Speech Technology at the University of Edinburgh, the Speech Communication and Technology group at KTH, Stockholm, the Language Technology group at DFKI, Germany, CSELT in Italy, and the Department of Language and Speech at the University of Nijmegen, the Netherlands. There are also major research programmes in other parts of the world, particularly in Japan. A more extensive list of projects and links is provided in Appendix 5.

The Commercial Deployment of Spoken Dialogue Technology

Speech is a rapidly emerging technology that provides an alternative and complementary interface to the widely accepted graphical user interface. Many large companies, such as IBM, Philips, Microsoft, AT&T, Intel, Apple, Motorola and Unisys, have active research and development programmes in speech technology. IBM has recently initiated an 8 year project entitled the "Super Human Speech Recognition Initiative" involving about 100 speech researchers in the development of new technology to support "conversational computing". Similarly, the Speech Technology Group at Microsoft is involved in a number of projects aimed at their vision of a fully speech-enabled computer. A number of companies, such as Nuance Communications and Scansoft, specialise in speech technology while some, such as VoiceGenie, BeVocal, Tellme, Voxeo, Hey Anita and Voxpilot, focus exclusively on VoiceXML applications. The Web pages of these companies provide a wide range of information about the nature of speech technology products, applications and commercial benefits.

The Market for Speech Applications

A number of market research firms have predicted a rapid growth in the speech technology market. In a recent report the Kelsey Group, a leading authority on the potential of speech technologies, estimated world-wide revenues from speech technologies and the accompanying infrastructure hardware and software to grow from $505 million in 2001 to more than $2 billion in 2006 (Kelsey Group, 2002). This growth in the core technologies is predicted to trigger a multiplier effect that will drive speech and enhanced telephony services revenues to $27 billion by 2006. Similarly Allied Business Intelligence has predicted that the number of fixed voice portal users in North America will grow from 4 million in 2001 to 17 million by 2005, and mobile voice portal users will grow in the same period from 1 million to over 56 million (www.abiresearch.com). Detailed market analysis of the voice portal and speech technology sectors is available from the market research firm DataMonitor (www.datamonitor.com) as well as from TMA Associates (www.tmaa.com).

The Voice Web: An Infrastructure for Interactive Speech Applications

The Voice Web has come about as a result of a convergence of the computing and communications industries that will allow people to access information and services on the Internet with pervasive access devices such as the telephone and Personal Digital Assistants (PDAs). Interactive speech technologies provide the key to the Voice Web as they allow users to interact with the Internet using natural spoken language. The Voice Browser subsection of the World Wide Web Consortium (W3C) is focussed on expanding access to the Web in this way (www.w3.org/Voice/).

One critical factor in the development of the Voice Web is the emergence of an infrastructure for voice-based interfaces. Until recently the development of interactive speech applications with computer–telephone integration required special Application Programming Interfaces (APIs) and proprietary hardware and software. New languages such as VoiceXML (Voice Extensible Markup Language) and SALT (Speech Application Language Tags) allow developers to build on the existing Web infrastructure using standard Internet protocols.

VoiceXML is promoted by the VoiceXML forum (www.voicexmlforum.org), which was founded by AT&T, IBM, Lucent and Motorola. Its aim is to promote VoiceXML, a new language that has been developed to make Internet content and services accessible over the phone using natural speech. By March 2001 the Forum had grown to over 420 members. A series of tutorials on VoiceXML is presented in Chapters 9 and 10.

SALT is promoted by the SALT Forum, which was founded in 2001 by Cisco, Comverse, Intel, Microsoft, Philips and SpeechWorks (www.saltforum.org). The aim of the SALT Forum is to develop and promote speech technologies for multimodal and telephony applications. A series of tutorials on SALT is presented in Chapter 11.

Benefits of Speech Technology

While recent developments in speech technology are interesting from a technological viewpoint, for speech technology to be successful commercially it should have clear benefits within the commercial domain. Two potential beneficiaries of speech technology can be identified:

1. Technology and services providers.
2. End users.

Benefits for Technology and Service Providers

Technology and service providers include companies who develop speech products and applications and those that make use of these products in the delivery of their services, for example, call centres and internet service providers. The main benefit for providers is that speech will enable them to provide a wider range of services at reduced costs. Speech may also enable providers to promote services that will differentiate them from their competitors and that will provide enhanced customer satisfaction.

A number of studies have quantified the return on investment (ROI) for companies adopting speech technology. In a recent report by Nuance Communications on the business case for speech in the call centre, it was estimated that speech could create savings of more than 90% of the cost of a call by off-loading calls from call centre agents (www.nuance.com/learn/buscasespeech.html). The cost of a call handled by an agent was estimated at $1.28, while the cost of a call handled by a speech-enabled automated system was $0.10. These estimates were based on comparisons between the annual costs of an agent – salary, benefits, equipment, recruitment, training, calls handled per hour – and the costs of an automated system, including hardware, software, application development, installation and maintenance. It was shown that the time required to recoup the cost of a complete speech system could be as little as 3 months and that a large-scale system, handling over 100,000 calls per day, could provide savings of around $2 million over the course of a year. Other estimates make similar predictions (see, e.g., Winther, 2001).

Benefits for End Users

End users of speech technology are people who make use of speech-enabled services to perform tasks such as retrieving information, conducting transactions or controlling devices. For these end users the main benefit of speech is convenience, as they are able to access information and services at any time, from any place and using speech, which is a natural mode of communication. This convenience has been referred to as "pervasive computing", described by

IBM as ". . . personalised computing power" freed from the desktop, enabling information access anywhere, anytime, on demand (www-3.ibm.com/pvc/index.shtml). With the growth of smaller communications devices, such as Internet-enabled mobile phones and PDAs, as well as embedded devices without keyboards, speech provides an interface as an alternative to more cumbersome methods using pens and text entry. For people with physical disabilities speech may be the only useful interface.

Speech is also convenient in other ways. With current IVR (Interactive Voice Response) systems that use touch-tones and menus to obtain services and information, users often have to navigate a series of menus in order to conduct a transaction. For example, to transfer funds between accounts using a traditional phone-based banking system, a customer may have to press keys to select the transfer option, to indicate the source and destination accounts, and to input the required amount – three key presses for the options and several more for the amount. With a speech-based interface an experienced caller can say something like "Transfer three hundred pounds from my current account to my savings account", reducing the transaction time and the cost of the call considerably. Finally, because human agents in call centres can be released from routine information-gathering tasks that can be taken over by automated systems, calls involving simple enquiries can be answered more quickly and the caller is less likely to be put on hold.

Nuance Communications conducted a quantitative survey of customer satisfaction, attitudes and usage of speech based self-service applications (www.nuance.com/assets/pdf/speech_study.pdf). It was found that overall customer satisfaction was high (87%) and that the rate was even higher with wireless users (96%). The main reasons cited for preferring speech over alternative interfaces were speed, efficiency and ease of use. Similar results were reported in a market research survey by Northwest Airlines, who deployed a reservations service system based on Nuance technology and reported that over 66% of respondents rated the speech-based system as better than the Web-based alternative.

Challenges for Speech Technology

There are some contexts in which speech technology is not appropriate. Traditional web interfaces based on a graphical user interface can display information in graphical and tabular form. This form of presentation cannot easily be translated into speech. Long lists, which can be easily scanned on a visual interface, are difficult to process in an auditory mode. Listening to a long list takes much more time than skimming it visually. As speech is transient, long periods of listening tax human short-term memory. Furthermore, speech is not appropriate in environments requiring privacy nor in noisy environments that cause problems for speech recognition.

Even in contexts where speech is an appropriate medium, there are a number of technological challenges, including imperfections in speech technology and unrealistic user expectations. One of the main tasks for providers is to convince potential users and deployers that the technology will work properly in all situations and for all users.

Speech technologies are imperfect in a number of ways. The speech recognition component may misrecognise words, and attempts to correct errors can lead to error amplification. Major advances in speech recognition algorithms along with careful design can reduce error rates and minimise their consequences, but misrecognition errors will always be a challenge for designers of spoken dialogue systems. On the output side there may be problems with speech synthesis errors, when the system pronounces an unfamiliar name incorrectly or mispronounces words that are homophones, that is, words with the same spelling but different pronunciations, such as "tear", which can be pronounced to rhyme with "bear" or with "beer".

While the main focus to date has been on errors of speech recognition, there may also be errors involving other components of the system. The language understanding component may produce an incorrect parse of the user's input, and errors can also be produced by the dialogue manager, for example, in misinterpreting the user's intentions.

Unrealistic user expectations are also a major challenge for speech technology. Users may expect a speech system to perform to the level of systems depicted in science fiction, such as the computer in the television series Star Trek. These expectations may lead users to speak in complex sentences or to ask for services and information that are outside the domain of the system. Problems may also occur if speakers have strong regional or nonnative accents, have speech impediments, or use speech that is too casual or disfluent. Current systems work best with users who behave cooperatively and who adjust their speech to match the capabilities of the system. It is a major challenge for designers to produce systems that enable users to interact appropriately and efficiently with the system in a natural way, without lengthy instructions and training.

Summary

This chapter has examined the sorts of applications that are amenable to spoken dialogue technology. The majority of current systems involve the retrieval of information and the automation of routine transactions. More complex applications, such as problem solving, are still being developed in the research laboratories. Spoken dialogue is also being used in educational contexts and in games and entertainment. An interesting development is the conversational companion whose function is mainly to maintain a conversation with the user rather than conduct a transaction.

The history of spoken dialogue systems can be traced back to early work in artificial intelligence in the 1960s. However, it was only towards the end of the 1980s that speech was used for user input and system output. A number of different approaches have been used, including theory-driven methods such as linguistic processing, planning and representations from artificial intelligence research, as well as data-driven approaches involving various forms of pattern matching. Some of these methods are converging in current conversational systems.

As well as being a fascinating topic for researchers in universities and research laboratories, spoken dialogue technology has become commercially important over the past few years, due in large part to the emergence of the Voice Web – the convergence of the infrastructure of the World Wide Web and the use of speech technology as a mode of communication with automated systems over the telephone.

This chapter has explored the nature of spoken dialogue technology and plotted its historical development. However, so far, the nature of dialogue – how dialogue is structured, and how people engage in dialogue – has not been examined. This is the topic of Chapter 3, in which the key characteristics of dialogue are discussed and a number of theoretical approaches to dialogue are critically evaluated.

Further Reading

McTear (1987) provides an overview of research in dialogue modelling in the 1970s and 1980s and examines what is required for a computer to be able to converse with humans using natural language. Markowitz (1996) is a good account of the applications of speech technology. Raman (1997) is a detailed account of how to develop auditory user interfaces that are particularly useful for users with visual impairment.

Dialogue and Intelligent Tutoring Systems

Publications from the University of Edinburgh tutorial dialogue group: http://www.cogsci.ed.ac.uk/~jmoore/tutoring/papers.html
Publications from the University of Pittsburgh project Spoken Dialogue for Intelligent Tutoring systems: http://www.cs.pitt.edu/~litman/why2-pubs.html

Exercises

1. Examine one of the spoken dialogue systems that you encountered in the exercise at the end of Chapter 1. Determine the extent to which the system focusses on a particular domain, for example, does it involve a restricted vocabulary and a set of grammatical structures? What would be involved in porting the system to another domain?
2. The following web sites contain links to chatterbots. Try out some of the chatterbots. Analyse your interactions in terms of how realistic the dialogues were.

Simon Laven page: http://www.simonlaven.com/
BotSpot Chatbots: http://www.botspot.com/search/s-chat.htm
Google Directory Chatterbots: http://directory.google.com/Top/Computers/Artificial_Intelligence/Natural_Language/Chatterbots/