Timo Baumann, 2009

Spoken Dialogue Technology

Springer
London
Berlin
Heidelberg
New York
Hong Kong
Milan
Paris
Tokyo

Michael F. McTear

Spoken Dialogue Technology

Toward the Conversational User Interface

Foreword by T. V. Raman

Springer

Michael F. McTear
School of Computing and Mathematics
University of Ulster
Newtownabbey BT37 0QB
Northern Ireland

British Library Cataloguing in Publication Data
McTear, Michael F., 1943–
Spoken dialogue technology : toward the conversational user interface
1. Natural language processing (Computer science) 2. User interfaces (Computer systems)
I. Title
006.3′5

ISBN 1852336722

Library of Congress Cataloging-in-Publication Data
McTear, Michael.
Spoken dialogue technology : toward the conversational user interface / Michael F. McTear.
p. cm.
Includes bibliographical references and index.
ISBN 1-85233-672-2 (softcover: alk. paper)
1. User interfaces (Computer systems) 2. Automatic speech recognition. I. Title.

QA76.9.U83M45 2004
005.4′37 – dc22 2004045619

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

ISBN 1-85233-672-2 Springer-Verlag London Berlin Heidelberg
Springer-Verlag is a part of Springer Science+Business Media
Springeronline.com

© Springer-Verlag London Limited 2004
Printed in the United States of America

The use of registered names, trademarks etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Typeset by SNP Best-set Typesetter Ltd., Hong Kong
34/3830-543210 Printed on acid-free paper SPIN 10886181

Foreword

The present coming of age of speech technologies coincides with the advent of mobile computing and the accompanying need for ubiquitous information access. This has generated enormous commercial interest around deploying speech interaction to IT-based services.

In his book, Michael gives an in-depth review of the nuts and bolts of constructing speech applications. Notice that creating good speech applications requires more than robust speech technologies; the creation of usable, efficient spoken interaction requires substantial thought toward crafting the man-machine conversation. This book does an excellent job in this regard by focusing on the creation of dialog-based applications.

As speech interaction enters the mainstream Web, these applications need to be developed, deployed and maintained against an efficient cost model. The speech industry is converging on a set of open XML-based standards developed under the auspices of the W3C in order to achieve this goal. VoiceXML, SSML and SRGF together form the underpinnings of the XML-based W3C Voice Web, and this book gives the reader a good foundation on building speech interaction using these interoperable standards. The closing chapters on multimodal interaction gives the reader the necessary background to start actively following and contributing to this exciting area.

T. V. Raman
Human Language Technologies
IBM Research

Preface

Writing this book has been a long process of discovery and learning that dates back to my first explorations in human–computer dialogue using natural language, documented in my book *The Articulate Computer* (Blackwell). Published in 1987, it described research in dialogue modeling where the primary input mode was the keyboard rather than speech. Since then things have come a long way. The fields of natural language processing and speech technology have come together, and the discipline of spoken dialogue technology has emerged. Moreover, there has been an increasing commercial interest in the technology, with a number of large, influential companies actively involved in spoken language technology and offering advanced toolkits and development platforms to researchers wishing to explore the exciting potential of conversing with a computer using spoken natural language.

Along the way I have met many people who have helped me and shaped my work. My first attempts with spoken dialogue interfaces involved acquiring and learning how to use the CSLU toolkit. This brought me into contact with Ron Cole, a pioneer in the development and promotion of freely available, open source software to support teachers and students wishing to learn about spoken language technologies. Through Ron I met several of his colleagues: Jacques de Villiers, Ed Kaiser and John-Paul Hosom at the Center for Spoken Language Understanding, who provided me with invaluable advice and support, and later, when Ron moved to the Center for Spoken Language Research (CSLR), Bryan Pellom and Wayne Ward, who introduced me to the CU Communicator and other toolkits and platforms being developed at CSLR.

More recently, as I started to learn about VoiceXML, I have found Jim Larson to be a tremendous source of inspiration and encouragement. As I moved on to look at more recent developments, such as XHTML + Voice, I found support and advice from a number of people in IBM's Pervasive Computing Group, most notably, T.V. Raman, David Jaramillo, Nick Metianu and Les Wilson. There are also a number of colleagues, who I meet regularly at conferences and other academic occasions and who have been a constant source of ideas. These include Harald Aust, Norman Fraser, Paul Heisterkamp, Alex Rudnicky, Paul McKevitt, Ian O'Neill, David Toney and Colleen Crangle.

Developing the practical chapters of the book would not have been possible without the help of many colleagues and students. At the University of Ulster Stephen Downey spent many hours helping me to get the early versions of the CSLU toolkit installed and running properly and provided untiring support both to myself and to the students we taught together in lab sessions. Lesley-Ann Black has worked with me more recently on VoiceXML practicals and she also provided useful feedback on earlier drafts of the book. Countless students, too many to name, have been subjected to practicals involving the CSLU toolkit, VoiceXML, X+V, and SALT. I would like to single out Elizabeth Rooney and Angela Murphy, on whose Master's dissertations I have drawn extensively in chapters 6, 7 and 8, and also Conor McCluskey, Garth Kennedy and Aislinn McAleer, whose excellent undergraduate dissertations provided me with many useful ideas throughout. I would also like to acknowledge Gerrit Bloothooft and his students at the Utrecht Institute of Linguistics, who provided useful feedback and corrections on chapters 7 and 8 involving the CSLU toolkit, as well as students at the ELSNET summer school at Lille, 2004, who also worked through these chapters with me.

I was encouraged to write this book by Beverley Ford, Editorial Director of Springer-Verlag London. Jean Lovell-Butt provided me with editorial assistance during the writing of the book, and Louise Farkas, Senior Production Editor, Springer-Verlag New York, oversaw the development of the book from the raw typescript that I submitted to the final version of the book. To Beverley, Jean and Louise, and all of their colleagues who helped along the way, I offer my gratitude.

Finally, I must acknowledge my wife Sandra, who has supported and encouraged me throughout, particularly during the long hours away from family life that I have spent working on the book.

Newtownabbey, Northern Ireland — Michael F. McTear

Contents

Introduction

People use speech to interact with one another in a seemingly effortless way. From an early age we speak in order to convey our wishes, express our opinions and ask questions. The ability to use written language comes later and is something that has to be taught rather than being acquired spontaneously. Conversely, when we interact with computers we use either some advanced form of written communication, as in programming languages, or a complex set of iconic symbols that we activate with a pointing device such as a mouse. Speech would seem to be the most natural way to interact with a computer, yet it is only recently that speech-based communication with computers has become possible.

The topic of this book is spoken dialogue technology. A spoken dialogue system enables people to interact with computers using spoken language, in contrast to most computing technology, which requires people to learn sequences of commands in order to use applications such as word processors, spreadsheets and email. Spoken dialogue technology is about developing applications that people can interact with using something that they have been using all their lives – their ability to speak and to understand speech.

Spoken dialogue technology has been the dream of many computer scientists for a long time, going back to the early days of Artificial Intelligence (AI) that saw the vision of an intelligent computer that could do lots of things, including interacting with people using natural language. Speaking to computers is also well-known to many of us through science fiction, with the computer HAL in Space Odyssey 2001 being one of the best-known examples. Considerable progress has been made in the various component technologies required to endow a computer with conversational abilities, such as speech recognition and natural language processing. However, it is only in the last decade or so that the conversational computer has become a reality, not only in scientific terms but also commercially. Within the past few years new technologies have emerged that have driven these advances forward. VoiceXML and SALT (Speech Application Language Tags) are new languages that are used to script spoken dialogues and to bring together the technologies of the Internet and the telephone. These languages are being proposed as standards to the World Wide Web consortium (W3C) and are supported actively by influential companies such as IBM, Microsoft and Intel.

Who Should Read This Book?

This book is intended for two types of reader:

1. Final year undergraduate and graduate students with some background in Computer Science and/or Linguistics who are taking courses in Artificial Intelligence, Natural Language Processing, Spoken Language Technology, Internet Technology, or Human Computer Interaction.
2. Professionals interested or working in the fields of Speech, Internet Technologies or Web Development.

The book aims to fill a gap between academic books on speech technology and computational linguistics on the one hand, and books for practitioners and professional developers on VoiceXML and similar topics on the other. Most of the academic texts are highly technical and suited only for those with specialist backgrounds in the relevant areas. The more practice-oriented texts, on the other hand, while showing how to develop realistic applications, tend not to provide a comprehensive overview of the background to spoken dialogue technology.

How This Book Is Organised

In keeping with the intended readership, this book is organised into three parts. The chapters in Part I examine research in dialogue and provide an introduction to the technologies involved in the development of a spoken dialogue system. Part II is concerned with how to develop spoken dialogue applications using some readily available development tools. The aim of these chapters is to enable readers to get a good feel for what is involved in developing spoken dialogue systems – to see how easy it is in some ways to develop an application, but also to appreciate some of the pitfalls and difficulties. Finally, the chapters in Part III identify and discuss some current research directions in order to provide the reader with an insight into how the technology is developing and how it might look in the future.

The following is a brief summary of the contents of each chapter.

Part I. Background to Spoken Dialogue Technology

Chapter 1. Talking with Computers: Fact or Fiction?

This chapter introduces spoken dialogue technology and illustrates aspects of the technology that have already been achieved in research laboratories and, in some cases, in commercial products.

Chapter 2. Spoken Dialogue Applications: Research Directions and Commercial Deployment

This chapter describes a range of application areas that are suitable for spoken dialogue technology and presents a brief historical review of research in this area. The commercial potential of the technology is explored.

Chapter 3. Understanding Dialogue

This chapter is concerned with the key characteristics of dialogue. It is important for developers of dialogue systems to appreciate how dialogues are structured and how people engage in dialogue. A number of influential approaches to dialogue are examined.

Chapter 4. Components of a Spoken Dialogue System – Speech Input and Output

In this chapter the main contributing technologies for spoken dialogue systems are discussed – speech recognition, language understanding, language generation, and text-to-speech synthesis.

Chapter 5. Dialogue Management

This chapter introduces dialogue management – the central component of a spoken dialogue system. Techniques for dialogue management are reviewed and illustrated.

Part II. Developing Spoken Dialogue Applications

Chapter 6. Dialogue Engineering: The Dialogue Systems Development Lifecycle

In this chapter the development lifecycle of a spoken dialogue system is examined and issues of best practice are discussed.

Chapter 7. Developing a Spoken Dialogue System Using the CSLU Toolkit

This chapter shows through a series of tutorials how to develop a spoken dialogue system using RAD. RAD is a component of the CSLU toolkit, a freely available development platform that has been widely used in educational environments to support research and teaching of spoken language technology.

Chapter 8. Developing Multimodal Dialogue Systems Using the CSLU Toolkit

This chapter shows how to develop multimodal dialogue systems in RAD. A series of educational applications for teaching elementary concepts to young children are developed. These applications use a talking head along with pictures and sound recordings.

Chapter 9. Developing a Directed Dialogue System Using VoiceXML

This chapter follows the same pattern as Chapters 7 and 8 with a series of tutorials on basic aspects of VoiceXML. The focus in this chapter is on directed dialogues, in which the system controls the dialogue and the user responds to prompts spoken by the system.

Chapter 10. More Advanced VoiceXML

This chapter examines some more advanced aspects of VoiceXML, such as mixed-initiative dialogue and the generation of dynamic VoiceXML in Web server applications. Some important aspects of VoiceXML are explained, such as the Form Interpretation Algorithm, the structure and use of recognition grammars, and the issue of scope in relation to variables and grammars.

Chapter 11. Multimodal Web-based Dialogue: XHTML + Voice and SALT

In this chapter two new languages for multimodal web-based applications – XHTML + Voice and SALT – are introduced. The elements of these languages are explained using simple examples.

Part III. Advanced Applications

Chapter 12. Advanced Dialogue Systems: Some Cases Studies

This chapter examines more advanced dialogue systems that are being developed in a number of research centres. Several well-known examples are presented as case studies.

Chapter 13. Research Topics in Spoken Dialogue Technology

In this chapter a number of current research topics in spoken dialogue technology are examined, including Information State Theory, error handling, adaptive dialogue systems, and systems that learn an optimal dialogue strategy using machine learning techniques.

Chapter 14. Future Directions: The Way Ahead

This chapter examines some issues for the future, including the integration of spoken dialogue technology into advanced multimodal applications. Future directions for academic and industrial research are explored.

Part I
Background to Spoken Dialogue Technology

Talking with Computers: Fact or Fiction?

1

A Little Fantasy

The alarm goes off at 6:30 a.m. Rachel stirs slightly, turns over and mutters "snooze". Ten minutes later the alarm goes off again. Rachel wakes up and says "off". She stretches, yawns and drags herself out of bed. She turns to the small box beside her bed and says "What's the weather like today?" "In Belfast this morning, cloudy at first with some sunshine later, temperature 12 degrees in the morning rising to around 16 in mid-afternoon," comes back the response from the box. "How about tomorrow?" "The outlook for Tuesday is some light showers at first with heavier rain and possibly some thundershowers spreading from the west towards late afternoon." "And in Barcelona?" "The weather forecast for Barcelona is sunny, with temperatures rising from 16 degrees in the early morning to 25 in mid-afternoon." "What about the Costa Brava?" "I'm sorry," replies the box, "I do not know the place you are asking about. Try asking for a city or country. For example, what cities do you know about in Spain?" "Never mind", responds Rachel with a hint of slight irritation on her way to the bathroom.

After her shower Rachel goes into the kitchen, takes a piece of bread from the bread bin and puts it into the toaster. "Not so well done this time." She goes to the fridge, takes out a carton of milk, and notices that it is almost empty. "Don't forget to order another carton of milk", she says to the fridge. "You're having some friends round for hot chocolate later, maybe I should order two cartons", says the fridge. "Okay", says Rachel.

After breakfast Rachel starts to think about the day ahead. She presses a button on the small device on her wrist that displays the time. The device greets her and asks what services she requires. "What meetings do I have today?" asks Rachel. "You have a meeting with Paul at 10, then a meeting with Anna at 11", responds the device. "Call Anna", says Rachel, and the device places a call to Anna. Rachel speaks with Anna to change the time of her meeting. She tells the device to update her calendar and asks to check her email. When she is finished she says "Thanks, bye for now" to the device on her wrist, and gets ready for the day ahead.

How realistic is this scenario? Is it a vision of the future or is it possible today? In this chapter we will see that most of what is described in this short story is not only possible but is in some cases already available commercially. However, before jumping to the conclusion that the Star Trek computer is ready for us to order from the local computing store, we need to ask a few questions about the sorts of technologies that are involved in this scenario and how exactly we can use them.

How Do You Talk to a Computer?

Typically, speech to a desktop computer evokes the image of a person sitting in front of a PC wearing a headset with speakers and speaking into a microphone that is connected to the computer's soundcard. However, this set-up is not required in order to talk to a computer. One of the most common ways to talk to a computer is over the telephone. This mode of conversation with computers is the main topic of this book. As well as this, however, microphones and speakers are increasingly being embedded into appliances or on to surfaces such as walls or automobile dashboards. These can be connected, using wireless technologies such as Bluetooth, to computers and processors that may be embedded in the appliances or worn on the body. In this way it is possible to speak to appliances such as a toaster, VCR or car audio system.

A number of companies specialise in embedded computers. Sensory Inc. has a range of products that allow people to control appliances and machines by speaking to them and by receiving spoken responses from them (www.sensoryinc.com). Sensory's speech technologies include speech recognition, speaker verification, speech and music synthesis, and animated speech. Their Integrated Circuits (ICs) and embedded software are used in consumer electronics, cell phones, Personal Data Assistants (PDAs), internet appliances, interactive toys, automobiles and other appliances. Sensory has also developed technologies for interactive kiosks that assist customers to find the products they are looking for in large stores (Sensory Inc., 2002). In addition to speech recognition and speech synthesis, the kiosk screen displays an animated character that uses facial animation and lip-synching to provide a human-like "persona" for the application.

HAL (Home Automated Living) is a system that allows a person to speak to and control simple devices in the home, such as lights and thermostats, as well as more complex appliances, such as video recorders and digital music players (www.automatedliving.com). The user simply picks up a phone and issues a command. HAL confirms the command and performs the action. HAL can be controlled from outside the home by phone, indeed from anywhere in the world. For example, the user can call HAL to close the garage door or turn on the front door lights when returning home late at night. HAL works on the basis of a series of preprogrammed commands to a device that are associated with and activated by spoken phrases.

Simple applications involving embedded appliances are a matter of the speech technology component capturing the acoustic signal that represents what the user has said, and translating it into a digital code or electronic signal to perform functions such as light control or opening and closing doors. These devices do not normally need to interact with the user, except to confirm that the instruction has been understood.

At a more complex level the appliances may need to engage in dialogue with the user to clarify the instructions or to take the initiative in the dialogue. These points are illustrated in the following scenarios taken from a demonstrator developed in the D'Homme project, an EU sponsored research project that examined language understanding and dialogue management issues for controlling and interacting with networked devices in the home (Milward et al., 2001).

1.1 (Scenario 1)
1 User: Turn the light off. (There is more than one salient light.)
2 System: There is more than one: the living room light, the kitchen light.
(The humidity sensor in the kitchen pot plants reaches its threshold.)
3 User: The kitchen light.
4 System: The kitchen light is now off.
5 User: By the way, the kitchen humidity sensor was triggered.

The instruction in utterance 1 is ambiguous because there is more than one light that could be turned off, so the system initiates a clarification query in utterance 2. At this point the system notices that the humidity sensor in the kitchen pot plants has reached its threshold. As this is not a critical event, the need to inform the user is deferred until the task of switching off the light is completed.

1.2 (Scenario 2)
1 User: Turn off the light. (There is more than one salient light.)
2 System: There is more than one: the living room light, the kitchen light.
(The smoke sensor in the hall reaches its threshold.)
3 System: WARNING! The smoke detector in the hall has been triggered.
(User runs upstairs to find her teenage son smoking an illicit cigarette out of the window.)
4 User: Ok. False alarm.
(The system assumes that utterance 4 closes the subdialogue.)
5 System: Now where were we? There is more than one light.
6 User: The living room light.

This scenario begins in the same way as Scenario 1 but, in this case, the system detects a critical event (the smoke sensor in the hall reaches its threshold) and interrupts the current task to inform the user. After utterance 4 the system assumes that the interruption subdialogue has been completed and resumes the original task, marking the resumption with the discourse marker "Now, where were we?"

Embedded speech technology is an active research area. A number of companies that specialise in speech technology in general, such as IBM (www.ibm.com/speech), Scansoft (www.scansoft.com) and Nuance (www.nuance.com), have moved recently into the embedded speech sector, while companies such as Xybernaut (www.xybernaut.com) have pioneered the development of wearable computer technology that includes voice activation. Embedded speech technology provides an alternative interface to the knobs, switches, buttons and digital displays of consumer appliances. Devices are becoming increasingly complex and, as the range of features increases, it has become more and more difficult for users to produce the appropriate sequences of key presses to set a control. A typical example is the inability of most people to use their remote control to set the timer on their video recorder to record programs broadcast at a later time or date. Moreover, as devices decrease in size, manual manipulation has become more difficult, if not impossible. There are, however, a number of challenges for speech technology, in particular, the issue of robust speech recognition in noisy environments such as a car, requiring techniques such as echo cancellation, noise subtraction and other noise reduction techniques.

What Kinds of Things Can You Ask a Computer to Do?

Speech to computers can be used for a number of different purposes. For example, Rachel used speech to control the alarm clock ("snooze" and "off"), to give instructions to the toaster ("not so well done this time") and to the fridge ("don't forget to order another carton of milk"), to ask for the weather forecast, and to check and amend her appointments. Let us look at some of the more common functions and examine what is involved.

Controlling Devices

Controlling devices are often referred to as a "Command-and-control" function (Markowitz, 1996). Command-and-control involves the use of speech to control equipment, which can include computers, consumer appliances and machinery. There are a number of products that enable users to issue commands to their computer applications, such as word processors, spreadsheets and operating systems, and to control consumer appliances, as described earlier. Some of the earliest applications of command-and-control were battle management applications for military vehicles, and there have been many applications for manufacturing equipment control.

Command-and-control is appropriate for "hands-busy" situations, where the user's hands are not free to control the equipment. For example, car drivers can control the car stereo or dial a number on their car telephone without taking their hands from the steering wheel or their eyes from the road. Users of computer software can also combine speech commands with traditional mouse and keyboard interfaces to accomplish some functions more easily. For example, issuing the speech command "Create a table with five rows and four columns" might be more convenient in Microsoft Word than the mouse sequence `Tools` → `Insert Table`, then selecting the required numbers of rows and columns from the Insert Table dialogue box.

Command-and-control is also useful for people with disabilities who would not otherwise be able to manipulate the control functions of the equipment. The Hill-Rom Enhancemate product, for example, described in Markowitz (1996), enables people with severe physical disabilities to control specially equipped hospital beds as well as other objects in their immediate environment.

Many command-and-control applications have a small vocabulary size (between 10 and 50 words), reflecting the operations required to control the equipment. Thus commands to control the lights might include "on", "off", "dim" and a few more words, depending on what additional operations are available. More complex systems such as military aircraft cockpits can require up to 5000 words to control the equipment. As far as embedded consumer products are concerned, a small vocabulary size is essential in order to minimise memory and CPU resources in the target system. Thus Sensory's voice activation software is limited to between 50–100 phrases in speaker-dependent mode and 10–20 phrases in speaker-dependent mode.

Most command-and-control applications involve the use of single words or phrases, such as "yes", "open garage door", or "lights on". Usually there is a straightforward mapping between the word or phrase and its semantics, i.e., the action to be carried out or the meaning to be associated with the words. More complex commands and queries can involve sentence-like structures, such as "open the garage door at 7 a.m." or "Every Saturday and Sunday turn living room lights on at 6 p.m. for three hours." These commands may map directly on to a set of actions such that each phrase is associated with a particular action set. In this case no complex language processing of the phrases is required. However, as the number of phrases and alternative wordings increases, this approach becomes unmanageable. For example, in the last example alternative words could be substituted for the days "Saturday" and "Sunday", for "turn on", "living room lights", "6 p.m." and "for three hours." In some cases the range of alternative words might be restricted, for example, to days of the week or times of the day. Even so, as the number of alternative wordings increases, the task of listing all possible combinations and associating them with a given set of actions becomes unmanageable and so a grammar is required that specifies, in a more abstract way, the words and phrases along with their permissible combinations. A wide coverage system would also need to include different ways of saying the same thing, for example, "turn the lights on at 7" as opposed to "at 7 turn the lights on" or "at 7 turn on the lights".

Data Entry

Data entry involves the use of speech to enter data into a software program. Many early applications were used by the military as well as in factories and warehouses to enable users to perform several tasks at the same time, such as monitoring equipment while entering data. Most data entry applications involve well-defined data items that can be entered into specific software programs. Applications include form completion, package sorting, equipment maintenance and traffic accident reports. Changing an appointment, as in the fantasy scenario, is one example of data entry.

Data entry applications usually involve a small range of words, such as digits, "yes", "no" and a few additional control words. However, there may also be a requirement for a larger number of application-specific words, depending on the application type. For example, a reporting system for an accident reporting system might require vocabulary for the names of all the streets and locations in a city.

Getting Information

There are already a large number of speech-based applications that enable people to obtain information about the weather, travel schedules, share prices and many other information sources and services by calling a number on the telephone and speaking to the system in a fairly natural conversational mode. The example in the fantasy scenario is based on a conversation between Victor Zue, Head of the Spoken Language Systems Group at MIT's Laboratory for Computer Science, and Jupiter, a weather information system (Zue, 1999). Jupiter is one of the applications that have been created on MIT's Galaxy architecture – others include Pegasus, which provides schedules for around 4000 commercial airline flights in the United States, and Voyager, which is a guide to navigation and traffic in the Boston area. These applications provide up-to-date information in near real-time and can be accessed by telephone. In the period from May 1977 to 1999 Jupiter had fielded more than 30,000 calls and had achieved a correct understanding rate of around 80% of queries from first-time users. Calls are recorded and processed so that the system's performance can be evaluated and to provide data for further research.

Data access and information retrieval systems require vocabularies that reflect the names of the database fields and their contents. A weather information system, for example, would require the names of all the cities and regions for which weather reports are generated.

Dictation

Dictation involves the automatic translation of speech into written form, either while the speaker is talking or with the use of a prerecorded speech sample as input to the system. Current dictation systems also allow the user to format the text using voice commands. Dictation is different from the other functions in that the user's input is not interpreted. If the user chooses to speak nonsense, this will be converted blindly by the system into text. There is no dialogue with the system, as in information retrieval applications, and a different type of speech technology is involved, as dictation applications require a large general vocabulary as opposed to the smaller more domain-specific vocabularies of other application types. However, dictation could be embedded within an interactive system. For example, an email reader could allow the user to dictate a response to an email. Currently, however, speech-based email systems record the user's message and send the recording as a voice file.

Dictation systems require large vocabularies and, in some cases, an application will include a specialist vocabulary for the application in question in addition to a more general vocabulary. Vocabulary size is a two-edged sword for speech systems. On the one hand, the smaller the vocabulary the less chance there is that similarly sounding, and thus more easily confusable, words are included. On the other hand, the larger the vocabulary the less likely the system is to encounter an unknown word. Unknown words are a major source of speech recognition errors as they lead to an incorrect guess in which the system assigns the unknown word to a word that is actually within its vocabulary.

A Speech-enabled Virtual Personal Assistant

In the fantasy scenario Rachel was able to call on a virtual assistant to check and rearrange her appointments, place calls and check her email. These tasks involve a combination of functions as well as the integration of speech technology with the telephone. There have been a number of successful speech-based telephone applications since the beginning of the 1990s, such as voice-activated dialling and automated directory enquiry. Call management is a more recent application type, while the most recent innovation involves the integration of speech technology, the telephone and the Internet into what has come to be known as the Voice Web.

One example of such an application is the Voice-Enabled Virtual Assistant, developed by Vialto and running on the VoiceGenie VoiceXML Gateway (VoiceGenie, 2003). The Virtual Assistant handles a number of automated functions, such as looking up and dialling telephone numbers, retrieving email and consulting and updating calendar entries. Users can speak commands such as "Call John Smith at work", "Appointments for January 18 after 11a.m.", and "Read email from John Smith".

Dialogue with a Computer

In some applications the system receives a simple command from the user, such as "off", and the system responds with the required action. In such a case the dialogue between the user and the system is minimal. In other cases there may be a more extended dialogue. For example, in a flight enquiry system the system may have to elicit a number of items of information from the user, such as destination city, departure city, date and time of travel, and so on, before it can retrieve the appropriate information from a database. Extended dialogue introduces a number of additional issues. On the one hand, a dialogue is more than a set of utterances that follow one another in some arbitrary sequence. For this reason it is necessary to keep track of what has been said in a dialogue and to relate new utterances to what has been said previously. A second issue is that in dialogues people sometimes mishear what the other person said or take it up in the wrong way. Thus there needs to be some way of dealing with errors and misunderstandings. The nature and complexity of dialogue will be discussed in Chapter 3, while Chapter 13 will present some ways for keeping track of what has been said in a dialogue and for recovering from errors.

Relations Between Utterances in Dialogue

There are a number of ways in which the utterances in a dialogue can be related to one another. First, the utterances of each speaker should relate to each other in terms of their function and content. So, if the user asks a question, then the system should provide an answer to that question. More specifically, if the question is for information such as a departure time, then the answer should include that information. There are, of course, cases where this basic prescription may be violated, for example, when the system does not have the required information. In such a case the user's question still needs to be addressed, possibly with some account for the inability to answer it.

A second aspect of relations between utterances is that words or phrases may refer back to something previously mentioned. For example, the user might say "Call Anna", followed by "No, change that to Paul". The system needs to know how the second command relates to the first and how to determine what the word "that" refers to. In this case the name of the callee is to be changed. In a different case, the command "No, change that to email" would need to be interpreted as a change in the medium of the communication. More generally, there are many ways in which words and phrases are related within a dialogue and there are a number of theories that address their use and interpretation within a discourse context (see Chapter 3).

A final example of the relations between utterances in a dialogue concerns topical coherence. In the simplest systems only one topic is involved, such as obtaining flight information. More complex systems may include several functions. For example, the scenarios modelled in the DARPA Communicator systems include flight information and reservations, car rental and hotel reservation (http://fofoca.mitre.org). One method of dialogue control would be to require the user to complete each function before moving on to the next. However, a more flexible system might allow the user to move between topics, for example, to decide on whether a car is required based on the location of the hotel that is to be reserved. In such a case, the system needs to keep track of what has been discussed and agreed in order to be able to return to topics that have not been closed.

Dealing with Errors and Misunderstandings

There are many ways in which errors and misunderstandings can arise in a dialogue. One example of these is illustrated in the fantasy scenario when Rachel asks about the weather forecast for the Costa Brava. This mirrors an actual exchange between Victor Zue and Jupiter in the MIT weather information system (Zue, 1999):

1.3
1 V.Z.: Is it sunny anywhere in the Caribbean?
2 Jupiter: Sorry, I am not sure what you said.
3 V.Z.: Is it sunny anywhere in the Caribbean?
4 Jupiter: I am sorry, I may not know the city you are asking about. Try asking for the state or country. For example, what cities do you know about in Massachusetts?

In the first exchange (1–2) the system responds that it is unsure about the input. One reason could be that one or more key words have not been correctly recognised (a speech recognition issue). As speech recognition cannot be guaranteed to be completely accurate, even in conversations between humans, a spoken dialogue system must include mechanisms for detecting and dealing with misrecognitions. The simplest ploy is to get the user to repeat, as happens in this exchange. However, there are many more sophisticated techniques that can be employed to resolve recognition problems (see Chapter 13).

In the second exchange (3–4) it becomes obvious that misrecognition is not the problem. The user has asked about something that is outside the scope of the system's knowledge, in other words, the item mentioned is not represented in the database. Handling this sort of problem is more complex. The simplest approach is to state that the system does not have any information about the item in question and invite the user to submit a different query. However, this approach is not particularly helpful and could involve the user in submitting several similar queries that the system cannot answer and subsequently giving up in frustration. An alternative strategy is to detect the source of the problem and to propose a solution. In this case, the system makes clear that it can only answer queries about cities, and provides an example of a query that will inform the user about which cities in particular it can handle. While the solution to this particular issue appears simple, it is a major challenge to provide a more general solution that would enable the system to behave in a cooperative way in the face of similar conversational misunderstandings.

Advanced Functions

Some of the features illustrated in the fantasy scenario are beyond the capabilities of current technology and are the subject of ongoing research. For example, when speaking to the toaster, Rachel said "Not so well done this time." There are several ways in which this short phrase would be problematic. The meaning of the expression "well done" is relative to the extent to which the bread had been toasted on the previous occasion, rather than a description of an absolute degree of toasting. If, on the previous occasion, the bread had been toasted to a moderate degree, then the expression would have the meaning "a little less than moderate". If, however, the toast had been burnt, the expression would have the meaning "a little less than burnt". Furthermore, the expressions "moderate" and "burnt" are themselves relative, but at some point there would need to be a correspondence between expressions, degrees of toasting and toaster settings for these expressions to be interpretable.

Another example is the brief conversation with the fridge about ordering milk:

1.4
1 Rachel: Don't forget to order another carton of milk.
2 Fridge: You're having some friends round for hot chocolate later. Maybe you should order two cartons.

In this episode Rachel instructs the fridge to order more milk. However, instead of accepting Rachel's instruction, the fridge makes a counter suggestion along with a justification. To perform this sort of function, the system needs to be able to do more than simply process instructions and translate them into actions or look up items in a database. Complex reasoning is required involving knowledge of Rachel's plans and what is required for these plans to be achievable.

Reasoning such as this has not been implemented in currently deployed systems, although there are a number of research projects that focus on the integration of reasoning and spoken dialogue technology. One of these is the TRIPS project, which is based in the Conversational Interaction and Spoken Dialogue Research Group at the University of Rochester (Allen et al., 2001a, b). The following is an example of an interaction with the TRIPS system (taken from Allen et al., 2001a, p. 40).

1.5
1 User: We need to get the woman in Penfield to Strong.
2 System: OK.
3 User: What vehicles are available?
4 System: There are ambulances in Pittsford and Webster.
5 User: OK. Use one from Pittsford.
6 System: Do you know that Route 96 is blocked due to construction?

The TRIPS system models dialogue as interactive collaborative problem solving. Unlike most other dialogue systems, in which the system responds to the user's questions and commands by looking up a database or executing some action, in TRIPS the system and user engage in dialogue to solve a real-world task. In the example above, which is a rescue task scenario, the objective is to identify a suitable vehicle and route to move an injured woman to hospital. The user makes a proposal in utterance 5 but the system responds with a possible objection to this proposal in utterance 6. To do this the system must be able to reason about the task at hand and evaluate the viability of the proposed solutions. The TRIPS system is described in greater detail in Chapter 12.

Summary

From the examples that have been presented in this chapter it can be seen that the vision of the talking computer, as depicted in science fiction (e.g., Star Trek and Space Odyssey 2001), is already verging on reality. Progress in speech technology is the result of a number of factors:

1. *Technological advances in computer hardware.* Microprocessors have increased dramatically in speed and power over the past few decades so that they are capable of handling the complex mathematical calculations required for advanced speech technology applications in real-time. Miniaturisation of hardware – the ability to put increasingly powerful components on to smaller chips – has facilitated the embedding of speech technology into consumer appliances. Finally, dramatic reductions in the prices of processors, storage and memory have made speech applications more economically feasible.

2. *Developments in software.* More advanced algorithms have been developed to cope more adequately with tasks such as speech recognition, language understanding, dialogue management and sophisticated reasoning. Moreover,

in addition to advancements in each of these component areas, significant advances have been achieved in the integration of the components into working spoken dialogue systems, either as research demonstrators or as commercial products.

3. *Infrastructure for the Voice Web.* With the rapid development of the World Wide Web over the past decade, an infrastructure and universally accepted interface for distributed information and communication systems have been created. The integration of this infrastructure with speech technology, using the telephone rather than the PC, has made possible a new and powerful interface to Internet-based services and information.

4. *Commercial impetus.* A number of commercial benefits have been suggested for speech technology, such as increased productivity, rapid return on investment and access to new markets. This commercial impetus has in turn driven industrially based research in a number of large corporations, such as Microsoft, IBM and Scansoft, and has led to the emergence of several companies specialising in speech technology, such as Nuance, VoiceGenie, BeVocal and Voxpilot.

The next chapter will examine the sorts of application areas that are suitable for spoken dialogue technology and will review the main research thrusts as well as the commercial potential of the technology.

Further Reading

The fantasy scenario presented in this chapter was inspired by a similar scenario "A Day in Jack's Life" presented in a paper by James Larson entitled "Speech-enabled appliances", which appeared in *Speech Technology Magazine* (Larson, 2000).

Speech Technology Magazine is a bimonthly online publication with in-depth papers on the development, application and integration of speech technology for business and consumer applications, covering recent advancements and future trends. Available at: http://www.speechtechmag.com/

Some Other Interesting References

Phillips's "Vision of the Future Project" describes the home of the future, in which wands are personal preference remote controls that are programmed to personalise and preselect media around the house. Simple voice commands offer an intuitive interface to home systems in this vision. Available at: http://www.design.philips.com/vof/toc1/home.htm

Hal's Legacy: 2001's *Computer as Dream and Reality* (Stork, 1998). This is a collection of papers by leading researchers in Artificial Intelligence (AI) describing to what extent the technologies imagined in the 1960s by Arthur Clarke and Stanley Kubrick in Space Odyssey 2001 are possible today.

The Age of Spiritual Machines: How We Will Live, Work and Think in the New Age of Intelligent Machines (Kurzweil, 2001). This book by Ray Kurzweil predicts the future of technology, suggesting that by 2020 computers will outpace the human brain in computational power.

Kurzweilai.net. This web site (http://www.kurzweilai.net/) covers technology of the future and also includes Ramona, a photorealistic, interactive, lifelike chatterbot with which you can converse on the Web. Using natural language processing techniques, Ramona conducts conversations with visitors, responding to typed questions or comments with a life-like face, lip-synched speech and appropriate facial expressions.

ELSNET's Roadmap for Human Language Technologies. ELSNET (European Network of Excellence in Human Language Technologies) is developing views on and visions of the longer-term future of the field of language and speech technologies and neighbouring areas (http://www.elsnet.org/). For the current state of the roadmap, see: http://elsnet.dfki.de

Exercises

The following web sites contain links to spoken dialogue systems, some of which can be tried out by dialling a telephone number. Others can be viewed as demos. Try out some of the systems listed and play some of the demos. Make a note of some systems that you could use for exercises in later chapters.

IBM

http://www-3.ibm.com/software/pervasive/tech/demos/voice_server_demo.shtml (WebSphere Voice Server demo: demonstrates the ease of accessing Internet information over the phone, including movie schedules, nutritional planning tool, and IBM directory dialer).

Nuance Communications

http://www.nuance.com/solutions/bankingcredit/index.html (several demos in the area of financial services: ATM locator, banking transfer funds, and others).

http://www.nuance.com/solutions/utilities/index.html (several demos in the area of utilities: meter reading, start a service, transfer a service).

Scansoft

http://www.scansoft.com/network/solutions/ (demos for financial services, healthcare, telecom, travel and hospitality, utilities).

VoiceGenie

http://www.voicegenie.com/content/10400.html-9d (ATM locator, taxi booking, also includes phone demos).

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/

Understanding Dialogue

3

It is important for developers of dialogue systems to have a sound understanding of the nature of dialogue, how it is structured and how people engage in dialogue. This is particularly the case where the aim is to model naturally occurring conversation. Even where conversational modelling is not the main aim, as is usually the case with systems intended for commercial deployment, it is also important to understand the complexities of human dialogue, if only to know how to constrain systems in the interests of performance and the avoidance of error.

This chapter begins with a definition of dialogue followed by a discussion of its key characteristics. Dialogue is then examined from two perspectives – its structural properties and the processes involved when people (and computers) engage in dialogue.

Dialogue: A Definition

The term "dialogue" is used in everyday language to describe a process of exchanging views, sometimes with the purpose of finding a solution to a problem or to resolve differences. Often when there is conflict between individuals, communities or nations, there is a proposal that the parties concerned should "engage in dialogue".

Dialogue may be contrasted with "conversation", a term that is generally used to describe more informal spoken interaction in which the main purpose is the development and maintenance of social relationships. Conversation is often used, however, particularly in research in the United States, to refer to more advanced dialogue systems that display human-like conversational competencies. Dialogue, on the other hand, tends to be used to signify more restricted systems that engage in specific types of interaction with a more transactional purpose, such as getting information, issuing instructions or providing a service. Often the phrase "task-oriented dialogue" is used to emphasise this function. Notwithstanding these distinctions, the terms "dialogue" and "conversation" are frequently used almost interchangeably in the literature to refer to computer systems that use spoken language to interact with people. In this book the term "dialogue" will be used generically to cover all types of spoken interaction with computers.

Key Characteristics of Dialogue

Dialogue has been studied within a wide range of academic disciplines, including linguistics, psychology, sociology, anthropology, philosophy, communication sciences and artificial intelligence. The study of dialogue has also been applied in a range of areas, such as management studies, conflict resolution and intercultural relations as a method for promoting negotiation and discussion of differing viewpoints. Much of this work is not directly relevant to current directions in spoken dialogue technology and will not be discussed further here. In the following sections some of the key characteristics of dialogue that are important in the context of spoken dialogue systems will be discussed:

- *Dialogue as discourse.* The analysis of the use of words, phrases and utterances in the context of extended discourse.
- *Dialogue as purposeful activity.* An examination of the purposes for which people engage in dialogue, the actions that they perform and the meanings that they convey.
- *Dialogue as collaborative activity.* How dialogue is best understood as a joint activity in which people work together to engage in conversational turn-taking and to achieve mutual understanding.
- *Utterances in dialogue.* The nature and form of utterances produced in naturally occurring dialogues.

Dialogue as Discourse

A dialogue consists of at least two turns, one by each speaker. A dialogue that is coherent will exhibit discourse phenomena which, broadly speaking, can be viewed as elements whose interpretation depends on the dialogue context. To participate in a dialogue it is necessary to be able to keep track of these elements. In some cases this involves maintaining a record (or history list) of entities that have been introduced into the dialogue model and that can be referred to subsequently using pronouns and other anaphoric devices, that is, linguistic expressions that are used to refer back to something previously mentioned. In other cases it is necessary to draw on a wider notion of context involving general and background knowledge.

The following examples illustrate some of the issues involved in resolving anaphoric reference using pronouns:

3.1
1 A: John won some money.
2 B: What is he going to do with it?

In this example it can be assumed that "it" refers to "some money". The pronoun matches syntactically (whereas "them" or "him" would not), and it refers back to the most recently mentioned entity. Many dialogue systems have been constructed using these two simple strategies to resolve reference. However, it is not difficult to find examples where these strategies do not return the correct result:

3.2
1 A: John won some money in the lottery.
2 B: What is he going to do with it?

The two simple strategies would propose "the lottery" as the referent for "it", yet clearly this is an unlikely interpretation. An additional strategy that would select the item "some money" as the correct referent would be to locate the focus of attention in A's utterance (i.e., "the money") and to propose this as the most likely referent. Similarly, with this example:

3.3
1 A: Jim caught up with Bill outside the pub.
2 B: Did he give him the tickets?

Using the strategy of recency, "he" would be matched with "Bill", i.e., B is asking whether Bill gave Jim the tickets. (Note also that once "he" is assigned to Bill, then "him" cannot also refer to Bill but has to refer to "Jim".) However, using a different strategy, in which the subject of a sentence is preferred over the object of the sentence as being more likely to be the centre of focus, Jim would be the preferred referent of the pronoun "he". Of course, this interpretation could turn out to be wrong if other knowledge could be brought to bear, for example, if A and B both knew that it was Bill who had the tickets.

In some cases, background (or general) knowledge is required to resolve reference. Consider the following example:

3.4
1 A: Did you see Bill in the pub last night?
2 B: No, the barman said he left early.

Definite descriptions, such as "the barman" are often used to refer to some entity that has been previously mentioned in the dialogue. For example, "There's a new barman in the pub" followed later by "What do you think of the barman?" However, in the dialogue presented here, no barman has been mentioned, yet A would be unlikely to have any problem understanding B's utterance in terms of finding a referent for the expression "the barman". The explanation in this case is that A can draw on background knowledge that a barman works in a pub, so that once a scenario involving a pub is introduced, persons and objects relevant to that scenario can be mentioned using definite reference.

Keeping track of elements within a dialogue also applies at a higher level to the different topics that are introduced during the dialogue, as participants need to be able to keep track of shifts in topic in order to resolve reference to previously mentioned items. The following piece of dialogue, in which A is helping B to install some software, illustrates this point:

3.5
1 A: Click on the "install" icon to install the program.
2 B: OK.
3 B: By the way, did you hear about Bill?
4 A: No, what's up?
5 B: He took his car to be fixed and they've found all sorts of problems.
6 A: Poor Bill. He's always in trouble.
7 A: OK. Is it ready yet?

The referent of "it" in A's last utterance (7) is not Bill's car, which is the most recently mentioned element that matches syntactically, but the program that B is installing. In this case the pronoun and its referent are separated by several turns. (Indeed, in one example cited by Grosz (1978, p. 246), the pronoun and its referent were separated by 60 utterances.)

How do participants keep track in cases such as this, as clearly, keeping history lists is not the solution? It has been suggested that participants keep track of the topics introduced in the dialogue, noting shifts to new topics, subtopics, and back to previous topics. In the example presented here, the main topic is the installation of a program. The intervening turns are part of an unrelated topic (or digression) about Bill and his car. However, the main topic remains open and it is possible to refer to elements that belong to the main topic later in the dialogue using anaphoric devices such as pronouns and definite descriptions. In this example the beginning of the digression is signalled by the phrase "by the way" and its end is signalled by the word "OK". Thus to process a dialogue containing a number of topics and to keep track of the entities mentioned and the ways in which they can be referred to subsequently, it is necessary to maintain a representation of the structure of the dialogue in terms of topics and subtopics.

Spoken dialogue systems often involve tasks such as flight reservations that can be broken down into subtasks, such as getting the flight details and making a reservation. Voice portals are similar, as users can make enquiries about a number of topics and services. In the simpler systems the system uses a menu and maintains strict control of the topics. The system offers a choice of topic to the user and the dialogue follows the topic selected, then another choice of topic is offered. The following is an example:

3.6
1 System: You can ask about restaurants, traffic, news, sport or weather.
2 User: Traffic.
⟨traffic dialogue⟩
3 System: What next? You can ask about restaurants, traffic, news, sport or weather.
4 User: Sport.
⟨sport dialogue⟩

A more advanced system would enable the user to take the initiative and switch topics, as in this example:

3.7
⟨Event dialogue in progress⟩
1 System: What sort of event are you interested in?

2 User: A concert.
3 System: There is a concert in the university grounds at 9 this evening.
4 User: What's the weather forecast?

Here the system is engaged in a dialogue about events and the user shifts to a different topic – the weather. Being able to switch topics in this way provides a more flexible system that can address the user's concerns as they arise. However, the disadvantage is that the system has to have a more advanced speech and language processing capability in order to be able to process "out-of-topic" utterances. When topic shifts occur, there may also be a need to keep track of previous topics that may not have been closed in order to be able to return to those topics.

Dialogue as Purposeful Activity

People engage in dialogue for a wide range of purposes, including transactional functions such as requesting, promising, persuading and informing, and interpersonal functions such as maintaining and regulating social relationships. One way to approach dialogue is in terms of the linguistic actions that people perform in order to carry out these functions. Utterances are produced with the aim of achieving certain effects within particular contexts. The following aspects of this use of language will be considered here:

- How the meanings of utterances depend on the context in which they are used.
- How the selection of an appropriate form of language depends on aspects of the context.
- How participants in dialogue often convey meanings in their utterances that go beyond the meanings of the actual words used.

Meaning and Context

To illustrate how meanings depend on context, consider the utterance "It's hot in here." In different contexts this utterance can have many different functions, some of which are as follows:

1. A neutral statement about the temperature.
2. A statement about the temperature with the purpose of persuading the other person to come in from the cold.
3. A statement about the temperature with the purpose of persuading the other person to go out into a cooler place.
4. A statement about the temperature with the purpose of requesting the other person to do something to change the situation, for example, by opening a window or turning on the air conditioning.

As can be seen from the different contexts described in 2–4, the meaning of the utterance depends on factors such as the physical situation and the relative utterance to be intended and understood as a command are (based on Searle (1969, p. 66)):

1. The utterance is concerned with some future act that the hearer should perform.
2. The hearer is able to do the act, and the speaker believes that the hearer can do the act.
3. It is not obvious to the speaker and hearer that the hearer will do the act in the normal course of events.
4. The speaker wants the hearer to do the act.

These conditions incorporate the intuitions that people normally only ask others to do actions that they want to have done, and that they believe the other person is able to carry out the act and would not otherwise have done so without being asked. The conditions underlying the performance of speech acts have also been used to explain the use of indirect speech acts, in which the function of an utterance is not obvious from its form, as in some of the examples discussed above.

Performing a speech act such as a request involves a dialogue agent in reasoning about beliefs, desires and intentions. In making a request an agent must want the action to be done and must bring about a situation in which the other agent wants to do the action. The requesting agent must believe that the other agent is able to do the action and, as a result of the request, the other agent will believe that the requesting agent wants the action to be performed. When planning a speech act, a dialogue agent will reason about mental states such as these in order to generate the plan. Similarly, the addressee of a speech act will engage in a process of plan recognition to determine the meaning of the act in terms of the speaker's beliefs, desires and intentions. The view that dialogue can be explained in this way has been developed within a theory of dialogue known as the BDI (belief, desire, intention) approach (Allen, 1995). This approach will be examined further in Chapter 12 in order to explain more complex processes of dialogue, such as the interpretation of indirect speech acts.

Conversational Implicature: Meaning and Inference

Participants in dialogues often express meanings in their utterances that go beyond the literal meanings of the words used. The recipients of these utterances are expected to make appropriate inferences in order to determine the intended meaning. Indirect speech acts are an example of these indirectly conveyed meanings. To explain these nonliteral meanings, the philosopher Grice proposed a set of conversational maxims that enable hearers to make the appropriate inferences (Grice, 1975). Grice proposed that participants in conversation are guided by a principle of cooperation that states that they should produce utterances that contribute to the purposes and direction of the conversation. This does not mean that they should always agree or comply with each other's demands, but that they should be guided by a set of maxims in order that what they say is relevant, true, informative and clear. The most interesting part of Grice's theory involved conversational implicatures, where speakers blatantly flout one of the maxims for a specific conversational purpose. For example,

3.9
1 A: Would you lend me some money to buy a coffee?
2 B: It's not Christmas.

To make sense of B's response, A needs to read some additional meaning into it beyond what is contained in the actual words uttered. In the example presented above, it appears that B is not following the maxim of relevance because the utterance "It's not Christmas" does not seem to be relevant to A's request for money to buy a coffee. However, if it is assumed by both participants that B is observing the general cooperative principle of conversation (without being cooperative in the conventional sense of the term), then A can conclude that B's utterance must mean more than what it appears to mean and can begin a process of inferencing to work out what B might have meant. In this case some general knowledge would also be required along the lines that Christmas is a time of giving and, since it is not Christmas, giving (or lending) is not appropriate.

Conversational implicature is a very important aspect of conversational competence because people often use indirect language for a variety of purposes – in this example, to be sarcastic, in other cases to be polite, for example, to soften a request. It is also important that the hearer should be able to make the appropriate inferences in order to derive the implicated meaning, otherwise the implicature will fail. In other words, the speaker and hearer need to share assumptions about the cooperative principle and the conversational maxims, and the speaker needs to design the utterance in such a way that the hearer can infer the intended meaning. As spoken dialogue systems become more complex, they will be required to deal with more indirectly conveyed meanings. Consider the following example of a travel system:

3.10
1 System: What time do you want to depart for London?
2 User: I have to be there for a meeting at 10 a.m.

The user does not answer the system's question directly by giving a departure time. In order to make sense of the user's reply, the system has to assume that mention of the time of the meeting is relevant at this point in the dialogue and then work out a departure time that fits in with the user's requirements.

Dialogue as Collaborative Activity

Dialogue is a joint activity between two or more participants. For dialogue to proceed in an orderly fashion there has to be collaborative activity between the participants in the dialogue. Collaboration does not necessarily imply that the participants have to agree with each other, as even in dialogues where there is strong disagreement there are certain processes that participants employ to enable the dialogue to function. Conversational turn-taking and conversational grounding are important aspects of collaborative activity in dialogue.

Turn-taking in Conversation

Participants in dialogues take turns. The length of the turns and the allocation of turns are not specified in advance, except in extremely formal interactions. In naturally occurring conversations turns are negotiated on a turn-by-turn basis according to an intricate set of rules (Sacks et al., 1974). Generally in a conversation one participant speaks at a time and transitions between speakers are accomplished with a minimum of gap between turns and minimal overlap between the speakers. Conversational participants who wish to take the next turn do not appear to wait until the current speaker stops talking, as otherwise there would be regular and noticeable gaps between the turns. Similarly, they do not seem to depend solely on nonverbal or prosodic cues, such as the speaker's gaze or a falling intonation contour, but rather they anticipate the potential completion of the turn and begin at that point. The evidence for this claim is that overlaps tend to occur at transition-relevant positions, for example, at the point where the current speaker's utterance is potentially complete. The following example illustrates this (underlining indicates overlapped speech):

3.11
1 A: That's an interesting house, *isn't it?*
2 B: *Do you* like it?

The overlap occurs because the current speaker continues beyond the transition-relevant point at the end of a potentially complete sentence. The second speaker has already anticipated the potential completion and begins speaking at precisely this point.

Both participants in a conversation are involved in ensuring smooth turn-taking in conversation. As has already been shown, the potential next speaker has to listen to and analyse the current turn to be able to produce a turn that is relevant and that begins at a transition-relevant place. Likewise, speakers need to construct their turn in such a way that the hearer can project its possible completion. When overlaps occur, one of the speakers must decide to relinquish the floor as conversation normally requires that only one speaker talks at a time. Naturally, there are many cases where current speakers refuse to relinquish a turn or when a next speaker cuts in before the current speaker has completed. However, these cases can be seen as violations of the normal turn-taking rules and they are often explicitly marked by phrases such as "If you'd let me finish".

Turn-taking in dialogues between humans and computers differs from conversational turn-taking between humans in two ways. In the first place, turn-taking with computers is generally more carefully regulated. In some cases the computer indicates with a signal, such as a beep, when it is ready to accept input from the human speaker. The benefit of this turn-taking cue is that the speech recognition process only needs to start after the beep, thus saving on computational resources. A major disadvantage occurs if the speaker begins before the beep. In this case the speech preceding the beep is not captured. Furthermore, when this problem is encountered, human speakers tend to break off and repeat the part of their turn that they think has been missed, resulting in many cases in ungrammatical utterances that the system is unable to process. Problems can also occur if the speaker delays speaking after the prompt as the system may detect the leading silence and react accordingly (e.g., by outputting something like "Sorry I didn't hear anything") just as the speaker begins to speak. In this case the computer output may be captured along with the speaker's input, leading to problems for the speech recognition component.

A more advanced facility provided in many spoken dialogue systems allows the human speaker to cut in on the computer's turn, causing the computer to stop speaking and to switch to listening mode. This is known as "barge-in". Barge-in requires a speech recognition platform with full-duplex capability and echo cancellation. It is particularly useful for experienced users who are familiar with a particular dialogue style and can avoid listening to lengthy prompts by anticipating what is being said and providing a response. However, barge-in can also cause problems for the speech recognition process. If the computer's output does not terminate quickly enough, the user may increase the volume of their speech in order to speak over what the computer is saying. This is known as the Lombard effect. The problem here is that the speech signal becomes distorted as a result of loud speech, making speech recognition less reliable. The other possible effect is stuttering, where the user repeats elements of the utterance that were overlapped.

Barge-in is an example of a process that is acceptable in human–computer dialogue but would be considered inappropriate, and even rude, in dialogues between humans. As systems become more "conversational", they will need to incorporate the more sophisticated rules of collaborative turn-taking that have been found in naturally occurring conversations between humans.

Grounding

Participants in conversation cannot be sure that their utterances have been understood. There are several ways in which miscommunication can arise and participants have to actively collaborate to repair misunderstandings. The most obvious case is where something is noticeably wrong and a conversational repair is initiated. However, more generally participants in dialogue seek for and provide evidence that what has been said in the dialogue has been understood (Brennan and Hulteen, 1995). This evidence can include simple indications of attention (utterances such as "uh huh", "hm" or continued eye contact), explicit acknowledgements such as "ok" and "right" and relevant next turns or actions. Feedback is particularly important in spoken dialogue systems due to errors in speech recognition. The process of achieving mutual understanding is often referred to as "grounding".

Clark and Schaefer (1989) proposed a model of grounding based on the notion of "contribution". According to this model a contribution in a dialogue consists of two parts involving two dialogue participants A and B:

- A "presentation" phase, in which A presents an utterance to B with the expectation that B will provide some evidence to A that the utterance has been understood.
- An "acceptance" phase in which B provides evidence of understanding, on the assumption that this evidence will enable A to believe that B understood A's utterance.

The acceptance phase may take several turns including sequences of clarification requests and repairs. Once both phases are complete, it is argued that it will be common ground between A and B, that B has understood what A meant. Thus this model proposes a collaborative view of dialogue in which the participants coordinate their models of what has been understood in the dialogue on a turn-by-turn basis.

The type of feedback provided in the acceptance phase is determined by factors such as the task at hand and the dialogue model. Tasks that can be undone can use lower levels of feedback as opposed to tasks that have a high probability of error or that have destructive or embarrassing consequences, where the feedback needs to be more explicit. Similarly, if the dialogue model indicates that there have been several problems, such as misrecognitions or misunderstandings, then more explicit grounding is required.

The original model of grounding, as proposed by Clark and others, has been developed in a number of ways to address some deficiencies and to make the model more useful computationally. The basic model focuses on dialogue as a product rather than on dialogue as a process. Contributions represent the models of both participants as seen by an analyst using a transcript of a completed dialogue. However, neither partner in a dialogue is omniscient, so that the models should represent the perspective of only one of the participants (Cahn and Brennan, 1999). Furthermore, as the models develop dynamically, they need to include interim representations of the dialogue participant's current dialogue state as a basis for that agent's decisions as to what to do or say next (Traum, 1999). This more dynamic view of grounding has been developed within the "information state" approach to dialogue modelling, which will be presented later in this chapter.

Utterances in Dialogue

In much of the discussion so far, the term "utterance" has been used to refer to what a speaker says in a dialogue without any clear definition of what an utterance actually is. There is considerable debate within linguistics about the relationship between utterances and sentences, with sentences being considered roughly as abstract idealised forms that can be analysed according to well-defined rules of syntax and semantics, while utterances are considered as realisations of sentences in context. However, there are several ways in which utterances in a dialogue cannot be described in terms of grammars for well-formed sentences:

1. They have the form of partial sentences, such as elliptical answers to questions as in 2 (below) and elliptical questions as in 3:

   3.12
   1. Is the milk in the fridge?
   2. Yes.
   3. The ice cream in the freezer?

2. They contain false starts, as in:

   3.13 Did – did you see – have you ever seen a lion in the wild?

In the remainder of this section several issues relating to the forms of utterances in dialogue will be considered.

The Range of Linguistic Structures Used in Spoken Dialogue

Comparisons of spoken and written language have shown that the utterances of spoken language have a simpler form than the sentences of written texts (Brown and Yule, 1983). For example, simple active declarative forms are more common in speech, while written language contains a wider variety of forms with greater use of subordinate clauses, passive constructions, and phrases such as "however", "moreover" and "nevertheless" that indicate the structure of the written text. Moreover, studies comparing speech to a human with speech to a computer have also found that human–computer dialogues are characterised by a more restricted vocabulary, a smaller set of grammatical constructions and slower speech (Hauptmann and Rudnicky, 1988; Richards and Underwood, 1984). These findings are encouraging for developers of spoken dialogue systems as it would appear that human users talk in a simpler way to computer systems so that the grammars required to recognise and understand the user's input would not need to be as comprehensive as grammars required to process written texts.

The Form of Spoken Utterances in Dialogue

On the down side, as mentioned earlier, naturally occurring spoken language is characterised by various types of disfluency, such as hesitations and ungrammatical constructions that make analysis using traditional grammatical approaches difficult. Consider the following examples from a corpus collected from subjects using either a simulated or an actual spoken language system in the ATIS (Air Traffic Information System) project (cited in Moore, 1994):

3.14 What kind of airplane goes from Philadelphia to San Francisco Monday stopping in Dallas in the afternoon (first class flight)?

3.15 (Do) (Do any of these flights) Are there any flights that arrive after five p.m.?

Dialogue 3.14 is a well-formed sentence followed by an additional fragment or after-thought, enclosed in brackets. Dialogue 3.15 is a self-correction in which the words intended for deletion are enclosed in parentheses. Neither of these sentences could be parsed by a conventional grammar containing rules for well-formed sentences.

More generally, spoken language is characterised by a number of features of disfluency, including:

1. False starts – as in Dialogue 3.15, where the speaker begins with "Do", stops and replaces this with "Do any of these flights", then stops again and restarts with a syntactically complete question.
2. Hesitation markers, such as "uh", "err", as well as unfilled pauses (periods of silence).
3. The use of fillers, such as "and so on", "you know", and "if you see what I mean".
4. Incomplete sentences and fragments, for example, as elliptical elements in a dialogue (see 3.12.2 and 3.12.3 above).

Some of these features are sufficiently predictable that they can be described using special rules or strategies to filter out the disfluencies and produce sentences that can be parsed using a conventional grammar. For example, utterances including false starts, normally referred to as "self-repairs", have a typical structure of the form: reparandum–editing term–alteration, as illustrated in the following example:

3.16

The meeting will be	on Mon-	uh	on Tuesday
	reparandum	editing term	alteration

The reparandum is the item that is to be corrected or replaced. The editing term, often indicated by a disruption in the prosodic contour, by a word that has been cut off or by a hesitation marker such as "uh", signals that a self-repair is occurring. Finally, the alteration is the corrected version of the reparandum. Frequently there is some similarity between the reparandum and the alteration in terms of the words used as well as their syntactic structures. For example, a word in the alteration that replaces a word in the reparandum will often be of a similar word class and have a similar meaning. Given these features, it is possible to devise methods for detecting and correcting self-repairs and other types of disfluency in naturally occurring speech (Heeman and Allen, 1994).

Prosodic Characteristics of Utterances

The information that is conveyed in utterances in a dialogue does not reside solely in the words themselves, but derives also from the prosodic features of the utterance. Prosody refers to features such as the following:

- *Overall pitch contour.* This can determine the dialogue act that is being performed, for example, "OK" with a rising tone indicates a checking act, whereas a falling tone indicates acceptance or confirmation.
- *Accentuation.* The item that receives prominence is generally being marked by the speaker as being "new" to the discourse as opposed to the other items that are treated as being "given".
- *Phrasing.* The grouping of an utterance into meaningful chunks. For example, "call the ticket office in Belfast" is taken to refer to a ticket office

that is in Belfast, whereas "call the ticket office | in Belfast", with a pause between "office" and "Belfast", would convey the meaning of calling the ticket office while the hearer is in Belfast.

Prosodic information can support the text-to-speech (TTS) synthesis component of a spoken dialogue system by using the correct prosodic forms to distinguish between otherwise ambiguous dialogue acts, to indicate what information is new, and to group the utterance into meaningful chunks that will assist the hearer to more easily understand the utterance (Hirschberg, 2002). Similarly, dialogue understanding can be improved by the use of prosodic information that enables the system to track the dialogue structure, to segment utterances correctly and to predict and interpret dialogue acts (Nöth et al., 2002; Wright et al., 2002).

Nonverbal Behaviour

Nonverbal behaviour, also known as "kinesics", includes gesture, gaze and body orientation. When examining nonverbal behaviour, it is important to distinguish between those behaviours that convey information unintentionally and those that have a communicative function. Clear cases of the former would include dress and gait, which often indicate a person's affiliations and attitudes, though not necessarily intentionally. Pointing as a means of indicating something or nodding to convey assent are clear cases of intentional nonverbal behaviours that function as surrogates or accompaniments of speech. Often it is difficult to distinguish intentional and nonintentional nonverbal behaviours, as the fine-grained analyses of Argyle (1975) and others have shown. Telephone-based dialogue systems do not require the analysis of nonverbal signals. However, with the movement towards multimodal systems, including those that simulate face-to-face interaction, it is becoming increasingly necessary to consider the functions of these signals in conversational interactions.

The Role of Silence in Dialogue

Silences and pauses are closely monitored by participants in dialogues between humans. A pause may indicate that the current speaker has completed their turn, or is unable to do so, thus giving the next speaker an opportunity to take the floor. Research in conversation analysis has indicated that a silence of approximately one second is an acceptable length of silence in conversation and that after such a period of silence speakers will begin talking to end the silence (Jefferson, 1989).

Silence can also cause the participants to make inferences about why the silence has occurred. Consider the following example, taken from Levinson (1983, p. 350):

3.17
1 A: So I was wondering would you be in your office on Monday by any chance?
(2 second pause)
2 A: Probably not.
3 B: Hmm, yes.

In this example A asks B a question. There is a two second pause, following which A continues "probably not". The brief delay in B's response is sufficient to trigger the inference that B is responding negatively. As it turns out in this example, A has made the wrong inference, as B goes on to reply in the affirmative. Nevertheless, the example demonstrates a powerful structural property of silence in conversation in that it is often used in conjunction with responses that in some way do not fill the expectations of the preceding turn, such as disagreements, refusals or rejections. These response types, known as "dispreferred" responses, are discussed further below.

There has been little analysis of silence in spoken dialogue systems (see, however, Wooffitt (1991)). Silence that occurs following an utterance by the user may be due to the time required by the system to process and interpret the user's utterance. If access to information is required before the system can respond, this can also result in a lengthy silence. However, as Wooffitt points out, users may make inferences as to why the silence has occurred, for example, that there was something wrong in their input. This can lead the user to attempt to correct their previous utterance or to provide additional information. However, it is precisely this sort of utterance that could cause trouble for a dialogue system, as the utterance is likely to contain items that are beyond the vocabulary and grammar of the system. Moreover, such corrections are often uttered using more exaggerated articulation – a slower rate, an increase in loudness and a rise in overall pitch – that will be difficult for the speech recognition component to process (Hirschberg et al., 1999).

Summary

This section has introduced some of the key characteristics of dialogue that are relevant to developers of spoken dialogue systems. The words, phrases and utterances of a dialogue have to be interpreted in context – both in terms of discourse phenomena, such as reference using pronouns, and in terms of how meanings are conveyed and understood. Furthermore, dialogue is a joint activity and many of the processes observable in dialogue, such as turn-taking and conversational grounding, make sense only in terms of a collaborative model. Finally, the content of dialogue – the utterances spoken by the participants – displays a number of properties that are important to consider. On the one hand, utterances are different in many ways from the well-formed sentences of written texts. On the other hand, there is additional information, conveyed in the prosodic features accompanying the utterances, in nonverbal behaviours and even in silences, that is a crucial part of the data of dialogue.

Modelling Dialogue: Structure and Process

Dialogue can be examined from two different perspectives. The first involves an analysis of transcripts of dialogues, usually with a dialogue-coding scheme, to discover regularly occurring patterns and structures. This approach is useful in order to gain an understanding of the basic elements of dialogue and to support the labelling of dialogues for corpus annotation. The second approach examines dialogue processes from the perspective of those participating in the dialogue, looking at issues such as how utterances are interpreted in context and what factors are involved in deciding what to say at a particular point in the dialogue. This approach is useful to support the computational modelling of dialogue agents. The following sections examine these two perspectives in greater detail.

Dialogue Structure

Since the 1960s, linguists working in the areas of discourse analysis, sociolinguistics and language acquisition have been developing models of dialogue structure that show how dialogues can be segmented into smaller functional units. Most research has focussed on the smallest unit, the dialogue act, in which a speaker performs a particular speech action, such as making a request. At the next level of analysis there has also been considerable research into sequences of dialogue acts, for example, a question followed by an answer, or more elaborate structures involving embedded sequences such as clarification requests and responses. There has been less research into larger units of analysis, although some attention has been devoted to ways in which topics are introduced, changed and reintroduced, and there has also been extensive research in sociolinguistic and ethnographic studies on speech events such as story-telling, interviews and conversations.

The general assumption behind much of this structural analysis of dialogue is that the units are hierarchically organised. Sinclair and Coulthard (1975), who developed a framework for the analysis of classroom talk between teachers and pupils, proposed a five-level hierarchy. Carletta et al. (1997), who developed a coding scheme for dialogues in the Map Task (see below), proposed a similar set of categories. Similar hierarchies have been described by Grosz and Sidner (1986) and Dahlbäck and Jönsson (1998). These units of dialogue will be discussed in the following sections, beginning with dialogue acts.

Dialogue Acts

The dialogue act can be considered to be the smallest unit of analysis in dialogue. A dialogue act describes the action performed by a speaker in an utterance. The term "dialogue act" was first introduced by Bunt (1979) and has since been used widely in dialogue analysis. Various other terms have been used, such as speech act (Searle, 1975), communicative act (Allwood, 1976), conversational move (Sinclair and Coulthard, 1975; Carletta et al., 1997) and dialogue move (Cooper et al., 1999). One of the first schemes for coding utterance functions was devised by Bales (1950) for the analysis of small group discussions. Several schemes were devised in the 1970s and 1980s to code utterance functions in various types of interaction, such as parent–child discourse, classroom interaction, therapeutic talk and job interviews. More recently, in the 1990s, new schemes were developed that have been used to tag utterances in large collections of dialogues to support automated analysis.

Dialogue act taxonomies differ in the types of dialogue activity they have been designed to describe, including casual conversation (Jurafsky et al., 1997), classroom discourse (Sinclair and Coulthard, 1975), collaborative scheduling (Alexandersson et al., 1998) and direction following (Carletta et al., 1997). Traum (2000) compares eight different schemes, showing how there are major differences in the distribution of act types across the various domains, schemes and corpora. There are also differences in level of detail, as some schemes have a small number of high-level categories while others try to capture finer distinctions, for example, with several subcategories of higher level acts such as "statement". In some cases the categories are grouped into hierarchies and levels so that the appropriate level of detail can be used as required. Two schemes will be briefly described – DAMSL and the coding scheme for the HCRC Map Task corpus.

DAMSL

DAMSL (Dialogue Act Markup in Several Layers) is a system for annotating dialogues (Allen and Core, 1997). The scheme was developed under the Discourse Resource Initiative, a group of researchers from several dialogue projects worldwide, primarily to provide a standard for the coding of task-oriented dialogues involving two agents collaborating to solve a problem. Utterances are tagged according to four main categories:

1. *Communicative Status.* Whether the utterance is intelligible and whether it was successfully completed.
2. *Information Level.* A characterisation of the semantic content of the utterance, in terms of whether it advances the task, discusses the problem-solving process, addresses the communication process or does not fall neatly into any category.
3. *Forward Looking Function.* How the current utterance affects the subsequent dialogue. For example, as the result of an utterance, is the speaker now committed to certain beliefs or to performing certain future actions?
4. *Backward Looking Function.* How the current utterance relates to the previous discourse.

Table 3.1 shows a list of forward-looking acts from the DAMSL scheme. These categories describe the functions of utterances mainly in terms of the speaker's intentions and the speaker's and hearer's obligations. Where there are a number of distinctions within an act, a decision tree is provided to assist the annotation process.

Requests for action (Influencing-addressee-future-action) obligate the hearer to either perform the action or at least acknowledge the request. However, the subcategory "Open-Options" does not oblige the hearer to respond. In the following example, the first utterance is an Open-Option (OO). B does not need to address it and can respond coherently with utterance 2.

tion, therapeutic talk and job interviews. More recently, in the 1990s, new schemes were developed that have been used to tag utterances in large collections of dialogues to support automated analysis.

Dialogue act taxonomies differ in the types of dialogue activity they have been designed to describe, including casual conversation (Jurafsky et al., 1997), classroom discourse (Sinclair and Coulthard, 1975), collaborative scheduling (Alexandersson et al., 1998) and direction following (Carletta et al., 1997). Traum (2000) compares eight different schemes, showing how there are major differences in the distribution of act types across the various domains, schemes and corpora. There are also differences in level of detail, as some schemes have a small number of high-level categories while others try to capture finer distinctions, for example, with several subcategories of higher level acts such as "statement". In some cases the categories are grouped into hierarchies and levels so that the appropriate level of detail can be used as required. Two schemes will be briefly described – DAMSL and the coding scheme for the HCRC Map Task corpus.

DAMSL

DAMSL (Dialogue Act Markup in Several Layers) is a system for annotating dialogues (Allen and Core, 1997). The scheme was developed under the Discourse Resource Initiative, a group of researchers from several dialogue projects worldwide, primarily to provide a standard for the coding of task-oriented dialogues involving two agents collaborating to solve a problem. Utterances are tagged according to four main categories:

1. *Communicative Status.* Whether the utterance is intelligible and whether it was successfully completed.
2. *Information Level.* A characterisation of the semantic content of the utterance, in terms of whether it advances the task, discusses the problem-solving process, addresses the communication process or does not fall neatly into any category.
3. *Forward Looking Function.* How the current utterance affects the subsequent dialogue. For example, as the result of an utterance, is the speaker now committed to certain beliefs or to performing certain future actions?
4. *Backward Looking Function.* How the current utterance relates to the previous discourse.

Table 3.1 shows a list of forward-looking acts from the DAMSL scheme. These categories describe the functions of utterances mainly in terms of the speaker's intentions and the speaker's and hearer's obligations. Where there are a number of distinctions within an act, a decision tree is provided to assist the annotation process.

Requests for action (Influencing-addressee-future-action) obligate the hearer to either perform the action or at least acknowledge the request. However, the subcategory "Open-Options" does not oblige the hearer to respond. In the following example, the first utterance is an Open-Option (OO). B does not need to address it and can respond coherently with utterance 2.

Table 3.1. DAMSL: Forward-looking dialogue acts

Statement	A claim made by the speaker or an answer to a question.
Assert	The speaker is trying to change the beliefs of the addressee.
Reassert	The speaker thinks the claim has already been made.
Other-statement	Other categories of statement.
Influencing-addressee-future-action	Request other than to perform an action.
Open-option	Suggests an action without obligating the hearer.
Action-directive	Obligates hearer to perform the action, or to communicate a refusal or an inability to perform the action.
Info-request	Questions and requests for information.
Committing-speaker-future-action	Commits the speaker to some future action.
Offer	Conditional on the hearer's agreement.
Commit	Not conditional on the hearer's agreement, e.g., promise.
Conventional	Conventional conversational acts.
Opening	Greetings, phrases used to start an interaction.
Closing	Saying goodbye, phrases used to close a dialogue.
Explicit-performative	Performing an action by virtue of saying the utterance, e.g., "I apologise" as way of doing an apology.
Exclamation	An exclamation, e.g., "Ouch".
Other-forward-function	Any other forward-looking function.

3.18

1	Open-Option	A:	There is an engine in Elmira.
2	Action-dir	B:	Let's take the engine from Bath.

However, in the following example, utterance 1 is an Action-directive and B has to address this request, by adopting it, explicitly rejecting it or offering an alternative:

3.19

1	Action-dir	A:	Let's use the engine in Elmira.
2	Reject(utt1)	B:	No.
3	Action-dir	B:	Let's take the engine from Bath.

Backward Looking Functions (shown in Table 3.2) indicate how the current utterance relates to the previous discourse. An utterance can answer, accept, reject or try to correct some previous utterance or utterances. Four dimensions are involved: agreement, signalling understanding, answering and relations to preceding discourse in terms of informational content. Backward-looking acts are coded in terms of the type of act as well as the previous elements of discourse that the acts relate to.

There are several subcategories within many of the backward-looking acts, and decision trees are provided to assist annotation. The following example lists a number of responses to an offer within the "agreement" category:

Table 3.2. DAMSL: Backward-looking acts

Agreement	The second participant's response to a previous proposal, request or claim.
Accept	Accepts all of the proposal, request or claim.
Accept-part	Accepts the proposal, request or claim in part.
Maybe	Defers a definite answer.
Reject-part	Rejects part of the proposal, request or claim.
Reject	Reject all of the proposal, request or claim.
Hold	Leaves the proposal, request or claim open, e.g., counterproposal, request for further information.
Understanding	Actions taken by the speakers to ensure they understand. each other as the dialogue proceeds.
Signal-non-understanding	Explicitly indicate a problem in understanding the utterance.
Signal-understanding	Explicitly signal understanding.
Acknowledge	Indicate that the previous utterance was understood without necessarily signalling acceptance.
Repeat-rephrase	Repeat or rephrase the previous utterance to signal understanding.
Completion	Signal understanding by completing the speaker's utterance.
Correct-misspeaking	Correction of the previous speaker's utterance.
Answer	Compliance with an Info-Request.
Information-relations	How the content of the current utterance relates to the content of the previous utterance.

3.20

1	A:	Would you like the book and its review?	
2a	B:	Yes please.	Accept(1)
2b	B:	I'd like the book.	Accept-part(1)
2c	B:	I'll have to think about it.	Maybe(1)
(intended literally rather than a polite reject)			
2d	B:	I don't need the review.	Reject-part(1)
2e	B:	No thanks.	Reject(1)

The coding scheme permits multiple tags for utterances that achieve several effects simultaneously. Recently a dialogue annotation tool, called "dat", has been made available to support the annotation task (DAMSL Dialog Annotation, 2001).

A more elaborate version of the DAMSL coding scheme, Switchboard Shallow-Discourse Function Annotation (SWBD-DAMSL), has been used to code the Switchboard corpus (Jurafsky et al., 1997). Switchboard is a collection of about 2430 spontaneous conversations between 543 speakers in which the subjects were allowed to converse freely about a given topic for 5 to 10 minutes. Shallow discourse function refers to a level of coding that captures basic information about how one type of utterance is responded to by another type, in contrast to deep discourse function that encodes conversations in terms of the goals and plans of the participants. A set of about 60 tags was used to code 1155 of the Switchboard conversations. Many of the tags provided more subtle distinctions of the original DAMSL categories. For example, there were a number of subcategories of the DAMSL tag "answer", such as "yes answer", "no answer", "affirmative non-yes answer" and several others.

The HCRC Map Task Coding Scheme

This coding scheme was developed in the Human Communicator Research Centre (HCRC) Map Task project at the University of Edinburgh. The HCRC Map Task was a project conducted at the University of Edinburgh designed to provide a corpus of dialogues that could serve as the basis for a variety of empirical studies of dialogue. In the Map Task two speakers sat opposite one another and each had a map which the other could not see. One speaker had a route marked on her map, while the other speaker had no route on her map. The task for the speaker without a map was to mark out a route on her map based on instructions from the speaker whose map had a route. The two maps were not identical, and the speakers had to discover how the maps differed. Dialogues were coded using a scheme that included three levels of dialogue unit – conversational moves, corresponding to dialogue acts, and conversational game and transaction, to be described below (Carletta et al., 1997).

Conversational moves, described here as Initiations and Responses, are similar to the forward- and backward-looking functions in the DAMSL scheme (see Table 3.3). The moves are described in terms of their communicative functions.

The labels for conversational moves are fairly self-explanatory and there is a decision tree to determine move categories. The following example illustrates the "check" move (Carletta et al., 1997, p. 17):

Table 3.3. Coding of conversational moves in the HCRC Map Task project

Initiations	
Instruct	Speaker tells hearer to carry out an action.
Explain	Speaker states information that has not been elicited by hearer.
Check	Speaker requests hearer to confirm some information that speaker is not sure about.
Align	Speaker checks hearer's attention or agreement.
Query-yn	A question that requires a "yes" or "no" answer.
Query-w	Any query not covered by the other categories, e.g., a "who" or "what" question.
Responses	
Acknowledge	Indicates that the previous utterance was heard and understood.
Reply-y	Reply to a Query-yn query indicating "yes".
Reply-n	Reply to a Query-yn query indicating "no".
Reply-w	Reply to a Query-w query.
Clarify	Reply with information over and above what was asked.

3.21
1 G: Ehm, curve round slightly to your right.
2 F: To my right? (check)
3 G: Yes.
4 F: As I look at it? (check)

Comparing the two schemes, the HCRC scheme uses more surface-level definitions while in DAMSL the definitions are more intention-based. Surface-level definitions are easier to operate. However, interpreting dialogue moves involves recognising intentions, which may not be directly expressed in the surface form. In the DAMSL coding manual it is noted that it is often difficult to determine the actions that the speaker intended to perform with an utterance as well as the effect that the utterance might have on the subsequent dialogue. For example, the effect might differ from what the speaker initially intended. To deal with these problems annotators are allowed to look ahead in the dialogue to help determine the effect that an utterance has on the dialogue. However, while this strategy is useful to support annotation of utterances in a dialogue, it is not a resource that is available to the participants. For this reason, representations describing the processes involved in dialogue, such as the participants' information states, need to be constructed dynamically with provision for change as the dialogue proceeds.

Exchanges and Games

A dialogue is not just a sequence of dialogue acts by different speakers. Rather the dialogue acts relate to each other in a number of ways, as suggested already in the notion of forward- and backward-looking functions (or initiations and responses). A number of schemes encoding these relations has been proposed, including adjacency pairs, exchanges, discourse segments and conversational games.

Adjacency Pairs

The term "adjacency pair" was coined by conversation analysts to describe pairs of utterances that belong together, such as greeting–greeting, question–answer and invitation–acceptance/refusal (Schegloff and Sacks, 1973). Instead of initiations and responses, the utterances were called first- and second-pair parts. Other related units are "presequences" and "inserted sequences". A presequence is an utterance that prepares the way for a subsequent utterance. For example, "Are you doing anything tonight?" could be interpreted not only as a question but as a preliminary to an invitation. Inserted sequences interrupt the normal flow of the dialogue, usually to clarify something that is unclear.

Many responses in adjacency pairs have alternatives, as in the pair invitation–acceptance/refusal. However, these alternatives are not equivalent, as the following examples from Atkinson and Drew (1979, p. 58) illustrate:

3.22
1 A: Why don't you come up and see me sometimes?
2 B: I would like to.

3.23
1 A: uh, If you'd care to come and visit a little while this morning I'll give you a cup of coffee.
2 B: heh, Well that's awfully sweet of you.
3 B: I don't think I can make it this morning.
4 B: Uhm, I'm running an ad in the paper and – and, uh, I have to stay near the phone.

The response to the invitation in Example 3.22 is an acceptance. It occurs promptly and is simple in form. In contrast, the response in 3.23 is marked in several ways – it is delayed, an appreciation is proffered and the refusal is explained. The difference between these responses has been described in terms of the notion of "preference". An acceptance to an invitation is preferred, in the sense that it is the default or unmarked response. A refusal is dispreferred, in that it is usually marked in some way, as shown in the example.

More generally, there seem to be three classes of response to an utterance: the preferred response, a set of dispreferred responses and a response (such as a silence) in which no mention is made of either alternative (see below for further discussion of preferred and dispreferred responses). Thus it is possible to explain inferences that arise when a person apparently fails to respond to a prior utterance, whether deliberately or not (Bilmes, 1988). For example, a person can either accept or reject an invitation. However, if the response seems to be neither an acceptance nor a rejection, then that response, which may be a silence, gives rise to the inference that a rejection is intended. The following schema illustrates this process, where X represents a preferred response, Y a dispreferred response and N represents no mention of either X or Y:

3.24

Invitation	
X	Accept.
Y	Refuse.
N	No mention of X or Y.
Inference	If N, then assume refusal.

This structure applies to other sequences such as requests and accusations. The following schema illustrates an accusation sequence:

3.25

Accusation	
X	Denial.
Y	Acceptance.
N	No mention of X or Y.
Inference	If N, assume acceptance of accusation.

This sequence applies in everyday conversation but has also been adopted in some legal systems where a failure to respond in court is taken as admission of guilt.

A number of similar dispreferred responses has been documented in studies by conversation analysts. As far as spoken dialogue systems are concerned, this difference in the form and organisation of responses to initiations is important for more advanced systems, either to support their interpretation of a user's response or to enable them to generate an appropriately polite response that is a dispreferred category.

Exchanges

Exchanges were proposed as a minimal unit of interaction in classroom discourse and similarly structured dialogues such as doctor–patient interviews. In the original formulation, exchanges were seen to consist regularly of three moves: an initiation, a response and a feedback (or follow-up) move, as in the following example (Sinclair and Coulthard 1975, p. 68):

3.26

1	Teacher:	What makes a road slippery?	Initiation
2	Pupil:	You might have snow or ice on it.	Response
3	Teacher:	Yes, snow, ice.	Follow-up
4	Teacher:	Anything else make a road slippery?	Initiation
5	Pupil:	Erm oil.	Response
6	Teacher:	Oil makes a road slippery when it's mixed with water, doesn't it?	Follow-up

In this example there are six moves which can be grouped into two exchanges comprising utterances 1–3 and 4–6. Furthermore, the teacher's turn at lines 3 and 4 consists of two moves, one of which is a follow-up within the first exchange, while the other initiates the second exchange. Thus the basic unit is not the turn but the move, as speakers can accomplish more than one move within a turn and turns may be divided across exchanges.

Exchange structure has been studied widely and a number of additions to the original scheme were proposed, including a move Response/Initiation that functions simultaneously as a response and an initiation, as in the following example:

3.27

1	A:	Where's the typewriter?	Initiation
2	B:	Is it in the cupboard?	Response/Initiation
3	A:	No.	Response

Various other combinations of moves were proposed within the same general framework (see, e.g., Stubbs (1983)). However, a major problem with the framework was that it could not be easily applied to the analysis of more open-ended dialogue, such as casual conversation. One difficulty is that many utterances appear to be multifunctional and cannot be assigned to a single category. For example, in a casual conversation most utterances might take the form Response/Initiation, as the participants not only respond to what the other has said but also say something that sets up expectations for a further response. Nevertheless, the framework has been influential in studies of dialogue and its spirit has been reflected in similar schemes such as discourse segments and conversational games.

Discourse Segments

In an influential paper on the structure of discourse, Grosz and Sidner (1986) proposed a model in which discourse is represented in terms of three components: linguistic structure, intentional structure and attentional state. The attentional state is a model of the discourse entities – people, objects, events, and so on – that are salient at each point in the discourse. This model changes dynamically as the discourse proceeds. The intentional structure describes the underlying purpose of a discourse (or dialogue) in terms of the intentions of the person who initiates the discourse. A discourse consists of a number of discourse segments, each of which also has a purpose, representing the intentions of the participants at the level of the segment. For example, a dialogue agent might intend that another agent believe some fact. Two relations are specified for discourse purposes. The first, called "dominance" in the theory, represents a hierarchical relationship between discourse purposes. Taking the example of a flight reservation, the purpose "flight reservation" dominates the purposes "ask for destination", "ask for departure date", and so on, because these smaller subtasks have to be completed before the overall task is accomplished. The second relation, called "satisfaction-precedence", represents the ordering of the discourse purposes. For example, in the flight reservation scenario, finding the destination and departure information would, in most circumstances, precede determining the price.

Discourse segments are similar to exchanges and adjacency pairs, but here the analysis focuses on the intentions (or purposes) behind the moves and a discourse segment is defined as a sequence of moves that share a common purpose.

Conversational Games

Conversational game theory was used initially in the HCRC Map Task project to describe sequences of the moves listed in Table 3.3 (Kowtko et al., 1993). Examples of games are instructing, obtaining information and getting the other dialogue participant to carry out an action. As with discourse segments, games are defined on the basis of the intention behind the game. A game begins with an initiating move and continues until the goal of the initiation has been fulfilled. There may also be nesting of games, when a subgame is initiated whose purpose contributes to the overall goal of the current game, for example, to request clarification about some crucial missing information. Games can also be broken off, for example, following a misunderstanding that is not cleared up. Games are similar in structure to exchanges. For example, a Query game might consist of the moves Query-W, Reply-W and Acknowledge.

Transactions

Transactions are considered to be the highest level unit in a dialogue. A dialogue may consist of one transaction but, more typically, dialogues consist of a number of transactions that generally correspond to subtasks or subtopics. In the Map Task dialogues the participants tended to break the task of describing the route into manageable segments that could be dealt with sequentially (Carletta et al., 1997).

Because participants did not always proceed along the route in an orderly fashion, categories were required to describe different transaction types:

- *Normal* – describes a subtask.
- *Review* – reviews parts of the task that have been completed.
- *Overview* – provides a context for an upcoming subtask.
- *Irrelevant* – discussion not relevant to the task.

Coding transactions involves marking the start of a transaction in the dialogue transcripts, assigning the transaction to one of the four types and, except for IRRELEVANT transactions, indicating the start and end point of the relevant route section using numbered crosses on a copy of the route giver's map.

Using Dialogue Coding Schemes

Dialogue coding schemes can be used for a variety of purposes:

1. For annotation and analysis of units of dialogue, occurrences and structures.

Dialogue act taxonomies can be used to provide labels for the annotation of utterances in dialogue corpora. Previously, work on dialogue tended to use isolated examples, which were either constructed or real. Given the capability of storing large corpora of dialogues on computers, it has been possible to develop more realistic analyses of dialogue structures using coded dialogue samples.

2. To support the design of dialogue systems.

Dialogue acts can be used to specify the moves in a spoken dialogue system. The sequencing of these acts can be specified in advance using a dialogue grammar or flow chart so that there is a finite set of ways in which a dialogue can proceed. Dialogue acts and sequences can also be used to encode part of the information state of a dialogue agent thus enabling the agent to interpret the meaning and intent of the other agent's utterances as well as assisting the agent in deciding what to say or do next.

3. To support machine learning of dialogue acts and sequences.

The main purpose of the label set in the Switchboard coding scheme SWBD-DAMSL was to support the labelling of Switchboard conversations for training stochastic dialogue grammars that would enable the construction of better Language Models for Automatic Speech Recognition of the Switchboard conversations. These dialogue grammars could also be used to train dialogue managers (see also Chapter 13).

4. Theoretical analysis.

Finally, dialogue act taxonomies have been used as a basis for the analysis of the pragmatic meanings of utterances in more theoretically oriented studies of dialogue (e.g., Bunt (1979); Allwood (1976)).

Dialogue Processes

The structures that have been described in the previous section are evidence of regularly occurring patterns within dialogues. Given such structures it should be possible to construct dialogue grammars that would specify well-formed sequences of utterances in the same way that grammars for sentences specify well-formed sequences of words and phrases. Such grammars could be used to develop spoken dialogue systems in which the system performs a particular dialogue act and then uses the grammar to predict the set of possible next acts that the user might perform according to the grammar. Given a particular dialogue act from the user, the system could use the grammar to determine which act it should perform next.

However, although dialogue grammars can be used in this way to model simple dialogues between the system and user, it is clear that such grammars are inadequate in several respects:

1. *Descriptive adequacy.* It is difficult to apply existing schemes to the analysis of transcripts of naturally occurring dialogue.

2. *Combinatorial explosion.* Even if more elaborate schemes are devised, given the range of alternative responses to a given dialogue act, and then the number of possible responses to these responses, and so on, the combinations to be described by the grammar quickly grow to unmanageable proportions.

3. *Implication of ill-formed dialogue sequences.* The specification of well-formed sequences implies that sequences that do not conform to these structures are ill-formed. However, it is difficult to find actual examples of ill-formed dialogues. Instead, when an expected response does not occur, participants tend to try to make sense of this nonoccurrence. For example, in Dialogue 3.17, the occurrence of a silence did not lead to a judgement of an ill-formed sequence but rather gave rise to inferences concerning the reasons for the silence.

More generally, dialogue grammars fail to model the processes involved when participants engage in dialogue. The fact that one dialogue act follows another does not explain why such a sequence occurs. The following sections outline some approaches to the analysis of dialogue processes that seek to explain the nature of dialogue behaviour from the perspective of the dialogue participants.

Dialogue as Social Behaviour

The approach to dialogue adopted by the Conversation Analysts was to identify regularly occurring patterns in conversation and to model the procedures used by participants when engaging in conversation. The methodology and theoretical orientation of Conversation Analysis (CA) derives from a branch of sociology known as Ethnomethodology, which sought to study the techniques used by members of society to interpret and act within their social worlds. As far as conversation was concerned, the aim was to identify the orderliness that is displayed in everyday conversation and the methods employed by participants to maintain this orderliness. The essential findings were that conversations are organised locally and that participants engage in ongoing interactional work to create and maintain orderly conversations.

These points can be illustrated with reference to the structural unit of the adjacency pair. Adjacency pairs could be used as a basis for dialogue grammars.

However, in the CA approach, adjacency pairs were viewed as providing a normative framework within which participants could interpret and make sense of the ongoing dialogue. After producing a first-pair part the speaker will examine the next utterance for the anticipated response. On producing this next utterance, the recipient of the first-pair part displays his (or her) understanding of that first-pair part. Thus each turn provides an ongoing opportunity for participants to monitor the conversation and display their understanding. In this sense the structures are locally managed and emergent rather than predetermined, as in a dialogue grammar. Moreover, adjacency pairs are characterised in terms of "conditional relevance" rather than in terms of well-formed and ill-formed sequences. What this means is that if a second-pair part within an adjacency pair fails to occur, then it is seen to be noticeably absent. On the recognition of this absence the speaker of the first-pair part will make inferences to explain the absence. For example, if there is a failure to respond to a question, it might be inferred that the addressee wishes to avoid answering the question. Observations of large samples of conversational data by conversation analysts demonstrated this process across a range of different adjacency pair types. Similarly, in the case of conversational breakdowns, it was shown that participants worked together to locate the problem and to apply an appropriate repair. More generally, the CA approach was able to explain how the patterns that could be identified in transcripts of conversations reflected an orderliness that was oriented to by the participants and that was achieved collaboratively on a turn-by-turn basis.

Dialogue as the Achievement of Mutual Understanding

Clark (1996) views dialogue as a joint action in which the participants work to achieve mutual understanding. The structures that are observable in conversation emerge from this joint activity. As Clark (1996, p. 319) argues:

> Conversations look planned and goal-oriented only in retrospect. In reality, they are created opportunistically piece by piece as the participants negotiate joint purposes and then fulfil them. . . . In the opportunistic view, the hierarchical structure of conversation is an emergent property. It appears because of principles that govern any successful joint action.

Achieving mutual understanding involves the process of grounding described earlier. Information contributed by participants has to be mutually acknowledged as having entered the "common ground" (Clark and Schaefer, 1989; Traum, 1994). This can be achieved in a number of ways, including assertions of understanding, producing a relevant next turn and displaying understanding. Grounding is a dynamic process that gives rise to structural units called Common Ground Units (CGUs) (Nakatani and Traum, 1999). CGUs represent all the linguistic material involved in achieving grounding of an initial presentation of information. They can be complex, containing repair and clarification sequences. There can also be overlapping CGUs, in which an utterance may simultaneously ground an open CGU while also initiating a new CGU. Finally, they can also be discontinuous and revisited later in the dialogue, for example, for further confirmation or to initiate a repair.

Table 3.4. Analysis using exchanges and games

3.28	Exchange	Game
1 S: Where are you travelling to?	Initiation	WH-query
2 U: London.	Response	WH-reply
3 S: You want to go to London?	Initiation? Feedback?	Check
4 U: Yes.	?	Clarify

Viewing dialogue structure in terms of CGUs provides some advantages over other structural units such as the exchange or game:

- CGUs provide an explanation of dialogue structure in terms of the motivation of participants to achieve mutual understanding.
- They are a dynamically evolving structure as opposed to the more static structures proposed in exchanges and games.
- As such, they are more descriptively adequate than other structures when describing sequences of dialogue.

For example, the sequence, shown in Table 3.4, would be difficult to analyse in terms of exchange structures or games. Analysing this sequence using exchange structures runs into problems at utterance 3. This utterance could be coded as an initiation, but this would not reflect how utterances 3 and 4 relate to the exchange in 1–2. If coded as feedback, then it would not be clear how to code utterance 4, as feedback utterances do not require a response, yet clearly utterance 4 responds to 3. Coding according to conversational game theory codes 1 and 2 as a WH-question game followed by an embedded Checking game, but fails to capture the intuition that the point of the Checking game is to ground the information requested in utterance 1. Indeed, the WH-question game could be said to be incomplete until the information is grounded – a process that could continue over several further turns.

In this way, CGUs are more adequate from a descriptive viewpoint as well as providing an explanation for the structures that emerge.

Dialogue as Rational Action

Dialogue has also been viewed in terms of a theory of rational action, in which intelligent dialogue behaviour is seen as a special case of general rational behaviour (Cohen, 1994; Sadek and de Mori, 1997). Much of the early work within this tradition in the 1980s focussed on modelling dialogue in terms of planning. Utterances were viewed as actions that are planned in order to achieve a goal. The goal may be some desired physical state, such as having a drink of beer in a bar. In this case, an utterance that functions as a request for a beer is incorporated into a plan that also involves physical actions, such as the customer handing over some money and the barman giving the beer. Interpreting utterances involved recognising the intention behind an utterance and matching this intention with some part of a plan that might achieve a particular goal (Allen, [illegible]). A cooperative system would adopt the user's goal, anticipate any obstacles to the plan, and produce a response that would promote the completion of the goal.

In more recent work, plans are not predetermined schemas of action but are derived deductively from rationality principles. While agents normally have the goal of behaving cooperatively in dialogue, an agent does not necessarily have to adopt another agent's goals, if there is good reason not to. For example, an agent should not supply information that is confidential or assist in actions that it knows to be illegal or harmful. In other words, an agent has to attempt to achieve a rational balance between its own mental attitudes and those of other agents and between these mental attitudes and desired plans of action. Thus, in this approach, dialogue structure emerges dynamically as a consequence of principles of rational cooperative interaction, and the processes of dialogue can be explained in terms of the plans, goals and intentions of the agents involved in the dialogue.

The view that dialogue is a special case of rational behaviour brings several advantages. Given that dialogue involves a joint commitment to mutual understanding, there is a motivation for agents to make their intentions clear through confirmations, clarifications, repairs and elaborations (Cohen, 1994). Although these behaviours are included in other approaches, there is no theoretical motivation for their inclusion. The theory also accounts for different contexts of interaction and explains why an agent might provide more information than is required by the other agent's query. For example, if a user asks for an address, the system might also provide a telephone number, if one is available. However, this additional information should not be implemented as an automatically generated response schema but rather as something to be determined within a particular context of interaction on the basis of the rationality principles. Finally, the theory provides a basis for more advanced dialogues, for example, those involving negotiation rather than simple information retrieval, where various types of cooperative and corrective responses may be required.

Summary

These three accounts of dialogue processes are similar in many ways. In particular, they emphasise the view that the structures to be found in dialogue evolve as a result of the processes that participants employ when they engage in dialogue. The structures are not used by the participants to determine whether a dialogue is well-formed or not, but are used as a normative framework in which inferences can be made. Furthermore, dialogue structure is viewed not as predetermined but as dynamically evolving and as a process of rational action, in which the participants work together to achieve their goals, including the conversational goal of understanding one another.

Representing Information States in Dialogue

Given the processes described above in which dialogue participants monitor the ongoing dialogue to assist their interpretation of what is being said and to support their decisions as to what to say next, it is clear that one aspect of dialogue processes is the knowledge that participants bring to bear when taking part in a dialogue. There are different types of knowledge that are involved. Some of this knowledge is static, such as knowledge about the domain being discussed in the dialogue and knowledge of general conversational principles. Other knowledge is dynamic, such as knowledge about what has been said so far in the dialogue, what information is part of the common ground, and what actions can be taken next. This dynamic knowledge has been referred to as the "information state". Using this information state a participant can decide what to say next, how to interpret what the other participant has said, and how modify the current information state based on utterances in the dialogue.

In the simplest case the information state may consist of a list of questions to ask and the answers that have been received. Thus an agent providing a travel service might have questions such as:

3.29
1 Where are you travelling from?
2 Where are you travelling to?
3 What date do you wish to travel?
4 What time do you wish to leave?

These might be represented as a set of attributes and values, representing the agent's information at a particular point in the dialogue, for example,

Origin: Belfast.

Destination: London.

Date: Unknown.

Time: Unknown.

There would also be a mechanism for updating the information on the basis of what is said in the dialogue, for example, changing values from "unknown" to one of the values required by the question, or updating the status of a value from known to confirmed. On the basis of such an information state, the agent could determine that it already knew the values for origin and departure, but not for date and time. Using this information, the agent is able to decide that it does not need to ask about origin and destination, but does need to ask the questions to elicit values for date and time. If values have not yet been confirmed, the agent would also have to decide which values to confirm and when. Representations such as this have been used extensively to determine the actions of dialogue agents in simple spoken dialogue systems.

Information state has generally been used to describe more complex representations that capture a wide range of information about the dialogue and its participants. This information may include the following:

- *Mental states.* The beliefs, desires and intentions of the participants.
- *Information status.* Whether the information is private to one participant, shared (part of the common ground) or semi-shared (introduced into the dialogue but not yet grounded).

- *Obligations.* The obligations and commitments of the participants, for example, to respond to a question.
- *Dialogue information.* Who has the current turn? What is the topic? What was the previous utterance? Which dialogue act did the previous utterance perform?
- *Plan.* The overall goals of the participants.
- *Agenda.* The immediate goals and obligations of the participants.

Information states may represent the dialogue information of one or both of the participants, or may even represent an external view of the dialogue. When annotating transcripts of a dialogue it is possible to represent the information states of each participant as well as representing the dialogue from the perspective of an external observer. Modelling each participant's information state is particularly useful when there are misunderstandings between the participants, as these can be identified from the discrepant information states. However, when modelling the processes of a dialogue participant over the course of a dialogue, it is only possible to represent information states from a single perspective, i.e., that of the dialogue agent (or system). Given that the agent is not omniscient, it can have knowledge of its own beliefs, desires and intentions, but it can only infer the beliefs, desires and intentions of the other participant. Thus while agent A might believe on the basis of what B has said that B wishes to travel to X, it could be the case that B had actually expressed a desire to travel to Y, and that A had misrecognised or misunderstood Y as X. For this reason establishing information as common ground is an important part of attempting to ensure that the participants' information states concur.

This brief discussion of dialogue information states has illustrated another aspect of the dynamic processes that are to be modelled in dialogue systems. Much of current research in information state theory is concerned with complex representation formalisms and mechanisms for updating information states. Some of this work will be described in greater detail in Chapter 13.

Summary

This chapter has been concerned with the characteristics of dialogue and with its structures and processes. The following are the main issues covered in this chapter:

- Dialogue is an example of extended discourse. Words, phrases and utterances are produced and interpreted in the context of discourse and not in isolation.
- Dialogue is a purposeful activity. People engage in dialogue for a purpose and their meanings and actions have to be interpreted in the context of these purposes.
- Dialogue is a collaborative activity in which two (or more) people work together to make the dialogue work. Examples of collaborative activity include ensuring smooth turn-taking and achieving mutual understanding.

- The spoken language typical of naturally occurring dialogue can differ considerably in form from the language of written texts. This has implications for the modules involved in processing naturally occurring speech for speech recognition and speech understanding.
- Dialogue is structured hierarchically. These structures can be used to support the design of dialogue systems and to train systems using machine learning techniques.

In addition to these structures, there are processes that participants in dialogue use to engage in dialogue. It is important to consider these processes in order to explain the structures that evolve in a dialogue.

Further Reading

Pragmatics and Discourse Analysis

There are numerous books on pragmatics and discourse analysis. Levinson (1983) is a standard text on pragmatics, covering topics such as speech act theory, conversational implicature and presupposition, with an excellent chapter on Conversation Analysis (CA). Stubbs (1983) and Brown and Yule (1983) provide comprehensive reviews of different aspects of discourse analysis. Schiffrin (1994) discusses a number of different approaches to discourse analysis, accompanied by detailed analysis of examples. See also Schiffrin et al. (2001) for an edited collection of a wide-ranging set of papers on discourse analysis. For an account of dialogue as coordinated action, in which it is argued that language use involves both individual and social processes, see Clark (1996).

Computational Pragmatics and Dialogue

Bunt and Black (2000a) is a recent collection of papers on computational pragmatics, including a useful introductory overview of the field (Bunt and Black, 2000b). Jurafsky (2004) discusses computational pragmatics, looking in particular at different computational models for the interpretation of indirect speech acts. Webber (2001) reviews computational perspectives on dialogue and discourse, while Leech and Weisser (2003) discuss pragmatics in relation to dialogue. The American Association for Artificial Intelligence (AAAI) Fall 1999 Symposium on Psychological Models of Communication in Collaborative Systems provides a useful set of papers on interactive systems from a psychological perspective (http://www.cs.umd.edu/users/traum/PM/papers.html). Button et al. (1995) argue against the possibility of the conversational computer, based on a critique of a wide range of views in modern cognitive science and the philosophy of mind.

Exercises

Transcribe one of the sample dialogues that you encountered in the exercises in Chapters 1 and 2. Annotate the dialogue using the following schemes described in this chapter:

1. DAMSL.
2. The HCRC Map Task coding scheme.

To what extent are these schemes useful for annotating the dialogue? Do they account for all the important aspects of the interaction?