19 DIALOGUE AND CONVERSATIONAL AGENTS

C: I want you to tell me the names of the fellows on the St. Louis team.
A: I'm telling you. Who's on first, What's on second, I Don't Know is on third.
C: You know the fellows' names?
A: Yes.
C: Well, then, who's playing first?
A: Yes.
C: I mean the fellow's name on first.
A: Who.
C: The guy on first base.
A: Who is on first.
C: Well what are you askin' *me* for?
A: I'm not asking you – I'm telling you. Who is on first.

Who's on First – Bud Abbott and Lou Costello's version of an old burlesque standard.

The literature of the fantastic abounds in inanimate objects magically endowed with sentience and the gift of speech. From Ovid's statue of Pygmalion to Mary Shelley's Frankenstein, Cao Xue Qin's Divine Luminescent Stone-in-Waiting in the Court of Sunset Glow to Snow White's mirror, there is something deeply touching about creating something and then having a chat with it. Legend has it that after finishing his sculpture of *Moses*, Michelangelo thought it so lifelike that he tapped it on the knee and commanded it to speak. Perhaps this shouldn't be surprising. Language itself has always been the mark of humanity and sentience, and **conversation** or **dialogue** is the most fundamental and specially privileged arena of language.

CONVERSATION
DIALOGUE

It is certainly the first kind of language we learn as children, and for most of us, it is the kind of language we most commonly indulge in, whether we are ordering curry for lunch or buying postage stamps, participating in business meetings or talking with our families, booking airline flights or complaining about the weather.

This chapter introduces the fundamental structures and algorithms in **conversational agents**, programs which communicate with users in natural language in order to book airline flights, answer questions, or act as a telephone interface to email. Many of these issues are also relevant for **business meeting summarization** systems and other spoken language understanding systems which must transcribe and summarize structured conversations like meetings. Section 19.1 begins by introducing some issues that make conversation different from other kinds of discourse, introducing the important ideas of **turn-taking**, **grounding**, and **implicature**. Section 19.2 introduces the **speech act** or **dialogue act**, and Section 19.3 gives two different algorithms for automatic speech act interpretation. Section 19.4 describes how structure and coherence in dialogue differ from the discourse structure and coherence we saw in Chapter 18. Finally, Section 19.5 shows how each of these issues must be addressed in choosing an architecture for a dialogue manager as part of a conversational agent.

19.1 What Makes Dialogue Different?

Much about dialogue is similar to other kinds of discourse like the text monologues of Chapter 18. Dialogues exhibit anaphora and discourse structure and coherence, although with some slight changes from monologue. For example when resolving an anaphor in dialogue it's important to look at what the other speaker said. In the following fragment from the air travel conversation in Figure 19.1 (to be discussed below), realizing that the pronoun *they* refers to *non-stop flights* in *C*'s utterance requires looking at *A*'s previous utterance:

A_4: Right. There's three non-stops today.
C_5: What are they?

Dialogue does differ from written monologue in deeper ways, however. The next few subsections highlight some of these differences.

Turns and Utterances

One difference between monologue and dialogue is that dialogue is characterized by **turn-taking**. Speaker A says something, then speaker B, then speaker A, and so on. Figure 19.1 shows a sample dialogue broken up into labeled turns; we've chosen this human-human dialogue because it concerns travel planning, a domain that is the focus of much recent human-machine dialogue research.

TURN-TAKING

C_1: . . . I need to travel in May.
A_1: And, what day in May did you want to travel?
C_2: OK uh I need to be there for a meeting that's from the 12th to the 15th.
A_2: And you're flying into what city?
C_3: Seattle.
A_3: And what time would you like to leave Pittsburgh?
C_4: Uh hmm I don't think there's many options for non-stop.
A_4: Right. There's three non-stops today.
C_5: What are they?
A_5: The first one departs PGH at 10:00am arrives Seattle at 12:05 their time. The second flight departs PGH at 5:55pm, arrives Seattle at 8pm. And the last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.
C_6: OK I'll take the 5ish flight on the night before on the 11th.
A_6: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.
C_7: OK.

Figure 19.1 A fragment from a telephone conversation between a client (C) and a travel agent (A).

How do speakers know when is the proper time to contribute their turn? Consider the timing of the utterances in conversations like Figure 19.1. First, notice that this dialogue has no noticeable overlap. That is, the beginning of each speaker's turn follows the end of the previous speaker's turn (overlap would have been indicated by surrounding it with the # symbol). The actual amount of overlapped speech in American English conversation seems to be quite small; Levinson (1983) suggests the amount is less than 5% in general, and probably less for certain kinds of dialogue like the task-oriented dialogue in Figure 19.1. If speakers aren't overlapping, perhaps they are waiting a while after the other speaker? This is also very rare. The amount of time between turns is quite small, generally less than a few hundred milliseconds even in multi-party discourse. In fact, it may take more than this few hundred milliseconds for the next speaker to plan the motor routines for producing their utterance, which means that speakers begin motor planning for their next utterance before the previous speaker has finished. For this to be possible, natural conversation must be set up in such a way that (most of the time) people can quickly figure out **who** should talk next, and exactly **when** they should talk. This kind of turn-taking behavior is generally studied in the field of **Conversation Analysis (CA)**. In a key conversation-analytic paper, Sacks et al. (1974) argued that turn-taking behavior, at least in American English, is governed by a set of turn-taking rules. These rules apply at a **transition-relevance place**, or **TRP**; places where the structure of the language allows speaker shift to occur. Here is a simplified version of the turn-taking rules, grouped into a single three-part rule; see Sacks et al. (1974) for the complete rules:

CONVERSATION ANALYSIS

(19.1) **Turn-taking Rule.** At each TRP of each turn:

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

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

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

There are a number of important implications of rule (19.1) for dialogue modeling. First, subrule (19.1a) implies that there are some utterances by which the speaker specifically selects who the next speaker will be. The most obvious of these are questions, in which the speaker selects another speaker to answer the question. Two-part structures like QUESTION-ANSWER are called **adjacency pairs** (Schegloff, 1968); other adjacency pairs include GREETING followed by GREETING, COMPLIMENT followed by DOWNPLAYER, REQUEST followed by GRANT. We will see that these pairs and the dialogue expectations they set up will play an important role in dialogue modeling.

ADJACENCY PAIRS

Subrule (19.1a) also has an implication for the interpretation of silence. While silence can occur after any turn, silence which follows the first part of an adjacency pair-part is **significant silence**. For example Levinson (1983) notes the following example from Atkinson and Drew (1979); pause lengths are marked in parentheses (in seconds):

SIGNIFICANT SILENCE

(19.2) A: Is there something bothering you or not?
(1.0)
A: Yes or no?
(1.5)
A: Eh?
B: No.

Since A has just asked B a question, the silence is interpreted as a refusal to respond, or perhaps a **dispreferred** response (a response, like saying "no" to a request, which is stigmatized). By contrast, silence in other places, for example a lapse after a speaker finishes a turn, is not generally interpretable in this way. These facts are relevant for user interface design in spoken dialogue systems; users are disturbed by the pauses in dialogue systems caused by slow speech recognizers (Yankelovich et al., 1995).

DISPREFERRED

Another implication of (19.1) is that transitions between speakers don't occur just anywhere; the **transition-relevance places** where they tend to occur are generally at **utterance** boundaries. This brings us to the next difference between spoken dialogue and textual monologue (of course dialogue can be written and monologue spoken; but most current applications of dialogue involve speech): the spoken **utterance** versus the written **sentence**. Recall from Chapter 9 that utterances differ from written sentences in a number of ways. They tend to be shorter, are more likely to be single clauses, the subjects are usually pronouns rather than full lexical noun phrases, and they include filled pauses, repairs, and restarts.

UTTERANCE

One very important difference not discussed in Chapter 9 is that while written sentences and paragraphs are relatively easy to automatically segment from each other, utterances and turns are quite complex to segment. Utterance boundary detection is important since many computational dialogue models are based on extracting an utterance as a primitive unit. The segmentation problem is difficult because a single utterance may be spread over several turns, or a single turn may include several utterances. For example in the following fragment of a dialogue between a travel agent and a client, the agent's utterance stretches over three turns:

(19.3) A: Yeah yeah the um let me see here we've got you on American flight nine thirty eight
C: Yep.
A: leaving on the twentieth of June out of Orange County John Wayne Airport at seven thirty p.m.
C: Seven thirty.
A: and into uh San Francisco at eight fifty seven.

By contrast, the example below has three utterances in one turn:

(19.4) A: Three two three and seven five one. OK and then does he know there is a nonstop that goes from Dulles to San Francisco? Instead of connection through St. Louis.

Algorithms for utterance segmentation are based on many boundary **cues** such as:

CUE WORDS

- **cue words:** Cue (or "clue") words like *well*, *and*, *so*, etc., tend to occur at the beginnings and ends of utterances (Reichman, 1985; Hirschberg and Litman, 1993).
- *N*-**gram word or POS sequences:** Specific word or POS sequences often indicate boundaries. *N*-gram grammars can be trained on a training set labeled with special utterance-boundary tags, and then a decoder can find the most likely utterance boundaries in a unlabeled test set (Mast et al., 1996; Meteer and Iyer, 1996; Stolcke and Shriberg, 1996a; Heeman and Allen, 1999).
- **prosody:** Prosodic features like pitch, accent, phrase-final lengthening and pause duration play a role in utterance/turn segmentation, as discussed in Chapter 4, although the relationship between utterances and prosodic units like the **intonation unit** (Du Bois et al., 1983) or **intonation phrase** (Pierrehumbert, 1980; Beckman and Pierrehumbert, 1986) is complicated (Ladd, 1996; Ford and Thompson, 1996; Ford et al., 1996, inter alia) .

INTONATION PHRASE

The relationship between turns and utterances seems to be more one-to-one in human-machine dialogue than the human-human dialogues discussed above. Probably this is because the simplicity of current systems causes people to use simpler utterances and turns. Thus while computational tasks like **meeting summarization** require solving quite difficult segmentation problems, segmentation may be easier for conversational agents.

Grounding

Another important characteristic of dialogue that distinguishes it from monologue is that it is a collective act performed by the speaker and the hearer. One implication of this collectiveness is that, unlike in monologue, the speaker and hearer must constantly establish **common ground** (Stalnaker, 1978), the set of things that are mutually believed by both speakers. The need to

COMMON GROUND

GROUND
ACKNOWLEDGE

achieve common ground means that the hearer must **ground** or **acknowledge** the speaker's utterances, or else make it clear that there was a problem in reaching common ground. For example, consider the role of the word *mm-hmm* in the following fragment of a conversation between a travel agent and a client:

A: . . . returning on U.S. flight one one one eight.
C: Mm hmm

CONTINUER
BACKCHANNEL

The word *mm-hmm* here is a **continuer**, also often called a **backchannel** or an **acknowledgement token**. A continuer is a short utterance which acknowledges the previous utterance in some way, often cueing the other speaker to continue talking (Jefferson, 1984; Schegloff, 1982; Yngve, 1970). By letting the speaker know that the utterance has "reached" the addressee, a continuer/backchannel thus helps the speaker and hearer achieve common ground. Continuers are just one of the ways that the hearer can indicate that she believes she understands what the speaker meant. Clark and Schaefer (1989) discuss five main types of methods, ordered from weakest to strongest:

1. **Continued attention:** B shows she is continuing to attend and therefore remains satisfied with A's presentation.
2. **Relevant next contribution:** B starts in on the next relevant contribution.
3. **Acknowledgement:** B nods or says a continuer like *uh-huh*, *yeah*, or the like, or an **assessment** like *that's great*.
4. **Demonstration:** B demonstrates all or part of what she has understood A to mean, for example by paraphrasing or **reformulating** A's utterance, or by **collaboratively completing** A's utterance.
5. **Display:** B displays verbatim all or part of A's presentation.

The following excerpt from our sample conversation shows a display of understanding by A's repetition of *on the 11th*:

C_6: OK I'll take the 5ish flight on the night before on the 11th.
A_6: On the 11th?

Such repeats or reformulations are often done in the form of questions like A_6; we return to this issue on page 739.

Not all of Clark and Shaefer's methods are available for telephone-based conversational agents. Without eye-gaze as a visual indicator of attention, for example, **continued attention** isn't an option. In fact Stifelman et al. (1993) and Yankelovich et al. (1995) point out that users of speech-based interfaces are often confused when the system doesn't give them an explicit acknowledgement signal after processing the user's utterances.

REQUEST FOR REPAIR

In addition to these acknowledgement acts, a hearer can indicate that there were problems in understanding the previous utterance, for example by issuing a **request for repair** like the following Switchboard example:

A: Why is that?
B: Huh?
A: Why is that?

Conversational Implicature

The final important property of conversation is the way the interpretation of an utterance relies on more than just the literal meaning of the sentences. Consider the client's response C_2 from the sample conversation above, repeated here:

A_1: And, what day in May did you want to travel?
C_2: OK uh I need to be there for a meeting that's from the 12th to the 15th.

Notice that the client does not in fact answer the question. The client merely states that he has a meeting at a certain time. The semantics for this sentence produced by a semantic interpreter will simply mention this meeting. What is it that licenses the agent to infer that the client is mentioning this meeting so as to inform the agent of the travel dates?

Now consider another utterance from the sample conversation, this one by the agent:

A_4: ...There's three non-stops today.

Now this statement would still be true if there were seven non-stops today, since if there are seven of something, there are by definition also three. But what the agent means here is that there are three **and not more than three** non-stops today. How is the client to infer that the agent means **only three** non-stops?

IMPLICATURE

These two cases have something in common; in both cases the speaker seems to expect the hearer to draw certain inferences; in other words, the speaker is communicating more information than seems to be present in the uttered words. These kind of examples were pointed out by Grice (1975, 1978) as part of his theory of **conversational implicature**. **Implicature** means a particular class of licensed inferences. Grice proposed that what enables hearers to draw these inferences is that conversation is guided by a set of **maxims**, general heuristics which play a guiding role in the interpretation of conversational utterances. He proposed the following four maxims: MAXIMS

- **Maxim of Quantity:** Be exactly as informative as is required: QUANTITY
  1. Make your contribution as informative as is required (for the current purposes of the exchange).
  2. Do not make your contribution more informative than is required.
- **Maxim of Quality:** Try to make your contribution one that is true: QUALITY
  1. Do not say what you believe to be false.
  2. Do not say that for which you lack adequate evidence.
- **Maxim of Relevance:** Be relevant. RELEVANCE
- **Maxim of Manner:** Be perspicuous: MANNER
  1. Avoid obscurity of expression.
  2. Avoid ambiguity.
  3. Be brief (avoid unnecessary prolixity).
  4. Be orderly.

It is the Maxim of Quantity (specifically Quantity 1) that allows the hearer to know that *three non-stops* did not mean *seven non-stops*. This is because the hearer assumes the speaker is following the maxims, and thus if the speaker meant seven non-stops she would have said seven non-stops ("as informative as is required"). The Maxim of Relevance is what allows the agent to know that the client wants to travel by the 12th. The agent assumes the client is following the maxims, and hence would only have mentioned the meeting if it was relevant at this point in the dialogue. The most natural inference that would make the meeting relevant is the inference that the client meant the agent to understand that his departure time was before the meeting time.

These three properties of conversation (**turn-taking**, **grounding**, and **implicature**) will play an important role in the discussion of dialogue acts, dialogue structure, and dialogue managers in the next sections.

19.2 DIALOGUE ACTS

An important insight about conversation, due to Austin (1962), is that an utterance in a dialogue is a kind of **action** being performed by the speaker.

PERFORMATIVE This is particularly clear in **performative** sentences like the following:

(19.5) I name this ship the *Titanic*.

(19.6) I second that motion.

(19.7) I bet you five dollars it will snow tomorrow.

When uttered by the proper authority, for example, (19.5) has the effect of changing the state of the world (causing the ship to have the name *Titanic*) just as any action can change the state of the world. Verbs like *name* or *second* which perform this kind of action are called performative verbs, and SPEECH ACTS Austin called these kinds of actions **speech acts**. What makes Austin's work so far-reaching is that speech acts are not confined to this small class of performative verbs. Austin's claim is that the utterance of any sentence in a real speech situation constitutes three kinds of acts:

- **locutionary act:** the utterance of a sentence with a particular meaning.
- **illocutionary act:** the act of asking, answering, promising, etc., in uttering a sentence.
- **perlocutionary act:** the (often intentional) production of certain effects upon the feelings, thoughts, or actions of the addressee in uttering a sentence.

For example, Austin explains that the utterance of example (19.8) might ILLOCUTIONARY FORCE have the **illocutionary force** of protesting and the perlocutionary effect of stopping the addressee from doing something, or annoying the addressee.

(19.8) You can't do that.

The term **speech act** is generally used to describe illocutionary acts rather than either of the other two levels. Searle (1975b), in modifying a taxonomy of Austin's, suggests that all speech acts can be classified into one of five major classes:

- **Assertives:** committing the speaker to something's being the case (*suggesting, putting forward, swearing, boasting, concluding*).
- **Directives:** attempts by the speaker to get the addressee to do something (*asking, ordering, requesting, inviting, advising, begging*).
- **Commissives:** committing the speaker to some future course of action (*promising, planning, vowing, betting, opposing*).
- **Expressives:** expressing the psychological state of the speaker about a state of affairs *thanking, apologizing, welcoming, deploring*.
- **Declarations:** bringing about a different state of the world via the utterance (including many of the performative examples above; *I resign, You're fired*.)

While speech acts provide a useful characterization of one kind of pragmatic force, more recent work, especially in building dialogue systems, has significantly expanded this core notion, modeling more kinds of conversational functions that an utterance can play. The resulting enriched acts are called **dialogue acts** (Bunt, 1994) or **conversational moves** (Power, 1979; Carletta et al., 1997). A recent ongoing effort to develop dialogue act tagging scheme is the DAMSL (Dialogue Act Markup in Several Layers) architecture (Allen and Core, 1997; Walker et al., 1996; Carletta et al., 1997; Core et al., 1999), which codes various levels of dialogue information about utterances. Two of these levels, the **forward looking function** and the **backward looking function**, are extensions of speech acts which draw on notions of dialogue structure like the adjacency pairs mentioned earlier as well as notions of grounding and repair. For example, the forward looking function of an utterance corresponds to something like the Searle/Austin speech act, although the DAMSL tag set is hierarchical, and is focused somewhat on the kind of dialogue acts that tend to occur in task-oriented dialogue:

DIALOGUE ACT

MOVES

STATEMENT	a claim made by the speaker
INFO-REQUEST	a question by the speaker
CHECK	a question for confirming information (see below)
INFLUENCE-ON-ADDRESSEE	(=Searle's directives)
OPEN-OPTION	a weak suggestion or listing of options
ACTION-DIRECTIVE	an actual command
INFLUENCE-ON-SPEAKER	(=Austin's commissives)
OFFER	speaker offers to do something, (subject to confirmation)
COMMIT	speaker is committed to doing something
CONVENTIONAL	other
OPENING	greetings
CLOSING	farewells
THANKING	thanking and responding to thanks

The backward looking function of DAMSL focuses on the relationship of an utterance to previous utterances by the other speaker. These include accepting and rejecting proposals (since DAMSL is focused on task-oriented dialogue), as well as grounding and repair acts discussed above.

AGREEMENT	speaker's response to previous proposal
ACCEPT	accepting the proposal
ACCEPT-PART	accepting some part of the proposal
MAYBE	neither accepting nor rejecting the proposal
REJECT-PART	rejecting some part of the proposal
REJECT	rejecting the proposal
HOLD	putting off response, usually via subdialogue
ANSWER	answering a question
UNDERSTANDING	whether speaker understood previous
SIGNAL-NON-UNDER.	speaker didn't understand
SIGNAL-UNDER.	speaker did understand
ACK	demonstrated via continuer or assessment
REPEAT-REPHRASE	demonstrated via repetition or reformulation
COMPLETION	demonstrated via collaborative completion

Figure 19.2 shows a labeling of our sample conversation using versions of the DAMSL Forward and Backward tags.

19.3 Automatic Interpretation of Dialogue Acts

The previous section introduced dialogue acts and other activities that utterances can perform. This section turns to the problem of identifying or interpreting these acts. That is, how do we decide whether a given input is a QUESTION, a STATEMENT, a SUGGEST (directive), or an ACKNOWLEDGEMENT?

At first glance, this problem looks simple. We saw in Chapter 9 that yes-no-questions in English have **aux-inversion**, statements have declarative syntax (no aux-inversion), and commands have imperative syntax (sentences with no syntactic subject), as in example (19.9):

(19.9) YES-NO-QUESTION Will breakfast be served on USAir 1557?
STATEMENT I don't care about lunch
COMMAND Show me flights from Milwaukee to Orlando on Thursday night.

It seems from (19.9) that the surface syntax of the input ought to tell us what illocutionary act it is. Alas, as is clear from Abbott and Costello's famous *Who's on First* routine at the beginning of the chapter, things are not so simple. The mapping between surface form and illocutionary act is not obvious or even one-to-one.

[assert]	C_1:	...I need to travel in May.
[info-req,ack]	A_1:	And, what day in May did you want to travel?
[assert, answer]	C_2:	OK uh I need to be there for a meeting that's from the 12th to the 15th.
[info-req,ack]	A_2:	And you're flying into what city?
[assert,answer]	C_3:	Seattle.
[info-req,ack]	A_3:	And what time would you like to leave Pittsburgh?
[check,hold]	C_4:	Uh hmm I don't think there's many options for non-stop.
[accept,ack]	A_4:	Right.
[assert]		There's three non-stops today.
[info-req]	C_5:	What are they?
[assert, open-option]	A_5:	The first one departs PGH at 10:00am arrives Seattle at 12:05 their time. The second flight departs PGH at 5:55pm, arrives Seattle at 8pm. And the last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.
[accept,ack]	C_6:	OK I'll take the 5ish flight on the night before on the 11th.
[check,ack]	A_6:	On the 11th?
[assert,ack]		OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.
[ack]	C_7:	OK.

Figure 19.2 A potential DAMSL labeling of the conversation fragment in Figure 19.1.

For example, the following utterance spoken to an ATIS system looks like a YES-NO-QUESTION meaning something like *Are you capable of giving me a list of... ?*:

(19.10) Can you give me a list of the flights from Atlanta to Boston?

In fact, however, this person was not interested in whether the system was *capable* of giving a list; this utterance was actually a polite form of a DIRECTIVE or a REQUEST, meaning something more like *Please give me a list of....* Thus what looks on the surface like a QUESTION can really be a REQUEST.

Similarly, what looks on the surface like a STATEMENT can really be a QUESTION. A very common kind of question, called a CHECK question (Carletta et al., 1997; Labov and Fanshel, 1977), is used to ask the other participant to confirm something that this other participant has privileged knowledge about. These CHECKs are questions, but they have declarative surface form, as the boldfaced utterance in the following snippet from another travel agent conversation:

A	OPEN-OPTION	I was wanting to make some arrangements for a trip that I'm going to be taking uh to LA uh beginning of the week after next.
B	HOLD	OK uh let me pull up your profile and I'll be right with you here. [pause]
B	CHECK	**And you said you wanted to travel next week?**
A	ACCEPT	Uh yes.

INDIRECT SPEECH ACTS

Utterances which use a surface statement to ask a question, or a surface question to issue a request, are called **indirect speech acts**.How can a surface yes-no-question like *Can you give me a list of the flights from Atlanta to Boston?* be mapped into the correct illocutionary act REQUEST? Solutions to this problem lie along a continuum of idiomaticity. At one end of the continuum is the *idiom* approach, which assumes that a sentence structure like *Can you give me a list?* or *Can you pass the salt?* is ambiguous between a literal meaning as a YES-NO-QUESTION and an idiomatic meaning as a request. The grammar of English would simply list REQUEST as one meaning of *Can you X*. One problem with this approach is that there are many ways to make an indirect request, each of which has slightly different surface grammatical structure (see below). The grammar would have to store the REQUEST meaning in many different places. Furthermore, the idiom approach doesn't make use of the fact that there are semantic generalizations about what makes something a legitimate indirect request.

The alternative end of the continuum is the *inferential* approach, first proposed by Gordon and Lakoff (1971) and taken up by Searle (1975a). Their intuition was that a sentence like *Can you give me a list of flights from Atlanta?* is unambiguous, meaning only *Do you have the ability to give me a list of flights from Atlanta?* The directive speech act *Please give me a list of flights from Atlanta* is **inferred** by the hearer.

INFERRED

The next two sections will introduce two models of dialogue act interpretation: an inferential model called the **plan inference** model, and an idiom-based model called the **cue** model.

Plan-Inferential Interpretation of Dialogue Acts

The plan-inference approach to dialogue act interpretation was first proposed by Gordon and Lakoff (1971) and Searle (1975a) when they noticed that there was a structure to what kind of things a speaker could do to make an indirect request. In particular, they noticed that a speaker could mention or question various quite specific properties of the desired activity to make an indirect request; here is a partial list with examples from the ATIS corpus:

1. The speaker can question the hearer's ability to perform the activity
   - Can you give me a list of the flights from Atlanta to Boston?
   - Could you tell me if Delta has a hub in Boston?
   - Would you be able to, uh, put me on a flight with Delta?
2. The speaker can mention speaker's wish or desire about the activity
   - I want to fly from Boston to San Francisco.
   - I would like to stop somewhere else in between.
   - I'm looking for one way flights from Tampa to Saint Louis.
   - I need that for Tuesday.
   - I wonder if there are any flights from Boston to Dallas.
3. The speaker can mention the hearer's doing the action
   - Would you please repeat that information?
   - Will you tell me the departure time and arrival time on this American flight?
4. The speaker can question the speaker's having permission to receive results of the action
   - May I get a lunch on flight U A two one instead of breakfast?
   - Could I have a listing of flights leaving Boston?

Based on this realization, Searle (1975a, p. 73) proposed that the hearer's chain of reasoning upon hearing *Can you give me a list of the flights from Atlanta to Boston?* might be something like the following (modified for our ATIS example):

1. X has asked me a question about whether I have the ability to give a list of flights.
2. I assume that X is being cooperative in the conversation (in the Gricean sense) and that his utterance therefore has some aim.
3. X knows I have the ability to give such a list, and there is no alternative reason why X should have a purely theoretical interest in my list-giving ability.

4. Therefore X's utterance probably has some ulterior illocutionary point. What can it be?
5. A preparatory condition for a directive is that the hearer have the ability to perform the directed action.
6. Therefore X has asked me a question about my preparedness for the action of giving X a list of flights.
7. Furthermore, X and I are in a conversational situation in which giving lists of flights is a common and expected activity.
8. Therefore, in the absence of any other plausible illocutionary act, X is probably requesting me to give him a list of flights.

The inferential approach has a number of advantages. First, it explains why *Can you give me a list of flights from Boston?* is a reasonable way of making an indirect request and *Boston is in New England* is not: the former mentions a precondition for the desired activity, and there is a reasonable inferential chain from the precondition to the activity itself. The inferential approach has been modeled by Allen, Cohen, and Perrault and their colleagues in a number of influential papers on what have been called **BDI** (belief, desire, and intention) models (Allen, 1995). The earliest papers, such as Cohen and Perrault (1979), offered an AI planning model for how speech acts are *generated*. One agent, seeking to find out some information, could use standard planning techniques to come up with the plan of asking the hearer to tell the speaker the information. Perrault and Allen (1980) and Allen and Perrault (1980) also applied this BDI approach to *comprehension*, specifically the comprehension of indirect speech effects, essentially cashing out Searle's (1975) promissory note in a computational formalism.

BDI

We'll begin by summarizing Perrault and Allen's formal definitions of belief and desire in the predicate calculus. We'll represent "S believes the proposition P" as the two-place predicate $B(S,P)$. Reasoning about belief is done with a number of axiom schemas inspired by Hintikka (1969) (such as $B(A,P) \wedge B(A,Q) \Rightarrow B(A,P \wedge Q)$; see Perrault and Allen (1980) for details). Knowledge is defined as "true belief"; *S knows that P* will be represented as $KNOW(S,P)$, defined as follows:

$$\text{KNOW}(S,P) \equiv P \wedge B(S,P)$$

In addition to *knowing that*, we need to define *knowing whether*. *S knows whether* (KNOWIF) a proposition P is true if S KNOWs that P or S KNOWs that $\neg P$:

$$\text{KNOWIF}(S,P) \equiv \text{KNOW}(S,P) \vee \text{KNOW}(S,\neg P)$$

The theory of desire relies on the predicate WANT. If an agent S wants P to be true, we say $WANT(S,P)$, or $W(S,P)$ for short. P can be a state or the execution of some action. Thus if ACT is the name of an action, $W(S,\text{ACT}(H))$ means that S wants H to do ACT. The logic of WANT relies on its own set of axiom schemas just like the logic of belief.

ACTION SCHEMA

The BDI models also require an axiomatization of actions and planning; the simplest of these is based on a set of **action schemas** similar to the AI planning model STRIPS (Fikes and Nilsson, 1971). Each action schema has a set of parameters with *constraints* about the type of each variable, and three parts:

- *Preconditions:* Conditions that must already be true in order to successfully perform the action.
- *Effects:* Conditions that become true as a result of successfully performing the action.
- *Body:* A set of partially ordered goal states that must be achieved in performing the action.

In the travel domain, for example, the action of agent A booking flight $F1$ for client C might have the following simplified definition:

BOOK-FLIGHT(A,C,F):

Constraints: Agent(A) ∧ Flight(F) ∧ Client(C)

Precondition: Know(A,departure-date(F)) ∧ Know(A,departure-time(F)) ∧ Know(A,origin-city(F)) ∧ Know(A,destination-city(F)) ∧ Know(A,flight-type(F)) ∧ Has-Seats(F) ∧ W(C,(BOOK(A,C,F))) ∧ . . .

Effect: Flight-Booked(A,C,F)

Body: Make-Reservation(A,F,C)

Cohen and Perrault (1979) and Perrault and Allen (1980) use this kind of action specification for speech acts. For example here is Perrault and Allen's definition for three speech acts relevant to indirect requests. INFORM is the speech act of informing the hearer of some proposition (the Austin/Searle *Assertive*, or DAMSL STATEMENT). The definition of INFORM is based on Grice's (1957) idea that a speaker informs the hearer of something merely by causing the hearer to believe that the speaker wants them to know something:

INFORM(S,H,P):

Constraints: Speaker(S) ∧ Hearer(H) ∧ Proposition(P)
Precondition: Know(S,P) ∧ W(S, INFORM(S, H, P))
Effect: Know(H,P)
Body: B(H,W(S,Know(H,P)))

INFORMIF is the act used to inform the hearer whether a proposition is true or not; like INFORM, the speaker INFORMIFs the hearer by causing the hearer to believe the speaker wants them to KNOWIF something:

INFORMIF(S,H,P):

Constraints: Speaker(S) ∧ Hearer(H) ∧ Proposition(P)
Precondition: KnowIf(S, P) ∧ W(S, INFORMIF(S, H, P))
Effect: KnowIf(H, P)
Body: B(H, W(S, KnowIf(H, P)))

REQUEST is the directive speech act for requesting the hearer to perform some action:

REQUEST(S,H,ACT):

Constraints: Speaker(S) ∧ Hearer(H) ∧ ACT(A) ∧ H is agent of ACT
Precondition: W(S,ACT(H))
Effect: W(H,ACT(H))
Body: B(H,W(S,ACT(H)))

Perrault and Allen's theory also requires what are called "surface-level acts". These correspond to the "literal meanings" of the imperative, interrogative, and declarative structures. For example the "surface-level" act S.REQUEST produces imperative utterances:

S.REQUEST (S, H, ACT):

Effect: B(H, W(S,ACT(H)))

The effects of S.REQUEST match the body of a regular REQUEST, since this is the default or standard way of doing a request (but not the only way). This "default" or "literal" meaning is the start of the hearer's inference chain. The hearer will be given an input which indicates that the speaker is requesting the hearer to inform the speaker whether the hearer is capable of giving the speaker a list:

S.REQUEST(S,H,InformIf(H,S,CanDo(H,Give(H,S,LIST))))

The hearer must figure out that the speaker is actually making a request:

REQUEST(H,S,Give(H,S,LIST))

The inference chain from the request-to-inform-if-cando to the request-to-give is based on a chain of *plausible inference*, based on heuristics called **plan inference (PI)** rules. We will use the following subset of the rules that Perrault and Allen (1980) propose:

PLAN INFERENCE

- **(PI.AE) Action-Effect Rule:** For all agents S and H, if Y is an effect of action X and if H believes that S wants X to be done, then it is plausible that H believes that S wants Y to obtain.
- **(PI.PA) Precondition-Action Rule:** For all agents S and H, if X is a precondition of action Y and if H believes S wants X to obtain, then it is plausible that H believes that S wants Y to be done.
- **(PI.BA) Body-Action Rule:** For all agents S and H, if X is part of the body of Y and if H believes that S wants X done, then it is plausible that H believes that S wants Y done.
- **(PI.KP) Know-Desire Rule:** For all agents S and H, if H believes S wants to KNOWIF(P), then H believes S wants P to be true:

  $$B(H,W(S,\text{KNOWIF}(S,P))) \overset{\text{plausible}}{\Longrightarrow} B(H,W(S,P))$$

- **(EI.1) Extended Inference Rule:** if $B(H,W(S,X)) \overset{\text{plausible}}{\Longrightarrow} B(H,W(S,Y))$ is a PI rule, then

  $$B(H,W(S,B(H,(W(S,X))))) \overset{\text{plausible}}{\Longrightarrow} B(H,W(S,B(H,W(S,Y))))$$

  is a PI rule. (i.e., you can prefix $B(H,W(S))$ to any plan inference rule).

Let's see how to use these rules to interpret the indirect speech act in *Can you give me a list of flights from Atlanta?* Step 0 in the table below shows the speaker's initial speech act, which the hearer initially interprets literally as a question. Step 1 then uses Plan Inference rule *Action-Effect*, which suggests that if the speaker asked for something (in this case information), they probably want it. Step 2 again uses the *Action-Effect* rule, here suggesting that if the Speaker wants an INFORMIF, and KNOWIF is an effect of INFORMIF, then the speaker probably also wants KNOWIF.

Rule	**Step**	**Result**
	0	S.REQUEST(S,H,InformIf(H,S,CanDo(H,Give(H,S,LIST))))
PI.AE	1	B(H,W(S,InformIf(H,S,CanDo(H,Give(H,S,LIST)))))
PI.AE/EI	2	B(H,W(S,KnowIf(H,S,CanDo(H,Give(H,S,LIST)))))
PI.KP/EI	*3*	*B(H,W(S,CanDo(H,Give(H,S,LIST))))*
PI.PA/EI	4	B(H,W(S,Give(H,S,LIST)))
PI.BA	5	REQUEST(H,S,Give(H,S,LIST))

Step 3 adds the crucial inference that people don't usually ask about things they aren't interested in; thus if the speaker asks whether something is true (in this case CanDo), the speaker probably wants it (CanDo) to be true. Step 4 makes use of the fact that CanDo(ACT) is a precondition for (ACT), making the inference that if the speaker wants a precondition (CanDo) for an action (Give), the speaker probably also wants the action (Give). Finally, step 5 relies on the definition of REQUEST to suggest that if the speaker wants someone to know that the speaker wants them to do something, then the speaker is probably REQUESTing them to do it.

In giving this summary of the plan-inference approach to indirect speech act comprehension, we have left out many details, including many necessary axioms, as well as mechanisms for deciding which inference rule to apply. The interested reader should consult Perrault and Allen (1980) and the other literature suggested at the end of the chapter.

Cue-based Interpretation of Dialogue Acts

The plan-inference approach to dialogue act comprehension is extremely powerful; by using rich knowledge structures and powerful planning techniques the algorithm is designed to address even subtle indirect uses of dialogue acts. The disadvantage of the plan-inference approach is that it is very time-consuming both in terms of human labor in development of the plan-inference heuristics, and in terms of system time in running these heuristics. In fact, by allowing all possible kinds of non-linguistic reasoning to play a part in discourse processing, a complete application of this approach is **AI-complete**. An AI-complete problem is one which cannot be truly solved without solving the entire problem of creating a complete artificial intelligence.

AI-COMPLETE

Thus for many applications, a less sophisticated but more efficient data-driven method may suffice. One such method is a variant of the *idiom* method discussed above. Recall that in the idiom approach, sentences like *Can you give me a list of flights from Atlanta?* have two literal meanings; one as a question and one as a request. This can be implemented in the grammar by listing sentence structures like *Can you X* with two meanings. The **cue-based** approach to dialogue act comprehension we develop in this section is based on this idiom intuition.

A number of researchers have used what might be called a cue-based approach to dialogue act interpretation, although not under that name. What characterizes a cue-based model is the use of different sources of knowledge (cues) for detecting a dialogue act, such as lexical, collocational, syntactic, prosodic, or conversational-structure cues. The models we will describe use (supervised) machine-learning algorithms, trained on a corpus of dialogues that is hand-labeled with dialogue acts for each utterance. Which cues are used depends on the individual system. Many systems rely on the fact that individual dialogue acts often have what Goodwin (1996) called a **microgrammar**; specific lexical, collocation, and prosodic features which are characteristic of them. These systems also rely on conversational structure. The dialogue-act interpretation system of Jurafsky et al. (1997), for example, relies on 3 sources of information:

MICROGRAMMAR

1. **Words and Collocations:** *Please* or *would you* is a good cue for a REQUEST, *are you* for YES-NO-QUESTIONs.
2. **Prosody:** Rising pitch is a good cue for a YES-NO-QUESTION. Loudness or stress can help distinguish the *yeah* that is an AGREEMENT from the *yeah* that is a BACKCHANNEL.
3. **Conversational Structure:** A *yeah* which follows a proposal is probably an AGREEMENT; a *yeah* which follows an INFORM is probably a BACKCHANNEL.

The previous section focused on how the plan-based approach figured out that a surface question had the illocutionary force of a REQUEST. In this section we'll look at a different kind of indirect request; the CHECK, examining the specific cues that the Jurafsky et al. (1997) system uses to solve this dialogue act identification problem. Recall that a CHECK is a subtype of question which requests the interlocutor to confirm some information; the information may have been mentioned explicitly in the preceding dialogue (as in the example below), or it may have been inferred from what the interlocutor said:

A	OPEN-OPTION	I was wanting to make some arrangements for a trip that I'm going to be taking uh to LA uh beginning of the week after next.
B	HOLD	OK uh let me pull up your profile and I'll be right with you here. [pause]
B	CHECK	**And you said you wanted to travel next week?**
A	ACCEPT	Uh yes.

Examples of possible realizations of CHECKs in English include:

1. As tag questions:

   (19.11) From the Trains corpus (Allen and Core, 1997)

   U **and it's gonna take us also an hour to load boxcars right?**
   S right

2. As declarative questions, usually with rising intonation (Quirk et al., 1985, p. 814)

   (19.12) From the Switchboard corpus (Godfrey et al., 1992)

   A and we have a powerful computer down at work.
   B Oh (laughter)
   B **so, you don't need a personal one (laughter)?**
   A No

3. As fragment questions (subsentential units; words, noun-phrases, clauses) (Weber, 1993)

   (19.13) From the Map Task corpus (Carletta et al., 1997)

   G Ehm, curve round slightly to your right.
   F **To my right?**
   G Yes.

Studies of checks have shown that, like the examples above, they are most often realized with declarative structure (i.e., no aux-inversion), they are most likely to have rising intonation (Shriberg et al., 1998), and they often have a following **question tag**, often *right*, (Quirk et al., 1985, 810-814), as in example (19.11) above. They also are often realized as "fragments" (subsentential words or phrases) with rising intonation (Weber, 1993). In Switchboard, the REFORMULATION subtype of CHECKs have a very specific microgrammar, with declarative word order, often *you* as subject (31% of the cases), often beginning with *so* (20%) or *oh*, and sometimes ending with *then*. Some examples:

Oh so you're from the Midwest too.
So you can steady it.
You really rough it then.

Many scholars, beginning with Nagata and Morimoto (1994), realized that much of the structure of these micrograms could be simply captured by training a separate word-*N*-gram grammar for each dialogue act (see e.g., Suhm and Waibel, 1994; Mast et al., 1996; Jurafsky et.al., 1997; Warnke et al., 1997; Reithinger and Klesen, 1997; Taylor et al., 1998). These systems create a separate mini-corpus from all the utterances which realize the same dialogue act, and then train a separate word-N-gram language model on each of these mini-corpora. Given an input utterance u consisting of a sequence of words W, they then choose the dialogue act d whose N-gram grammar assigns the highest likelihood to W:

$$d^* = \operatorname*{argmax}_d P(d|W) = \operatorname*{argmax}_d P(d)P(W|d) \tag{19.14}$$

This simple N-gram approach does indeed capture much of the micro-grammar; for example examination of the high-frequency bigram pairs in Switchboard REFORMULATIONS shows that the most common bigrams include good cues for REFORMULATIONS like *so you, sounds like, so you're, oh so, you mean, so they*, and *so it's*.

Prosodic models of dialogue act microgrammar rely on phonological features like pitch or accent, or their acoustic correlates like F0, duration, and energy discussed in Chapter 4 and Chapter 7. For example many studies have shown that capturing the rise in pitch at the end of YES-NO-QUESTIONS can be a useful cue for augmenting lexical cues (Sag and Liberman, 1975; Pierrehumbert, 1980; Waibel, 1988; Daly and Zue, 1992; Kompe et al., 1993; Taylor et al., 1998). Pierrehumbert (1980) also showed that declarative utterances (like STATEMENTS) have **final lowering**: a drop in F0 at the end of the utterance. One system which relied on these results, Shriberg et al. (1998), trained CART-style decision trees on simple acoustically-based prosodic features such as the slope of F0 at the end of the utterance, the average energy at different places in the utterance, and various duration measures. They found that these features were useful, for example, in distinguishing the four dialogue acts STATEMENT (S), YES-NO QUESTION (QY), DECLARATIVE-QUESTIONS like CHECKS (QD) and WH-QUESTIONS (QW). Figure 19.3 shows the decision tree which gives the posterior probability $P(d|f)$ of a dialogue act d type given sequence of acoustic features F. Each node in the tree shows four probabilities, one for each of the four dialogue acts in the order S, QY, QW, QD; the most likely of the four is shown as the label for the node. Via the Bayes rule, this probability can be used to compute the likelihood of the acoustic features given the dialogue act: $P(f|d)$.

FINAL LOWERING

A final important cue for dialogue act interpretation is conversational structure. One simple way to model conversational structure, drawing on the idea of adjacency pairs (Schegloff, 1968; Sacks et al., 1974) introduced above, is as a probabilistic sequence of dialogue acts. The identity of the

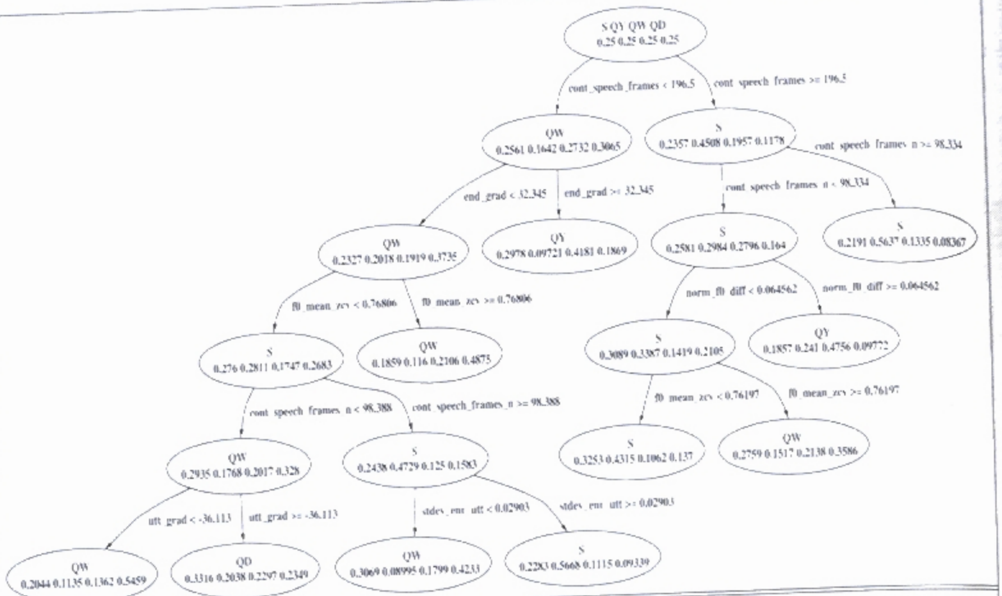

Figure 19.3 Decision tree for the classification of STATEMENT (S), YES-NO QUESTIONS (QY), WH-QUESTIONS (QW) and DECLARATIVE QUESTIONS (QD), after Shriberg et al. (1998). Note that the difference between S and QY toward the right of the tree is based on the feature `norm_f0_diff` (normalized difference between mean F0 of end and penultimate regions), while the difference between WQ and QD at the bottom left is based on `utt_grad`, which measures F0 slope across the whole utterance.

previous dialogue acts can then be used to help predict upcoming dialogue acts. Many studies have modeled dialogue act sequences as dialogue-act-*N*-grams (Nagata and Morimoto, 1994; Suhm and Waibel, 1994; Warnke et al., 1997; Chu-Carroll, 1998; Stolcke et al., 1998; Taylor et al., 1998), often as part of an HMM system for dialogue acts (Reithinger et al., 1996; Kita et al., 1996; Woszczyna and Waibel, 1994). For example Woszczyna and Waibel (1994) give the dialogue HMM shown in Figure 19.4 for a Verbmobil-like appointment scheduling task.

How does the dialogue act interpreter combine these different cues to find the most likely correct sequence of correct dialogue acts given a conversation? Stolcke et al. (1998) and Taylor et al. (1998) apply the HMM intuition of Woszczyna and Waibel (1994) to treat the dialogue act detection process as HMM-parsing. Given all available evidence E about a conversation, the goal is to find the dialogue act sequence $D = \{d_1, d_2 \ldots, d_N\}$ that has the highest posterior probability $P(D|E)$ given that evidence (here we

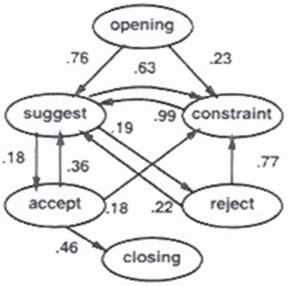

Figure 19.4 A dialogue act HMM (after Woszczyna and Waibel (1994))

are using capital letters to mean *sequences* of things). Applying Bayes' Rule we get

$$\begin{aligned} D^{*} &= \operatorname*{argmax}_{D} P(D|E) \\ &= \operatorname*{argmax}_{D} \frac{P(D)P(E|D)}{P(E)} \\ &= \operatorname*{argmax}_{D} P(D)P(E|D) \end{aligned} \qquad (19.15)$$

Here $P(D)$ represents the prior probability of a sequence of dialogue acts D. This probability can be computed by the dialogue act N-grams introduced by Nagata and Morimoto (1994). The likelihood $P(E|D)$ can be computed from the other two sources of evidence: the microsyntax models (for example the different word-N-gram grammars for each dialogue act) and the microprosody models (for example the decision tree for the prosodic features of each dialogue act). The word-N-grams models for each dialogue act can be used to estimate $P(W|D)$, the probability of the sequence of words W. The microprosody models can be used to estimate $P(F|D)$, the probability of the sequence of prosodic features F.

If we make the simplifying (but of course incorrect) assumption that the prosody and the words are independent, we can estimate the evidence likelihood for a sequence of dialogue acts D as follows:

$$P(E|D) = P(F|D)P(W|D) \qquad (19.16)$$

We can compute the most likely sequence of dialogue acts D^{*} by substituting equation (19.16) into equation (19.15), thus choosing the dialogue act sequence which maximizes the product of the three knowledge sources (conversational structure, prosody, and lexical/syntactic knowledge):

$$D^* = \operatorname*{argmax}_D P(D)P(F|D)P(W|D)$$

Standard HMM-parsing techniques (like Viterbi) can then be used to search for this most-probable sequence of dialogue acts given the sequence of input utterances.

The HMM method is only one way of solving the problem of data-driven dialogue act identification. The link with HMM tagging suggests another approach, treating dialogue acts as *tags*, and applying other part-of-speech tagging methods. Samuel et al. (1998b), for example, applied Transformation-Based Learning to dialogue act tagging.

Summary

As we have been suggesting, the two ways of doing dialogue act interpretation (via inference and via cues) each have advantages and disadvantages. The cue-based approach may be more appropriate for systems which require relatively shallow dialogue structure which can be trained on large corpora. If a semantic interpretation is required, the cue-based approach will still need to be augmented with a semantic interpretation. The full inferential approach may be more appropriate when more complex reasoning is required.

19.4 Dialogue Structure and Coherence

Section 18.2 described an approach to determining coherence based on a set of coherence relations. In order to determine that a coherence relation holds, the system must reason about the constraints that the relation imposes on the **information** in the utterances. We will call this view the *informational* approach to coherence. Historically, the informational approach has been applied predominantly to monologues.

The BDI approach to utterance interpretation gives rise to another view of coherence, which we will call the **intentional** approach. According to this approach, utterances are understood as actions, requiring that the hearer infer the plan-based speaker intentions underlying them in establishing coherence. In contrast to the informational approach, intentional approach has been applied predominantly to dialogue.

The intentional approach we describe here is due to Grosz and Sidner (1986), who argue that a discourse can be represented as a composite of three interacting components: a **linguistic structure**, an **intentional structure**, and an **attentional state**. The linguistic structure contains the utterances in the discourse, divided into a hierarchical structure of discourse segments. (Recall the description of discourse segments in Chapter 18.) The attentional state is a dynamically-changing model of the objects, properties, and relations that are salient at each point in the discourse. This aligns closely with the notion of a discourse model introduced in the previous chapter. Centering (see Chapter 18) is considered to be a theory of attentional state in this approach.

LINGUISTIC STRUCTURE

INTENTIONAL STRUCTURE

ATTENTIONAL STATE

We will concentrate here on the third component of the approach, the intentional structure, which is based on the BDI model of interpretation described in the previous section. The fundamental idea is that a discourse has associated with it an underlying purpose that is held by the person who initiates it, called the **discourse purpose** (DP). Likewise, each discourse segment within the discourse has a corresponding purpose, called a **discourse segment purpose** (DSP). Each DSP has a role in achieving the DP of the discourse in which its corresponding discourse segment appears. Listed below are some possible DPs/DSPs that Grosz and Sidner give.

DISCOURSE PURPOSE

DISCOURSE SEGMENT PURPOSE

1. Intend that some agent intend to perform some physical task.
2. Intend that some agent believe some fact.
3. Intend that some agent believe that one fact supports another.
4. Intend that some agent intend to identify an object (existing physical object, imaginary object, plan, event, event sequence).
5. Intend that some agent know some property of an object.

As opposed to the larger sets of coherence relations used in informational accounts of coherence, Grosz and Sidner propose only two such relations: **dominance** and **satisfaction-precedence**. DSP_1 dominates DSP_2 if satisfying DSP_2 is intended to provide part of the satisfaction of DSP_1. DSP_1 satisfaction-precedes DSP_2 if DSP_1 must be satisfied before DSP_2.

As an example, let's consider the dialogue between a client (C) and a travel agent (A) that we saw earlier, repeated here in Figure 19.5.

Collaboratively, the caller and agent successfully identify a flight that suits the caller's needs. Achieving this joint goal required that a top-level discourse intention be satisfied, listed as I1 below, in addition to several intermediate intentions that contributed to the satisfaction of I1, listed as I2-I5:

I1: (Intend C (Intend A (A find a flight for C)))

I2: (Intend A (Intend C (Tell C A departure date)))

C_1: I need to travel in May.
A_1: And, what day in May did you want to travel?
C_2: OK uh I need to be there for a meeting that's from the 12th to the 15th.
A_2: And you're flying into what city?
C_3: Seattle.
A_3: And what time would you like to leave Pittsburgh?
C_4: Uh hmm I don't think there's many options for non-stop.
A_4: Right. There's three non-stops today.
C_5: What are they?
A_5: The first one departs PGH at 10:00am arrives Seattle at 12:05 their time. The second flight departs PGH at 5:55pm, arrives Seattle at 8pm. And the last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.
C_6: OK I'll take the 5ish flight on the night before on the 11th.
A_6: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.
C_7: OK.

Figure 19.5 A fragment from a telephone conversation between a client (C) and a travel agent (A) (repeated from Figure 19.1).

I3: (Intend A (Intend C (Tell C A destination city)))

I4: (Intend A (Intend C (Tell C A departure time)))

I5: (Intend C (Intend A (A find a nonstop flight for C)))

Intentions I2–I5 are all subordinate to intention I1, as they were all adopted to meet preconditions for achieving intention I1. This is reflected in the dominance relationships below:

I1 dominates I2

I1 dominates I3

I1 dominates I4

I1 dominates I5

Furthermore, intentions I2 and I3 needed to be satisfied before intention I5, since the agent needed to know the departure date and destination city in order to start listing nonstop flights. This is reflected in the satisfaction-precedence relationships below:

I2 satisfaction-precedes I5

I3 satisfaction-precedes I5

The dominance relations give rise to the discourse structure depicted in Figure 19.6. Each discourse segment is numbered in correspondence with the intention number that serves as its DP/DSP.

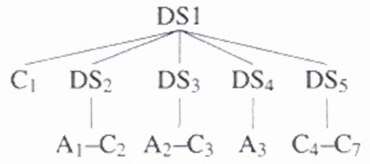

Figure 19.6 Discourse Structure of the Flight Reservation Dialogue

On what basis does this set of intentions and relationships between them give rise to a coherent discourse? It is their role in the overall *plan* that the caller is inferred to have. There are a variety of ways that plans can be represented; here we will use the simple STRIPS model described in the previous section. We make use of two simple action schemas; the first is the one for booking a flight, repeated from page 735.

BOOK-FLIGHT(A,C,F):

Constraints: Agent(A) ∧ Flight(F) ∧ Client(C)
Precondition: Know(A,departure-date(F)) ∧ Know(A,departure-time(F)) ∧ Know(A,origin-city(F)) ∧ Know(A,destination-city(F)) ∧ Know(A,flight-type(F)) ∧ Has-Seats(F) ∧ W(C,(BOOK(A,C,F))) ∧ ...

Effect: Flight-Booked(A,C,F)
Body: Make-Reservation(A,F,C)

As can be seen, booking a flight requires that the agent know a variety of parameters having to do with the flight, including the departure date and time, origin and destination cities, and so forth. The utterance with which the caller initiates the example dialogue contains the origin city and partial information about the departure date. The agent has to request the rest; the second action schema we use represents a simplified view of this action (see Cohen and Perrault (1979) for a more in-depth discussion of planning wh-questions):

REQUEST-INFO(A,C,I):
Constraints: Agent(A) $\wedge$ Client(C)
Precondition: Know(C,I)
Effect: Know(A,I)
Body: B(C,W(A,Know(A,I)))

Because the effects of REQUEST-INFO match each precondition of BOOK-FLIGHT, the former can be used to serve the needs of the latter. Discourse segments DS2 and DS3 are cases in which performing REQUEST-INFO succeeds for identifying the values of the departure date and destination city parameters respectively. Segment DS4 is also a request for a parameter value (departure time), but is unsuccessful in that the caller takes the initiative instead, by (implicitly) asking about nonstop flights. Segment DS5 leads to the satisfaction of the top-level DP from the caller's selection of a nonstop flight from a short list that the agent produced.

SUBDIALOGUES

INFORMATION-SHARING SUBDIALOGUES

Subsidiary discourse segments like DS2 and DS3 are also called **subdialogues**. The type of subdialogues that DS2 and DS3 instantiate are generally called **knowledge precondition** subdialogues (Lochbaum et al., 1990; Lochbaum, 1998), since they are initiated by the agent to help satisfy preconditions of a higher-level goal (in this case addressing the client's request for travel in May). They are also called **information-sharing subdialogues** (Chu-Carroll and Carberry, 1998).

CORRECTION SUBDIALOGUES

Later on in a part of the conversation not given in Figure 19.5 is another kind of subdialogue, a **correction subdialogue** (Litman, 1985; Litman and Allen, 1987) (or **negotiation subdialogue**; Chu-Carroll and Carberry (1998)). Utterances C_{20} through C_{23a} constitute a correction to the previous plan of returning on May 15:

A_{17}: And you said returning on May 15th?

C_{18}: Uh, yeah, at the end of the day.

A_{19}: OK. There's #two non-stops . . . #

C_{20}: #Act. . . actually#, what day of the week is the 15th?

A_{21}: It's a Friday.

C_{22}: Uh hmm. I would consider staying there an extra day til Sunday.

A_{23a}: OK. . . OK.

A_{23b}: On Sunday I have . . .

SUBTASK

Finally, perhaps the earliest class of subdialogues to be addressed in the literature was the **subtask** subdialogue (Grosz, 1974), which is used to deal with subtasks of the overall task in a task-oriented dialogue.

Determining Intentional Structure Algorithms for inferring intentional structure in dialogue (and spoken monologue) work similarly to algorithms for inferring dialogue acts. Many algorithms apply variants of the BDI model (e.g., Litman, 1985; Grosz and Sidner, 1986; Litman and Allen, 1987; Carberry, 1990; Passonneau and Litman, 1993; Chu-Carroll and Carberry, 1998). Others rely on similar cues to those described for utterance- and turn-segmentation on page 724, including cue words and phrases (Reichman, 1985; Grosz and Sidner, 1986; Hirschberg and Litman, 1993), prosody (Grosz and Hirschberg, 1992; Hirschberg and Pierrehumbert, 1986; Hirschberg and Nakatani, 1996), and other cues. For example Pierrehumbert and Hirschberg (1990) argue that certain **boundary tones** might be used to suggest a dominance relation between two intonational phrases.

BOUNDARY TONES

Informational vs. Intentional Coherence As we just saw, the key to intentional coherence lies in the ability of the dialogue participants to recognize each other's intentions and how they fit into the plans they have. On the other hand, as we saw in the previous chapter, informational coherence lies in the ability to establish certain kinds of content-bearing relationships between utterances. So one might ask what the relationship between these are: does one obviate the need for the other, or do we need both?

Moore and Pollack (1992), among others, have argued that in fact both levels of analysis must co-exist. Let us assume that after our agent and caller have identified a flight, the agent makes the statement in passage (19.17).

(19.17) You'll want to book your reservations before the end of the day.
Proposition 143 goes into effect tomorrow.

This passage can be analyzed either from the intentional or informational perspective. Intentionally, the agent intends to convince the caller to book her reservation before the end of the day. One way to accomplish this is to provide motivation for this action, which is the role served by uttering the second sentence. Informationally, the two sentences satisfy the Explanation relation described in the last chapter, since the second sentence provides a cause for the effect of wanting to book the reservations before the end of the day.

Depending on the knowledge of the caller, recognition at the informational level might lead to recognition of the speaker's plan, or vice versa. Say, for instance, that the caller knows that Proposition 143 imposes a new tax on airline tickets, but did not know the intentions of the agent in uttering the second sentence. From the knowledge that a way to motivate an action is to provide a cause that has that action as an effect, the caller can surmise that the agent is trying to motivate the action described in the first sentence. Alternatively, the caller might have surmised this intention from the discourse scenario, but have no idea what Proposition 143 is about. Again, knowing the relationship between establishing a cause-effect relationship and motivating something, the caller might be led to assume an Explanation relationship, which would require that she infers that the proposition is somehow bad for airline ticket buyers (e.g., a tax). Thus, at least in some cases, both levels of analysis appear to be required.

19.5 Dialogue Managers in Conversational Agents

The idea of a conversational agent is a captivating one, and conversational agents like **ELIZA**, **PARRY**, or **SHRDLU** have become some of the best-known examples of natural language technology. Modern examples of conversational agents include airline travel information systems, speech-based restaurant guides, and telephone interfaces to email or calendars. The dialogue manager is the component of such conversational agents that controls the flow of the dialogue, deciding at a high level how the agent's side of the conversation should proceed, what questions to ask or statements to make, and when to ask or make them.

This section briefly summarizes some issues in dialogue manager design, discussing some simple systems based on finite-state automata and production rules, and some more complex ones based on more sophisticated BDI-style reasoning and planning techniques.

The simplest dialogue managers are based on finite-state automata. For example, imagine a trivial airline travel system whose job was to ask the user for a departure city, a destination city, a time, and any airline preference. Figure 19.7 shows a sample dialogue manager for such a system. The states of the FSA correspond to questions that the dialogue manager asks the user, and the arcs correspond to actions to take depending on what the user responds.

SINGLE INITIATIVE
SYSTEM INITIATIVE

Systems which completely control the conversation in this way are called **single initiative** or **system initiative** systems. While this simple dialogue manager architecture is sufficient for some tasks (for example for implementing a speech interface to an automatic teller machine or a simple geography quiz), it is probably too restricted for a speech based travel agent system (see the discussion in McTear (1998)). One reason is that it is convenient for users to use more complex sentences that may answer more than one question at a time, as in the following ATIS example:

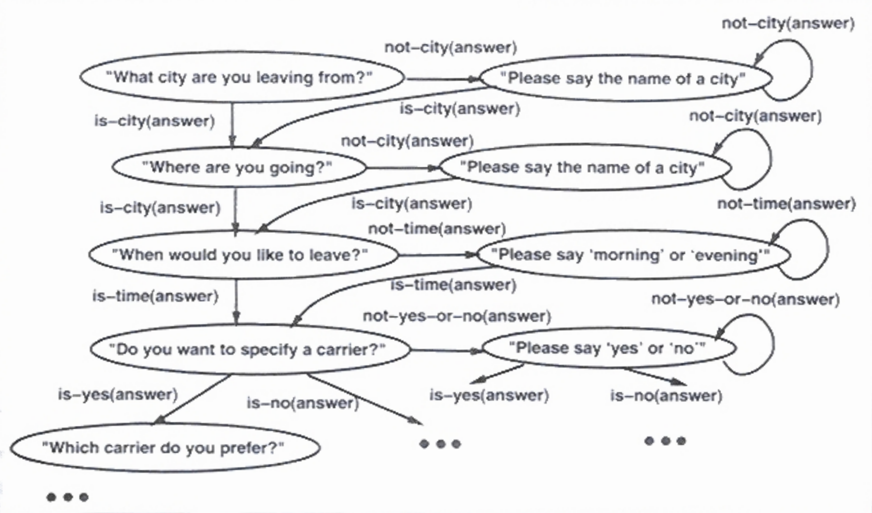

Figure 19.7 A simple finite-state automaton architecture for a dialogue manager.

I want a flight from Milwaukee to Orlando one way leaving after
five p.m. on Wednesday.

Many speech-based question answering systems, beginning with the influential GUS system for airline travel planning (Bobrow et al., 1977), and including more recent ATIS systems and other travel and restaurant guides, are **frame**- or **template**-based. For example, a simple airline system might have the goal of helping a user find an appropriate flight. It might have a frame or template with slots for various kinds of information the user might need to specify. Some of the slots come with prespecified questions to ask the user:

FRAME
TEMPLATE

Slot	Optional Question
From_Airport	"From what city are you leaving?"
To_Airport	"Where are you going?"
Dep_time	"When would you like to leave?"
Arr_time	"When do you want to arrive?"
Fare_class	
Airline	
Oneway	

Such a simple dialogue manager may just ask questions of the user, filling out the template with the answers, until it has enough information to perform a data base query, and then return the result to the user. Not every slot may have an associated question, since the dialogue designer may not want the user deluged with questions. Nonetheless, the system must be able to fill these slots if the user happens to specify them.

Even such simple domains require more than this single-template architecture. For example, there is likely to be more than one flight which meet the user's constraints. This means that the user will be given a list of choices, either on a screen or, for a purely telephone interface, by listing them verbally. A template-based system can then have another kind of template which has slots for identifying elements of lists of flights (*How much is the first one?* or *Is the second one non-stop?*). Other templates might have general route information (for questions like *Which airlines fly from Boston to San Francisco?*), information about airfare practices (for questions like *Do I have to stay a specific number of days to get a decent airfare?*) or about car or hotel reservations. Since users may switch from template to template, and since they may answer a future question instead of the one the system asked, the system must be able to disambiguate which slot of which template a given input is supposed to fill, and then switch dialogue control to that template. A template-based system is thus essentially a production rule system. Different types of inputs cause different productions to fire, each of which can flexibly fill in different templates. The production rules can then switch control based on factors such as the user's input and some simple dialogue history like the last question that the system asked.

The template or production-rule dialogue manager architecture is often used when the set of possible actions the user could want to take is relatively limited, but where the user might want to switch around a bit among these things.

The limitations of both the template-based and FSA-based dialogue managers are obvious. Consider the client's utterance C_4 in the fragment of sample dialogue of Figure 19.5 on page 746, repeated here:

A_3: And what time would you like to leave Pittsburgh?

C_4: Uh hmm I don't think there's many options for non-stop.

A_4: Right. There's three non-stops today.

C_5: What are they?

A_5: The first one departs PGH at 10:00a.m. . . .

INITIATIVE

What the client is doing in C_4 is taking control or **initiative** of the dialogue. C_4 is an indirect request, asking the agent to check on non-stop flights. It would not be appropriate for the system to just set the WANTS NON-STOP field in a template and ask the user again for the departure time.

The system needs to realize that the user has indicated that a non-stop flight is a priority and that the system should focus on that next.

Conversational agents also need to use the **grounding** acts described on page 725. For example, when the user makes a choice of flights, it's important for the agent to indicate to the client that it has understood this choice. Repeated below is an example of such grounding excerpted from our sample conversation:

C_6: OK I'll take the 5ish flight on the night before on the 11th.
A_6: On the 11th? OK.

It is also important for a computational conversational agent to use requests for repairs, since given the potential for errors in the speech recognition or the understanding, there will often be times when the agent is confused or does not understand the user's request.

In order to address these and other problems, more sophisticated dialogue managers can be built on the BDI (belief, desire, intention) architecture described on page 734. Such systems are often integrated with logic-based planning models, and treat a conversation as a sequence of actions to be planned.

Let's consider the dialogue manager of the TRAINS-93 system; the system is described in Allen et al. (1995), the dialogue manager in Traum and Allen (1994). The TRAINS system is a spoken-language conversational planning agent whose task is to assist the user in managing a railway transportation system in a microworld. For example, the user and the system might collaborate in planning to move a boxcar of oranges from one city to another. The TRAINS dialogue manager maintains the flow of conversation and addresses the conversational goals (such as coming up with a operational plan for achieving the domain goal of successfully moving oranges). To do this, the manager must model the state of the dialogue, its own intentions, and the user's requests, goals, and beliefs. The manager uses a conversation act interpreter to semantically analyze the user's utterances, a domain planner and executer to solve the actual transportation domain problems, and a generator to generate sentences to the user. Figure 19.8 shows an outline of the TRAINS-93 dialogue manager algorithm.

The algorithm keeps a queue of conversation acts it needs to generate. Acts are added to the queue based on **grounding**, **dialogue obligations**, or the agent's **goals**. Let's examine each of these sources. Grounding acts were discussed on page 724; recall that a previous utterance can be grounded by an explicit backchannel (e.g., *uh-huh*, *yeah*, or under certain circumstances *ok*),

```
DIALOGUE_MANAGER

while conversation is not finished
  if user has completed a turn
  then interpret user's utterance
  if system has obligations
  then address obligations
  else if system has turn
  then if system has intended conversation acts
      then call generator to produce NL utterances
      else if some material is ungrounded
      then address grounding situation
      else if high-level goals are unsatisfied
      then address goals
      else release turn or attempt to end conversation
  else if no one has turn
  then take turn
  else if long pause
  then take turn
```

Figure 19.8 A dialogue manager algorithm, slightly modified from Traum and Allen (1994).

or by repeating back part of the utterance. Utterances can also be grounded implicitly by "taking up" the utterance, i.e., continuing in a way which makes it clear that the utterance was understood, such as by answering a question.

Obligations are used in the TRAINS system to enable the system to correctly produce the second-pair part of an adjacency pair. That is, when a user REQUESTs something of the system (e.g., REQUEST(Give(List)), or REQUEST(InformIf(NonStop(FLIGHT-201)))), the REQUEST sets up an obligation for the system to address the REQUEST either by accepting it, and then performing it (giving the list or informing whether flight 201 is non-stop), or by rejecting it.

Finally, the TRAINS dialogue manager must reason about its own goals. For the travel agent domain, the dialogue manager's goal might be to find out the client's travel goal and then create an appropriate plan. Let's pretend that the human travel agent for the conversation in Figure 19.5 was a system and explore what the state of a TRAINS-style dialogue manager would have to be to act appropriately. Let's start with the state of the dia-

METHODOLOGY BOX: DESIGNING DIALOGUE SYSTEMS

How does a dialogue system developer choose dialogue strategies, architectures, prompts, error messages, and so on? The three design principles of Gould and Lewis (1985) can be summarized as:

> **Key Concept #8. User-Centered Design:** Study the user and task, build simulations and prototypes, and iteratively test them on the user and fix the problems.

1. Early Focus on Users and Task: Understand the potential users and the nature of the task, via interviews with users and investigation of similar systems. Study of related human-human dialogues can also be useful, although the language in human-machine dialogues is usually simpler than in human-human dialogues. (For example pronouns are rare in human-machine dialogue and are very locally bound when they do occur (Guindon, 1988)).

2. Build Prototypes: In the children's book *The Wizard of Oz* (Baum, 1900), the Wizard turned out to be just a simulation controlled by a man behind a curtain. In Wizard-of-Oz (WOZ) or PNAMBIC (Pay No Attention to the Man BehInd the Curtain) systems, the users interact with what they think is a software system, but is in fact a human operator ("wizard") behind some disguising interface software (e.g. Gould et al., 1983; Good et al., 1984; Fraser and Gilbert, 1991) . A WOZ system can be used to test out an architecture without implementing the complete system; only the interface software and databases need to be in place. It is difficult for the wizard to exactly simulate the errors, limitations, or time constraints of a real system; results of WOZ studies are thus somewhat idealized.

3. Iterative Design: An iterative design cycle with embedded user testing is essential in system design (Nielsen, 1992; Cole et al., 1994, 1997; Yankelovich et al., 1995; Landauer, 1995). For example Stifelman et al. (1993) and Yankelovich et al. (1995) found that users of speech systems consistently tried to interrupt the system (**barge in**), suggesting a redesign of the system to recognize overlapped speech. Kamm (1994) and Cole et al. (1993) found that **directive prompts** ("Say *yes* if you accept the call, otherwise, say *no*") or the use of constrained forms (Oviatt et al., 1993) produced better results than open-ended prompts like "Will you accept the call?".

logue manager (formatted following Traum and Allen (1994)) after the first utterances in our sample conversation (repeated here):

C_1: I want to go to Pittsburgh in May.

The client/user has just finished a turn with an INFORM speech act. The system has the discourse goal of finding out the user's travel goal (e.g., "Wanting to go to Pittsburgh on may 15 and returning ..."), and creating a travel plan to accomplish that goal. The following table shows the five parameters of the system state: the list of obligations, the list of intended speech acts to be passed to the generator, the list of the user's speech acts that still need to be acknowledged, the list of discourse goals, and whether the system or the user holds the turn:

Discourse obligations:	NONE
Turn holder:	system
Intended speech acts:	NONE
Unacknowledged speech acts:	INFORM-1
Discourse goals:	get-travel-goal, create-travel-plan

After the utterance, the dialogue manager decides to add two conversation acts to the queue; first, to acknowledge the user's INFORM act (via "address grounding situation"), and second, to ask the next question of the user (via "address goals"). This reasoning would be worked out by the system's STRIPS-style planner as described on page 747; given the goal *get-travel-goal*, the REQUEST-INFO action schema tells the system that asking the user something is one way of finding it out. The result of adding these two conversation acts is:

Intended speech acts: REQUEST-INFORM-1, ACKNOWLEDGE-1

These would be combined by a very clever generator into the single utterance:

A_2: And, what day in May did you want to travel?

Note that the grounding function was achieved both by beginning with the discourse marker *and* and by repeating back the month name *May*. The request for information is achieved via the wh-question.

Let's skip ahead to the client's utterance C_4. Recall that C_4 is an indirect request, asking the agent to check on non-stop flights.

A_3: And what time would you like to leave Pittsburgh?

C_4: Uh hmm I don't think there's many options for non-stop.

Let's assume that our dialogue act interpreter correctly interprets C_4 as REQUEST-INFORM-3. The state of the agent after client utterance C_4 is then:

Discourse obligations:	address(REQUEST-INFORM-3)
Turn holder:	system
Intended speech acts:	NONE
Unacknowledged speech acts:	REQUEST-INFORM-3
Discourse goals:	get-travel-goal, create-travel-plan

The dialogue manager will first address the discourse obligation of responding to the user's request by calling the planner to find out how many non-stop flights there are. The system must now answer the question, but must also ground the user's utterance. For a direct request, the response is sufficient grounding. For an indirect request, an explicit acknowledgement is an option; since the indirect request was in the form of a *negative* check question, the form of acknowledgement will be *right* (*no* would have also been appropriate for acknowledging a negative). These two acts will then be pulled off the queue and passed to the generator:

A_4: Right. There's three non-stops today.

Dialogue managers also will need to deal with the kind of dialogue structure discussed in Section 19.4, both to recognize when the user has started a subdialogue, and to know when to initiate a subdialogue itself.

19.6 Summary

Dialogue is a special kind of discourse which is particularly relevant to speech processing tasks like **conversational agents** and **automatic meeting summarization**.

- Dialogue differs from other discourse genres in exhibiting **turn-taking**, **grounding**, and **implicature**.
- An important component of dialogue modeling is the interpretation of **dialogue acts**. We introduced **plan-based** and **cue-based** algorithms for this.
- Dialogue exhibits **intentional structure** in addition to the **informational structure**, including such relations as **dominance** and **satisfaction-precedence**.
- Dialogue managers for conversational agents range from simple template- or frame-based **production systems** to complete **BDI** (**belief-desire-intention**) models.

METHODOLOGY BOX: EVALUATING DIALOGUE SYSTEMS

Many of the metrics that have been proposed for evaluating dialogue systems can be grouped into the following three classes:

1. User Satisfaction: Usually measured by interviewing users (Stifelman et al., 1993; Yankelovich et al., 1995) or having them fill out questionnaires asking e.g. (Shriberg et al., 1992; Polifroni et al., 1992):

- Were answers provided quickly enough?
- Did the system understand your requests the first time?
- Do you think a person unfamiliar with computers could use the system easily?

2. Task Completion Cost:

- Completion time in turns or seconds (Polifroni et al., 1992).
- Number of queries (Polifroni et al., 1992).
- Number of system non-responses (Polifroni et al., 1992) or "turn correction ratio": the number of system or user turns that were used solely to correct errors, divided by the total number of turns (Danieli and Gerbino, 1995; Hirschman and Pao, 1993).
- Inappropriateness (verbose or ambiguous) of system's questions, answers, and error messages (Zue et al., 1989).

3. Task Completion Success:

- Percent of subtasks that were completed (Polifroni et al., 1992).
- Correctness (or partial correctness) of each question, answer, error message (Zue et al., 1989; Polifroni et al., 1992).
- Correctness of the total solution (Polifroni et al., 1992).

How should these metrics be combined and weighted? The PARADISE algorithm (Walker et al., 1997) (PARAdigm for DIalogue System Evaluation) applies multiple regression to this problem. The algorithm first uses questionnaires to assign each dialogue a user satisfaction rating. A set of cost and success factors like those above is then treated as a set of independent factors; multiple regression is used to train a weight (coefficient) for each factor, measuring its importance in accounting for user satisfaction. The resulting metric can be used to compare quite different dialogue strategies.

Bibliographical and Historical Notes

Early work on speech and language processing had very little emphasis on the study of dialogue. One of the earliest conversational systems, ELIZA, had only a trivial production system dialogue manager; if the human user's previous sentence matched the regular-expression precondition of a possible response, ELIZA simply generated that response (Weizenbaum, 1966). The dialogue manager for the simulation of the paranoid agent PARRY (Colby et al., 1971), was a little more complex. Like ELIZA, it was based on a production system, but where ELIZA's rules were based only on the words in the user's previous sentence, PARRY's rules also rely on global variables indicating its emotional state. Furthermore, PARRY's output sometimes makes use of script-like sequences of statements when the conversation turns to its delusions. For example, if PARRY's **anger** variable is high, he will choose from a set of "hostile" outputs. If the input mentions his delusion topic, he will increase the value of his **fear** variable and then begin to express the sequence of statements related to his delusion.

The appearance of more sophisticated dialogue managers awaited the better understanding of human-human dialogue. Studies of the properties of human-human dialogue began to accumulate in the 1970's and 1980's. The Conversation Analysis community (Sacks et al., 1974; Jefferson, 1984; Schegloff, 1982) began to study the interactional properties of conversation. Grosz's (1977b) dissertation significantly influenced the computational study of dialogue with its introduction of the study of substructures in dialogues (subdialogues), and in particular with the finding that "task-oriented dialogues have a structure that closely parallels the structure of the task being performed" (p. 27). The BDI model integrating earlier AI planning work (Fikes and Nilsson, 1971) with speech act theory (Austin, 1962; Gordon and Lakoff, 1971; Searle, 1975a) was first worked out by Cohen and Perrault (1979), showing how speech acts could be generated, and Perrault and Allen (1980) and Allen and Perrault (1980), applying the approach to speech-act interpretation.

The cue-based model of dialogue act interpretation was inspired by Hinkelman and Allen (1989), who showed how lexical and phrasal cues could be integrated into the BDI model, and by the work on micrograммar in the Conversation Analysis literature (e.g. Goodwin, 1996). It was worked out at a number of mainly speech recognition labs around the world in the late 1990's (e.g. Nagata and Morimoto, 1994; Suhm and Waibel, 1994; Mast et al., 1996; Jurafsky et al., 1997; Warnke et al., 1997; Reithinger and Klesen, 1997; Taylor et al., 1998).

Models of dialogue as collaborative behavior were introduced in the late 1980's and 1990's, including the ideas of reference as a collaborative process (Clark and Wilkes-Gibbs, 1986), and models of **joint intentions** (Levesque et al., 1990), and **shared plans** (Grosz and Sidner, 1980). Related to this area is the study of **initiative** in dialogue, studying how the dialogue control shifts between participants (Walker and Whittaker, 1990; Smith and Gordon, 1997).

EXERCISES

19.1 List the dialogue act misinterpretations in the *Who's On First* routine at the beginning of the chapter.

19.2 Write a finite-state automaton for a dialogue manager for checking your bank balance and withdrawing money at an automated teller machine.

19.3 Dispreferred responses (for example turning down a request) are usually signaled by surface cues, such as significant silence. Try to notice the next time you or someone else utters a dispreferred response, and write down the utterance. What are some other cues in the response that a system might use to detect a dispreferred response? Consider non-verbal cues like eye-gaze and body gestures.

19.4 When asked a question to which they aren't sure they know the answer, people use a number of cues in their response. Some of these cues overlap with other dispreferred responses. Try to notice some unsure answers to questions. What are some of the cues? If you have trouble doing this, you may instead read Smith and Clark (1993) which lists some such cues, and try instead to listen specifically for the use of these cues.

19.5 The sentence "*Do you have the ability to pass the salt?*" is only interpretable as a question, not as an indirect request. Why is this a problem for the BDI model?

19.6 Most universities require Wizard-of-Oz studies to be approved by a human subjects board, since they involve deceiving the subjects. It is a good idea (indeed it is often required) to "debrief" the subjects afterwards and tell them the actual details of the task. Discuss your opinions of the moral issues involved in the kind of deceptions of experimental subjects that take place in Wizard-of-Oz studies.

19.7 Implement a small air-travel help system. Your system should get constraints from the user about a particular flight that they want to take, expressed in natural language, and display possible flights on a screen. Make simplifying assumptions. You may build in a simple flight database or you may use an flight information system on the web as your backend.

19.8 Augment your previous system to work over the phone (or alternatively, describe the user interface changes you would have to make for it to work over the phone). What were the major differences?

19.9 Design a simple dialogue system for checking your email over the telephone. Assume that you had a synthesizer which would read out any text you gave it, and a speech recognizer which transcribed with perfect accuracy. If you have a speech recognizer or synthesizer, you may actually use them instead.

19.10 Test your email-reading system on some potential users. If you don't have an actual speech recognizer or synthesizer, simulate them by acting as the recognizer/synthesizer yourself. Choose some of the metrics described in the Methodology Box on page 758 and measure the performance of your system.