Discourse Information for Disambiguation: The Phoenix Approach in JANUS M.S. Project Report

Maite Taboada
Computational Linguistics Program
Carnegie Mellon University
Pittsburgh, PA 15213
taboada+@cmu.edu

Committee:
Lori Levin Alex Waibel
Nancy Green Alon Lavie

May 12, 1997

1 Problem Statement

For any given utterance out of what we can loosely call *context*, there is usually more than one possible interpretation. A speaker's utterance of an elliptical expression, like the figure "twelve fifteen", might have a different meaning depending on the discourse context, the way the conversation has evolved until that point, and the previous speaker's utterance.

If this is a problem for any human listener, the problem grows considerably when it is a parser doing the disambiguation. In this project, I intend to help a parser disambiguate among different possible parses for an input sentence, with the final goal of improving the translation in an end-to-end speech translation system.

2 Significance for Computational Linguistics

One of the problems machine translation within a speech system faces is the presence of disfluencies and recognizer errors. A grammar designed to accept only perfectly formed sentences will fail on this type of input. The Phoenix parser [Ward 91, Ward 94] was designed to capture the content of spoken dialogue through *semantic grammars*, which assign an input string to a concept, putting together a sequence of concepts that might form a complete thought or sequence, usually assigning a speech act to the sequence. If we are able to understand illocutionary force in the input language, we can produce the same illocutionary force in the output language, irrespective of the actual

surface forms the two languages might use. In this sense, we depart from syntax-based, literal translation to provide a translation of content. However, the speech act, and the illocutionary force it carries can only be fully interpreted when contextual information is taken into account.

3 System Background

JANUS is a multi-lingual speech-to-speech translation system designed to translate spontaneous dialogue between two speakers in a limited domain [Waibel 96, Lavie et al. 96b]. It is designed to deal with the kind of problems that naturally occur in spontaneous speech—such as mispronunciations, restarts, noises, loose notions of grammaticality, and the lack of clear sentence boundaries—as well as additional errors introduced by the speech recognizer. The machine translation component of JANUS handles these problems using two different approaches: GLR* and Phoenix. The GLR* parser [Lavie 95, Lavie and Tomita 93] is designed to be more accurate, whereas the Phoenix parser [Ward 91, Ward 94] is more robust. Both are language-independent and follow an interlingua-based approach. The current system translates spontaneous dialogues in the Scheduling domain, with English, Spanish, and German as both source and target languages. See Figure 1 for an outline of all the system components.

The input string in the source language is first analyzed independently by the parsers, to produce a language-independent content representation. From that representation the generation component in each of the modules generates the output string in the target language. Additionally, the GLR* module contains a discourse processor, which disambiguates the speech act of each sentence, normalizes temporal expressions and incorporates the sentence into a discourse plan tree. The discourse processor is described in [Rosé et al. 95, Qu et al. 96a, Qu et al. 96b].

This project focuses on the Phoenix module of the machine translation component. The JANUS Phoenix translation module [Mayfield et al. 1995] was designed for semantic grammars. The parsing grammar specifies patterns in order to introduce grammatical constraints at the phrase level rather than at the sentence level. This method captures the semantic content of a complete input string, regardless of the ungrammaticalities often occurring between phrases. The patterns in the grammar help the parser extract a structure of concepts by means of tokens. Top-level tokens represent speech acts, whereas the intermediate and lower-level tokens capture the more specific parts of the utterance—such as days of the week or times in the scheduling domain. The inputs to the parser are fragments of a turn, each one of them a complete thought, although not necessarily a complete sentence. The segmentation of input is described in more detail in [Lavie et al. 96a]. Each one of those segments is parsed and generated devoid of contextual information.

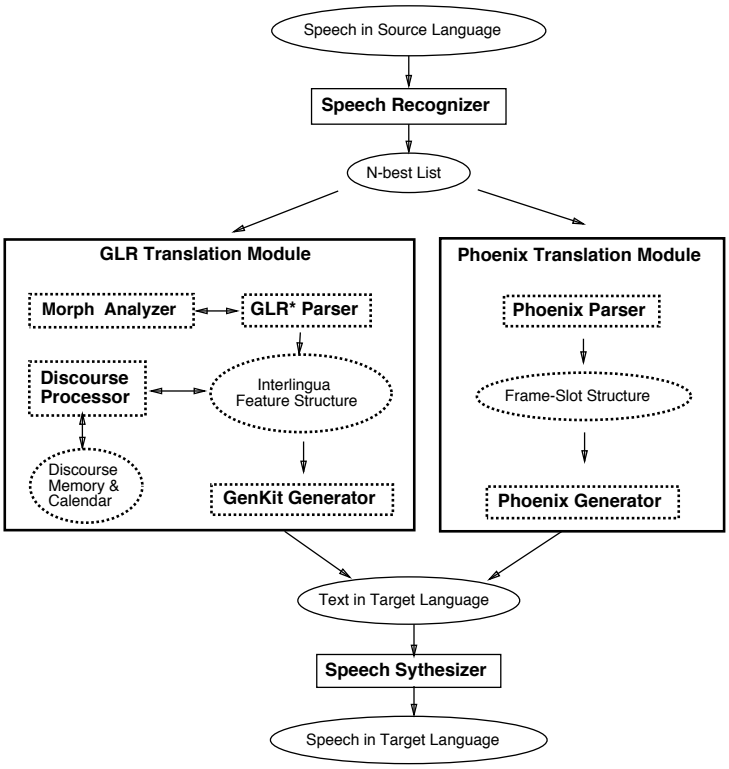

Figure 1: **The JANUS System**

4 Review of Relevant Literature

The work carried out in this research project draws from different areas of research both in Linguistics and in Computational Linguistics. On the linguistic side, the relevant aspects are the study and taxonomy of speech acts and general studies on conversation analysis. From the Computational Linguistics literature the many studies on processing discourse will be relevant, no matter how different their approaches to discourse processing and discourse structure itself might be.

4.1 Speech Act Theory

Austin's theory of speech acts [Austin 62], further extended by Searle [Searle 79], is the initiating piece of work in the theory of speech acts. Austin classified the various acts connected with speaking into *locutionary, illocutionary* and *perlocutionary acts*. *Locutionary acts* can be divided in three subtypes, of which the *phonetic* and *phatic* have to do with producing sounds, whereas the *rhetic* is the act of using sounds with a certain sense and reference.

An *illocutionary act* is the force attached to a rhetic act, an act performed in speaking, independent of the surface form used. A simple statement, like (1), may be used by the speaker to prompt the hearer to water the plant, it might be an informative act whereby the speaker diagnoses the reason for the plant's bad looks, or it might even be a criticism to the person in charge of watering the plant.

(1) This plant needs water.

The determination of the illocutionary force is the main goal of any taxonomy of speech acts that intends to perform an automatic classification. Attaching illocutionary force to a rhetic act becomes relevant when the taxonomy is to be applied to a particular domain. The sentence in example (2) is a statement that can have different refinements. It can express the intention of the speaker to communicate the hearer that she wants him[1] to meet her mother-in-law, it might state the feelings of happiness or unhappiness of the speaker towards the event, etc. But if we are in a domain where two participants are scheduling an appointment, the most likely interpretation is that the speaker is expressing a constraint in her schedule for that day.

(2) My mother-in-law is coming to town next Saturday.

Likewise, in a different domain, such as making travel arrangements, we find the same lack of close correlation between surface form (yes/no question) and illocutionary force (request), as in example (3), where the speaker is not really inquiring into whether she can be of help, she is rather prompting the hearer to make a request.

(3) Pittsburgh Travel. Can I help you?

The third category in Austin's classification is that of *perlocutionary acts*, non-conventional acts where the speaker causes a natural condition or state in a person just by uttering an expression. We will not be concerned here with perlocutionary acts, which include acts such as harassing, irritating, pleasing and boring.

Searle [Searle 79] takes an illocutionary act to consist of two parts: the force and the proposition (which corresponds to Austin's rhetic act). From the five types of illocutionary acts Austin had established—verdictives, exercitives, commissives, behabitives, and expositives—, Searle presented a different taxonomy—which included assertives, directives, commissives, expressives, and declarations—based on features such as the role of authority and discourse relations.

There exists an important amount of work on the concept of speech act itself, together with other types of research which make use of speech acts, especially in natural language processing.

In the Verbmobil scenario, Schmitz and Quantz [Schmitz and Quantz 95] defend the combination of illocutionary and propositional information in their *dialogue acts*. Illocutionary acts are detached from propositional content and are too abstract for dialogue processing. A machine translation system needs to be able to automatically classify different utterances according to the type of dialogue act they represent. For that purpose the classification of dialogue acts needs to be specific and restricted to those characteristic of a certain domain and type of dialogue. Dialogue acts are divided into *dialogue control acts* and *task-oriented acts*.

[1] In this paper, I will use the convention of assigning female gender to the speaker or initiator of the conversation, and male to the hearer

The view on speech acts used in this project is mainly Searle's, where the name assigned to the speech act tries to capture both the illocutionary force and the proposition, such as `request-info`, `acknowledge`, or `provide-confirmation`.

One of the main tasks within this project is to combine different classifications of speech acts into a general, but domain-oriented taxonomy. One such attempt at devising a taxonomy that is reusable and general enough to code discourse is the DRI classification, as in Figure 1 [Allen and Core 97].

4.2 Conversation Analysis

Both the Birmingham school [Sinclair and Coulthard 75], from an ethnomethodological point of view, and the hallidayan circle [Halliday 94, Martin 92], from a functional perspective, have studied what kind of processes develop in conversation. The work of Sinclair and Coulthard brought about the notion of moves, a way to structure sub-pieces of an interaction. Schegloff and Sacks [Schegloff and Sacks 73] introduced *adjacency pairs*, to capture the expected follow-up for certain utterances. Conversational analysis in general provides insight into the global organization of dialogues

The early work of Sinclair [Sinclair 66] emphasized paragraphing in spoken discourse, affirming that conversations are everyday examples of the fact that several participants can jointly produce coherent texts. The kind of questions he wanted to answer were: how are successive utterances related; who controls the discourse; how does he or she do it; how, if at all, do other participants take control; how do the roles of speaker and listener pass from one participant to another; how are new topics introduced and old ones ended; what linguistic evidence is there for discourse units larger than the utterance.

The findings from corpus analysis were interesting: in normal conversation, where participants are of equal status and have equal rights to determine the topic, changes in topic are almost unpredictable. In the following example, taken from [Sinclair and Coulthard 75], the first utterance is the ending of the topic 'rifles'. Then the speakers try to establish another topic, but each speaker has the right to veto, so both 'driving' and 'discipline' are rejected, until they settle on 'drilling'.

(4) S1 We got these exercises and you're to take the butt and hold point it away up there and we couldn't. Our arm used to shoot up and down it came.

S2 Well I joined for these reasons and plus the driving you get taught you're taught to drive.

S3 Well also my father said I need a bit of discipline you know.

S1 There's none there.

S2 You won't get any there honestly it's just terrific.

S3 That's why I'm joining it make him think I'm getting discipline.

S1 Oh it's great fun isn't it.

S2 Oh but wait have you been on a drilling yet?

S3 No.

S2 Just you wait.

Communicative-Status	Uninterpretable Abandoned Self-talk
Information-Level	Task Task-management Communication-management Other-level
Forward Communicative Function	Statement Assert Reassert Other-statement Influencing-addressee-future-action Open-option Directive - Info-request - Action-directive Commiting-speaker-future-action Offer Commit Other-forward-function Conventional-opening Conventional-closing Explicit-performative Exclamation Other-forward-function
Backward Communicative Function	Agreement Accept Accept-part Maybe Reject-part Reject Hold Understanding Signal-non-understanding Signal-understanding - Acknowledge - Repeat-rephrase - Completion Correct-misspeaking Answer Information-relation

Table 1: DRI Classification

Non-Linguistic Organization	DISCOURSE	Grammar
course		
period	LESSON	
topic	TRANSACTION	
	EXCHANGE	
	MOVE	sentence
	ACT	clause
		group
		word
		morpheme

Figure 2: Levels and Ranks

Part of the advantage of task-oriented dialogues, like the ones I have been dealing with in this project, is that topics are not negotiated in a strict sense. They are introduced in the conversation by one of the participants, and usually accepted by the other one.

The second difficulty Sinclair and Coulthard found in analyzing conversation was the introduction of digressions. In our corpus, we have to deal with clarifications and corrections. The third problem is the abuse of language. In example (5)[2] the father is using the resources of language to avoid giving a direct command. However, because he uses an interrogative formulation, his son is able to ignore the intended command and reply as if it were a question. We assume the speakers in our dialogues to be cooperative, so we assume that they will not make such use of language. Our problem in that sense lies with the parser, because it can understand the sentences "literally", i.e., the same surface form can have many different illocutionary forces associated with them, so the parser might not be able to differenciate among them.

(5) Father: Is that your coat on the floor again?
Son: Yes. (Goes on reading).

For their analysis of classroom interaction, Sinclair and Coulthard developed a system of analysis that would capture the different levels of organization of discourse. Figure 2 outlines their structuring.

The relevant part here is the level that starts at the exchange under Discourse[3], generally a subdialogue dealing with a specific subtopic. A move is the turn each participant takes, and the acts—clauses in the grammar—are the segments that the parser is trying to analyze in order to give them a speech act label.

Coulthard and Brazil [Coulthard and Brazil 92] stress the importance of Conversation Analysis, and especially of Sacks's notion of *adjacency pairs*. Sacks [Sacks n.d.] observes that a conversation is a string of at least two turns, some of them more closely related than others, the adjacency pairs. They have certain features: they are two utterances long; the utterances are produced by

[2]Example taken from [Sinclair and Coulthard 75].

[3]This organization was designed to model the interaction in a classroom, that is why Leson is the higher unit at the Discourse level.

		initiation	expected response	discretionary alternative
give	goods & services	offer	acceptance	rejection
demand	goods & services	command	undertaking	refusal
give	information	statement	acknowledgement	contradiction
demand	information	question	answer	disclaimer

Figure 3: Speech Functions and Responses

two different speakers in succession; the utterances are ordered into *first pair parts* and *second pair parts*; the utterances are related in such a way that not any second class part can follow a first pair part; the first pair often selects next speaker and next action, setting up an expectation that the next speaker is supposed to fulfill.

However, in conversation, it is not difficult to find a question without an answer. Schegloff [Schegloff 72] talks about *insertion sequences* when there is an embedded sequence that interrupts the normal flow of an adjacency pair. That is the case with clarifications and corrections to an utterance. Once the clarification has been completed, the dialogue returns to the completion of the adjacency pair:

(6) A: Is the Conference Room reserved today?
B: You mean the Red Room or the Blue Room?
A: The Red Room.
B: Yes, it's reserved from 1 to 2 pm.

Halliday's [Halliday 94] *speech functions* are the expression of adjacency pairs in the situation where the clause is considered an exchange. He claims that in the act of speaking the speaker is adopting a speech role, and in so doing assigns to the listener a complementary role which she wishes him to adopt in his turn. For example, in asking a question the speaker is taking up the role of seeker of information, and assigning the role of supplier of information to the listener. The most fundamental types of speech roles are giving and demanding, and across that distinction there is another one that relates to the nature of the commodity being exchanged: goods-and-services or information. These two variables, taken together, define the four basic speech functions of OFFER, COMMAND, STATEMENT and QUESTION. Those are matched by a set of desired responses, with some alternatives, as in Figure 3.

A different approach to the layering of structure in discourse is taken in [Traum and Hinkelman 92], where they introduce the notion of *Conversation Acts*, a more general concept than speech acts, which includes not only the traditional speech acts but turn-taking, grounding, and higher-level argumentation acts. Their theory of speech acts is supposed to be more adjusted to deal with task-oriented dialogues. From the conventional concept of speech acts they reject a few assumptions:

- Utterances are heard and understood correctly.
- Speech acts are single agent plans executed by the speaker, with the listener as a passive agent.

Discourse Level	**Turn-taking**	**Sample Act**
Sub-utterance unit	Turn-taking	take-turn keep-turn release-turn assign-turn
Utterance unit	Grounding	Initiate Continue Ack Repair ReqRepair ReqAck Cancel
Discourse Unit	Core Speech Acts	Inform WHQ YNQ Accept Request Reject Suggest Eval ReqPerm Offer Promise
Multiple Discourse Units	Argumentation	Elaborate Summarize Clarify Q&A Convince Find-Plan

Figure 4: Conversation Act Types

- Each utterance encodes a single speech act.

Traum and Hinkelman claim that usually conversations are structured to deal with frequent misunderstandings. The assumption that an utterance has been heard and understood is not held until there is a explicit signal given by the listener. The signal can be a explicit acknowledgment, like a *backchannel response*—"okay", "right", "uh huh"—, it can be a linguistic action taken by the listener, such as providing a relevant response—the second part in an adjacency pair—, or it can be a physical signal —head nodding, continued eye contact.

The main difference in their approach is the lack of embedding in different levels. Their classification is not in ranks, where a higher-level unit is composed of lower-level ones—such is the case in syntax, with words, phrases, clauses, and sentences—, but in levels. The levels represent coordination of different types of activity. Turn-taking coordinates who is in immediate control of the speaking channel; grounding coordinates the state of mutual understanding on what is being contributed; argumentation coordinates the higher discourse purposes that the agents have for engaging in the conversation; and core speech acts coordinate the local flow of changes in belief, intentions, and obligations. Traum and Hinkelman's classification is summarized in Figure 4, taken from [Traum and Hinkelman 92].

4.3 Computational Discourse Processing

On the computational side of discourse processing I will start by looking at the MINDS system, one of the first speech projects where contextual information was used. A great deal of information will be gathered from the experience of applying discourse information to one of the translation modules within JANUS, the module based on the GLR* parser. In the same line of analyzing spoken task-oriented dialogues is the Verbmobil system.

4.3.1 MINDS

Young [Young 91] describes the use of domain semantics, dialogue, communication conventions, and problem-solving behavior to enhance automatic speech recognition and understanding. The approach in this project is not one of *late disambiguation*, but of *prediction and restriction*. The system uses knowledge to guide and influence the actual recognition of utterances from an input signal, as opposed to using knowledge to either correct misrecognitions or to select among alternate possible word strings.

The method consisted of continually and reliably modifying the search space—dynamically reducing the space so as to limit the alternatives the recognition system should consider. The pruning of the search space eliminates utterances that would not make sense in a given context, even though they may be reasonable in the overall domain.

A number of heuristics were developed to use the knowledge predictively. For example, using dialogue history, the system would not expect a speaker to ask a question already answered. Particular emphasis was placed on inferring and tracking progress through domain plans and goals. Knowing what had already been accomplished, the various active plans and goals, and the structure of possible domain plans enabled prediction of what could reasonably follow. Knowledge of communication and discourse conventions was used to narrow the possible ways the predicted information could be expressed — replying to a question with an answer, a clarification, or a correction.

The basic processing loop was:

1. Input the speech signal and the predicted nets and lexicon to the speech recognizer.
2. Recognize words in the speech signal.
3. Understand what was said.
4. Generate, process and output a database query.
5. Update models of discourse, plans and history using input from both the input utterance and any system response.
6. Generate predictions and translate them into associated grammar subnets and lexicons for input to 1.

Layered set of predictions were generated after each new utterance was spoken, being used to process the next utterance. To ensure the system would still perform and continue to generate appropriate predictions even in the case of unexpected input, the predictions were layered, allowing

a less restrictive search space to be considered when the evaluation of the 'goodness' of the input match proved to be poor.

4.3.2 Discourse Processing for GLR*

The GLR* discourse processor [Rosé et al. 95, Qu et al. 96a, Qu et al. 96b] was developed in order to reduce ambiguity and improve translation accuracy. The first reports on the discourse processor refer to the use of both statistical methods and plan inference [Levin et al 95], processing multiple hypotheses into the system, rather than passing one hypothesis from one component (the parser, for instance) into the next (the discourse processor), although in some cases, and to avoid an unmanageable number of hypothesis, local pruning is performed.

The discourse module is based on Lambert's tripartite model [Lambert and Carberry 92], [Lambert 93]. It disambiguates the speech act of each sentence, normalizes temporal expressions and incorporates the sentence into a discourse plan tree, at the same time updating a calendar to keep track of the dates being discussed by the speakers. The final disambiguation combines a series of scores from different sources: speech recognizer, parser, and discourse processor.

Plan inference in the discourse processor starts from the surface form of the sentences, from which speech acts are inferred. The surface form is represented in ILTs (InterLingua Texts), with the possibility of multiple speech acts assigned to one ILT, each one with a separate inference chain. The preference for picking one is determined by focusing heuristics, which provide ordered expectations of discourse actions given the existing plan tree.

The second approach taken in GLR* is the use of a finite state processor that describes representative sequences of speech acts in the scheduling domain. The states of the finite state machine represent possible speech acts in the domain, and the transitions include speaker information (whether the speaker changes or stays the same). The processor is used to record the standard dialogue flow and to check whether the predicted speech act follows idealized dialogue act sequences. When there is more than one possible transition the prediction is based on statistical methods. The finite state machine approach proved to be more reliable against cumulative error, produced when an incorrect hypothesis is chosen and incorporated in the model that serves as a basis for the subsequent predictions. In the case of a conflict between the predictions of the finite state machine and the parser's predictions for a speech act, the parser is chosen, which correspond to a *jump* from one state to another, unrelated one.

The plan-based approach handles knowledge-intensive tasks, whereas the finite state approach provides a fast and efficient alternative. Discourse processing is performed by means of these two different approaches separatedly, although there is work under progress on combining them in a layered architecture, with the finite state machine as a lower-level processor for the assignment of speech acts, and the plan system as a higher-level, knowledge-intensive processor for subdialogue recognition and robust ellipsis resolution.

4.3.3 Verbmobil Project

Verbmobil is another recent example of discourse processing for a limited domain. In Verbmobil—which deals with scheduling dialogues aswell—the system acts as a backup for a conversation carried in English by two non-native speakers of the language. The users activate Verbmobil only

when their fluency does not allow them to carry on with the conversation. For that reason the processing done when the system is not active is only a *shallow processing*, with a keyword spotter that permits a partial follow-up of the conversation. The dialogue module in Verbmobil has three subcomponents, as described in [Reithinger 95]:

- A Statistic Module that predicts the following speech act using frequencies drawn from a corpus.
- A Finite State Machine that describes the sequence of admissible speech acts for the domain.
- A Planner that processes speech acts making use of contextual knowledge.

The dialogue memory builds a tree of the structure of the dialogue as it takes place chronologically, and updates a conceptual structure which keeps track of the temporal expressions mentioned by the participants. One of the problems of such a system is that it does not carry out a *deep processing* of the whole dialogue, using a keyword spotter to follow the conversation when Verbmobil is not active [Maier 96]. The use of keywords is fragile, especially when dealing with indirect speech acts, because indirect speech acts require an abstraction from the surface form towards more semantic and pragmatic considerations.

The finite state machine is used to predict the transitions from the current dialogue act to the next. There are seventeen dialogue acts altogether, and the average number of predictions in any state of the main network is five, increased with other five dialogue acts that can occur anywhere—clarifications and digressions, mainly. Since the prediction of ten from a total number of seventeen is not restrictive enough, a statistical prediction method is used. The statistical module makes use of deleted interpolation to compute the probability of a sequence of dialogue acts to narrow down the analysis search [Reithinger and Maier 95].

As reported in [Reithinger 96], the most significant results are obtained when the dialogue acts are annotated with information on the speaker: a reject by Speaker A to a question posed by A might be a clarification, explanation or correction; a reject by Speaker B would be a rejection of the content of A's question. Thus, speech acts are duplicated: `reject` is replaced with `reject-ab` and `reject-ba`, depending on the direction. Once speaker information is integrated in the system, the corpus can be doubled by *mirroring* each dialogue with a counterpart where the speaker's roles are reversed.

Recently, Verbmobil has extended their dialogue processing in order to cover both global and local information, combining their approaches [Alexandersson et al. 97]. The approach is to use the planner to extract information of the global structure of the dialogue, whereas the finite state machine is used to predict the next speech act.

5 Contribution

This project addresses the difficult problem of choosing the most appropriate semantic parse for any given input. The approach is to combine discourse information with the output of the Phoenix parser, a set of possible parses for an input string. The new discourse module interacts with the parser, selecting one of these possibilities. The decision is based on the information provided by the previous discourse context together with pragmatic considerations, such as the structure of adjacency pairs [Schegloff and Sacks 73] and the responses to speech functions [Halliday 94, Martin 92]. The context module keeps a history of the conversation, which will be able to estimate, for instance, the likelihood of a greeting once the opening phase of the conversation is over. A more local history determines the expected second part in any adjacency pair, such as a question-answer sequence. The approach is that of *late disambiguation*, where as much information as possible is collected before proceeding on to disambiguation, rather than restricting the parser's search earlier on. The two-layered approach permits to take into account both global and local considerations, which are integrated in the same module. This provides a robust and efficient processing, thus avoiding the expensive planning architecture for the global structure, as used in the Verbmobil system and intended for the GLR* processing system (Section 4.3).

The discourse module interacts with other modules within the overall system, of which many did not need to be modified to incorporate the output of this module. (For a picture of the discourse component, see Figure 5). The module is able to operate both on output from the speech recognizer and on transcribed data. Transcriptions of dialogues are used for development purposes, in order to avoid possible errors introduced by the speech recognizer. For this project I explored a new dialogue domain, generally named *travel planning domain*. This domain consists of dialogues where a customer makes travel arrangements with a travel agent or a hotel clerk, in order to book hotel rooms, flights or other forms of transportation. They are *task-oriented dialogues*, in which the speakers have specific goals of carrying out a task that involves the exchange of both information and services. Since the dialogue domain is a new one for the JANUS system, part of the work needed involved writing the appropriate grammars and deciding on the set of speech acts for this domain. The task is thus divided in four different areas: (i) selection of appropriate speech acts, (ii) parsing and generation grammars, (iii) coding of the discourse module, and (iv) coding of a training corpus to obtain probabilities.

5.1 Speech Act Taxonomy

The selection of speech acts is a very important part of the project, since we are relying on their specificity or generality to produce the appropriate predictions. For the selection of speech acts, I took several taxonomies into account. The classical Searle [Searle 79] classification of speech acts was slightly adapted to deal with negotiation dialogues, as in [Halliday 94] and [Martin 92]. In the computational arena, both the Enthusiast [Rosé and Qu 96] and the Verbmobil classification [Jekat et al. 95] provide a very extensive set of speech acts already tested and evaluated as regards their efficiency in machine translation. Some effort was made to use the classification derived from a joint effort by the Discourse Resource Initiative [Allen and Core 97], which attempts to develop a standard classification scheme for discourse processing. Unfortunately, the annotation scheme is still under continuous revisions, preventing thus the total compatibility with my taxonomy.

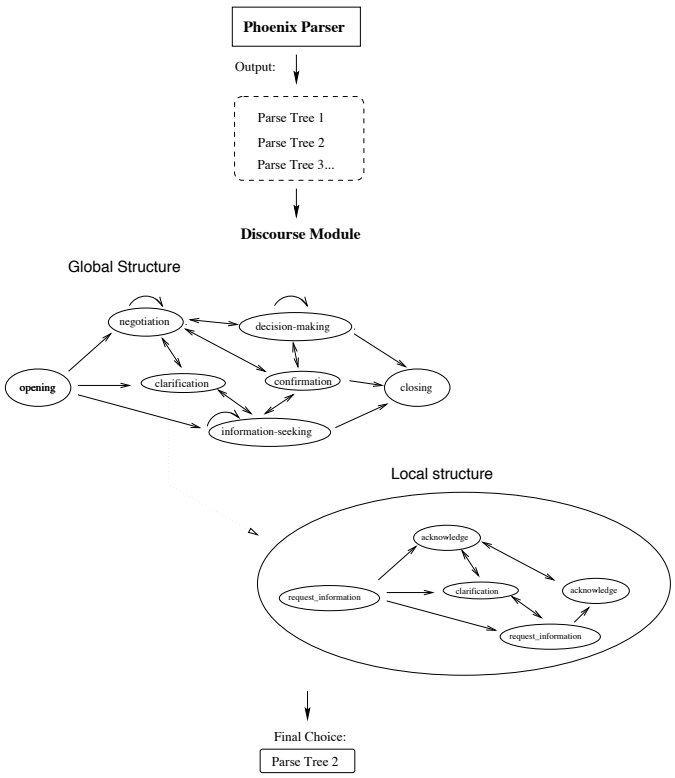

Figure 5: **The Discourse Module**

Since the finite state machine is divided in two layers, the speech acts are grouped according to the subdialogue in which they can occur. A set of 6 subdialogues and 14 speech acts seemed the most appropriate for this domain, as listed in table 6. For a complete description of both subdialogues and speech acts, see Appendices A and B.

5.2 Parsing and Generation Grammars

The selected speech acts are encoded in the grammar, in the Phoenix case a semantic grammar, the tokens of which are concepts that the segment in question represents. Any utterance is divided in SDUs—Semantic Dialogue Units—which are fed to the parser one at a time. SDUs represent a full concept, expression or thought, but not necessarily a complete grammatical sentence. Let us take an example input and a possible parse for it:

(7) Speaker 1 Could you tell me the prices at the Holiday Inn?
```
([request] (COULD YOU
([request_info] (TELL ME
```

Subdialogue	Speech Act
opening	greeting offer-help identify-self affirm acknowledge
state-problem	provide-info request-info acknowledge affirm
information-seeking-21	request-info provide-info acknowledge affirm accept
information-seeking-12	request-info provide-info negate acknowledge affirm
confirmation	request-confirmation affirm
closing	farewell thank promise acknowledge

Figure 6: Subdialogue and Speech Act Taxonomy

```
([price_info] (THE PRICES
([establishment] (AT THE
([establishment_name] (HOLIDAY INN))))))))))
```

The top-level concepts of the grammar are speech acts themselves, the ones immediately after are usually further refinements of the speech act and the lower level concepts capture the specifics of the utterance, such as the name of the hotel in the above example.

The set of speech acts determines the structure of the grammar at its highest level, leaving the depth of the parse tree to be either a refinement of the speech act or a more domain-specific concept. The generality of the speech act classification determines the speech act to follow in an adjacency pair: a request might be followed by some sort of response. The more detailed description of the speech act type leads to consider more specific options, response-offer in example (8), response-statement in example (9):

(8) Speaker 1 Can you tell me the prices at the Holiday Inn?
Speaker 2 Sure I can.

(9) Speaker 1 Can you tell me the prices at the Holiday Inn?
Speaker 2 It's one twenty five for a double, and one ten for a single.

Again, as we travel further down in the parse tree, we get more information on what we can expect from the following utterance. The fact that the previous speaker has asked for prices in a hotel allows us to determine that the figures in the response refer to prices for rooms in a hotel.

5.3 Discourse Module

The discourse module processes the global and local structure of the dialogue in two different layers. The first one is a general organization of the dialogue's subparts; the layer under that processes the possible sequence of speech acts in a subpart. The assumption is that negotiation dialogues developed in a fixed way—the same assumption was made for scheduling dialogues in the Verbmobil project [Maier 96]—, with three clear phases: *initialization*, *negotiation* and *closing*. We will call the middle phase in our dialogues the *task performance phase*, since it is not always a negotiation *per se*. Within the task performance phase very many subdialogues can take place—such as information-seeking, decision-making, payment, clarification, etc. Those are not usually produced in such a predictable order. Figure 7 outlines the structure of a typical dialogue, with the three main phases, some subdialogues within `negotiation`, and a few representative speech acts for each subdialogue.

Discourse processing has frequently made use of sequences of speech acts as they occur in the dialogue, through bigram probabilites of occurrences or through modelling in a finite state machine. However, if we only take into account the speech act for the previous segment in order to pick a correct parse for the current one, we might have insufficient information to decide—as is the case in some elliptical utterances which do not follow a strict adjacency pair sequence:

(10) (*talking about flight times...*)
Speaker 1 I can give you the arrival time. Do you have that information already?
Speaker 2 No, I don't.
Speaker 1 It's twelve fifteen.

If we are in parsing the segment "it's twelve fifteen", and our only source of information is the previous segment, "no, I don't", there is no way we can find out what is the referent for "twelve fifteen", unless we know we are in a subdialogue discussing flight times, and arrival times have been previously mentioned.

My approach aims at obtaining information both from the subdialogue structure and the speech act sequence by modelling the global structure of the dialogue with a finite state machine, with `opening` and `closing` as initial and final states and other possible subdialogues in the intervening states. Each one of those states contains a finite state machine itself, which determines the allowed speech acts in a given subdialogue and their sequence. The gains from such an approach are: first, the constraints imposed on the possible speech acts in a subdialogue—disallowing a greeting interpretation if we are not in the greeting phase of the dialogue—, and second, the information

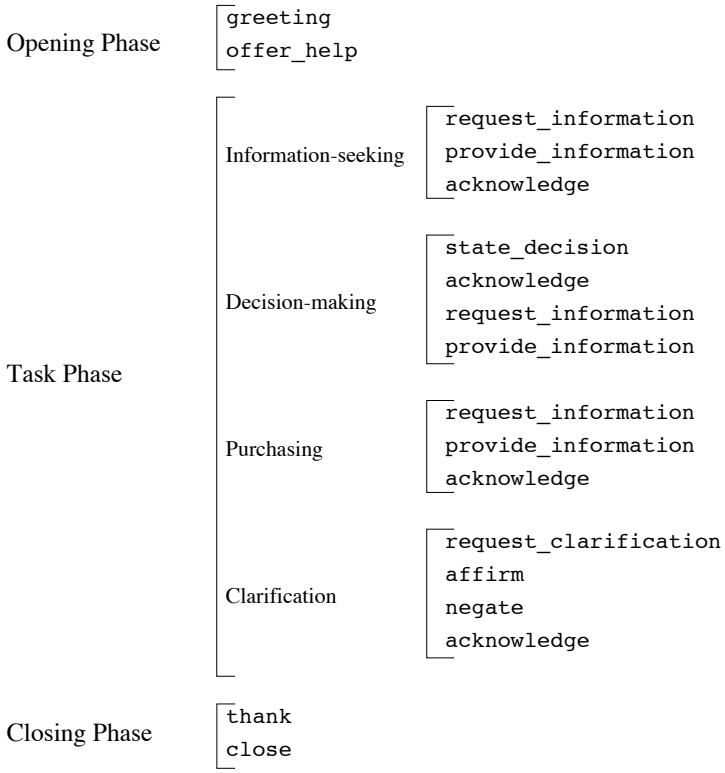

Figure 7: **Structure of a Typical Dialogue**

obtained to process ambiguous parses—deciding that the figure "twelve fifteen" is a flight time if we are in a subdialogue where there is a negotiation for an appropriate flight time.

The discourse component takes as its input the output of the parser. It does not exactly interact with it, because it does not provide any information for the parser in its search for an appropriate parse for a given segment. The only interaction anticipated will be in the case where the discourse module is unable to make a choice among the parses returned, or when its prediction is very different from the available choices. That situation can come about because there is a previous error in the information the discourse module contains—cumulative error [Qu et al. 96a]—, or because the input from the dialogue could not be possibly expected in a "normal" conversation. Given that situation, the two possible solutions are either to leave the parser heuristics decide or have a statistical-based probability for the input, extracted from a corpus. The next sections describe the workings of the finite state machine in more detail.

5.3.1 Inputs and Outputs of the Discourse Processor

The discourse processor takes as inputs the parses produced by the Phoenix parser. Those are sequences of concepts to which the input tokens are matched, as in example (11).

(11) Input:
Could you tell me the prices at the Holiday Inn?

```
([request] (COULD YOU
([request-info] (TELL ME
([price-info] (THE PRICES
([establishment] (AT THE
([establishment-name] (HOLIDAY INN))))))))))
```

The input string might be ambiguous with respect to the speech act it represents. Thus, in the following examples, "okay" represents three different speech acts, namely a prompt for an answer (12), an acceptance of a previous offer (13) or a backchanneling element, i.e., an acknowledgement that the previous speaker's utterance has been understood (14).

(12) S1 So we'll switch you to a double room, okay?

(13) S1 So we'll switch you to a double room.
S2 Okay.

(14) S1 The double room is $90 a night.
S2 Okay, and how much is a single room?

In such cases, the parser will return different speech acts for that same input, as in the examples below.

(15) `[prompt] ( OKAY )`

(16) `[accept] ( OKAY )`

(17) `[acknowledge] ( OKAY )`

Once these three possibilities have been fed into the discourse processor, the context will determine which one to choose, and that will be the output. Presently, the discourse processor does not admit the whole parse tree as input, only the top-level speech act. Below there are two typical input-output sequences, one where the speech act is ambiguous (18), and one where there is no ambiguity (19).

(18) Input string: OKAY
Parser's output:

```
[prompt] ( OKAY )
[accept] ( OKAY )
[acknowledge] ( OKAY )
```

Input to the discourse processor:

```
[prompt],[accept],[acknowledge]
```

Output of the discourse processor (dependent on context):

```
[accept]
```

Output sent to the generator:

```
[accept] ( OKAY )
```

(19) Input string: CAN YOU GIVE ME YOUR CREDIT CARD NUMBER
Parser's output:

```
[request] ([request-info] ([cc-number] ( CAN YOU GIVE ME YOUR CREDIT CARD NUMBER )))
```

Input to the discourse processor:

```
[request]
```

Output of the discourse processor:

```
[request]
```

Output sent to the generator:

```
[request] ([request-info] ([cc-number] ( CAN YOU GIVE ME YOUR CREDIT CARD NUMBER )))
```

5.3.2 The Finite State Machine Grammar

The finite state machine grammar is a representation of the states and transitions in the finite state machine, as in Figure 5. It defines the sequence of possible subdialogues and speech acts, together with their possible adjacencies. Thus, a state in the finite state machine is defined through the subdialogue where it appears plus the speech act it represents. Example (20) shows an example of the speech acts contained in the `opening` subdialogue.

(20)
```
opening
zero
opening
identify_self
opening
offer_help
opening
greeting
opening
affirm
opening
acknowledge
```

For each one of these states there is a possible sequence of follow-up states, again represented by a subdialogue-speech act combination, plus information on whether the follow-up happens when the speaker is the same or when the speaker changes. The `opening -- identify-self` state can be followed by either an `opening -- acknowledge`, if the speaker changes, represented here by a 1, or it can be followed by an `opening -- offer-help`, when the speaker is the same, represented by a 0.

(21)
```
opening
identify-self
opening
acknowledge
1
opening
identify-self
opening
offer-help
0
```

5.3.3 The Algorithm

The discourse module processes one conversation at a time, represented in the top-level speech acts of the parses the Phoenix parser produces. For each speech act there are two main possible situations: that the speech act is unambiguous or that it is ambiguous, i.e., there is more than one possibility to choose from.

1. Unambiguous Speech Act

   In the situation where the parser returns just one speech act for the input string, the discourse processor might find that the parse matches one of the possibilities for next speech act given the current state or that there is no match between finite state grammar prediction and parser output, as seen in examples (22) and (23).

   (22) Parser's output:

   ```
   [provide_info]
   ```

   Discourse Processor state:

   ```
   Current state: request_info
   Possible next states: provide_info, acknowledge
   Match found!
   ```

   (23) Parser's output:

   ```
   [provide_info]
   ```

   Discourse Processor state:

```
Current state: greeting
Possible next states: greeting
No match found, jump
```

In the first example, the speech act `provide_info` is one of two the finite state grammar predicts. Since there is a match, we add `provide_info` to the conversation history and proceed. In (23), there is no match of the finite state grammar predictions and the parser's output. In that case, there is a jump from the current state of the finite state machine to another, unpredicted state.

2. Ambiguous Speech Act

   An ambiguous speech act is encountered when the parser returns more than one possible top-level speech act for the same input string. At this point we have three possibilities:

   (a) One parse matches only one possibility.

   (b) More than one parse matches more than one possibility.

   (c) There is no match between possibilities and parses.

   (a) **One parse matches only one possibility**
   This is the simplest case. As illustrated in example (24) below, the discourse processor adds the matching parse to the conversation history.

   (24) Parser's output:

   ```
   [provide_info],[request_info]
   ```

   Discourse Processor state:

   ```
   Current state: request_info
   Possible next states: provide_info, acknowledge
   Match found! We added provide_info to the conversation history
   ```

   (b) **More than one parse matches more than one possibility**
   In this case, the finite state grammar itself does not solve the ambiguity. We resort to probabilities of speech acts and choose accordingly.

   (25) Parser's output:

   ```
   [provide_info],[request_info]
   ```

   Discourse Processor state:

   ```
   Current state: request_info
   Possible next states: provide_info, request_info, acknowledge
   Both parses were possible. Looking up probabilities...
   Speech act provide_info added to history
   ```

(c) **There is no match between possibilities and parses**

As in the previous case, we need to consult the probabilities to decide among the possible speech acts, in this case because none of them were predicted in the finite state grammar. Once we have chosen a speech act according to probabilities, we have also performed a jump, since we did not follow the sequence predicted by the finite state machine.

(26) Parser's output:

```
[provide_info],[request_info]
```

Discourse Processor state:

```
Current state: request_confirmation
Possible next states: provide_confirmation, affirm, negate
No match found. Looking up probabilities...
Speech act provide_info added to history. Jump
```

5.4 Coding of the Training Corpus

As explained in the previous section, there are situations where the path followed in the two layers of the structure does not match the parse possibility we are trying to accept or reject. In those cases, the transition is determined by unigram probabilities of the occurence of the speech act we are trying to disambiguate. To obtain those probabilities, a corpus of 29 dialogues, totalling 1,344 utterances and over 2,500 speech acts, was coded with speech acts. The probabilites were smoothed using a standard absolute discounting algorithm.

When the discourse processor needs to look up the probabilities it just opens a file that contains the name of the speech act plus its corresponding probability, an example is shown in (27)

(27)
```
provide_info 0.917
request_info 0.268
```

	Without Discourse Processor	With Discourse Processor
Perfect	20.72 %	24.09 %
OK	30.31 %	26.94 %
Bad	48.96 %	48.96 %

Figure 8: End-to-End Evaluation

6 Evaluation

The evaluation of the discourse module was performed on unseen data, consisting of 5 dialogues that the system had not used before for training or testing purposes. There was a total of 228 utterances and 398 speech acts, translated from English into Spanish. There were two different evaluations, which followed two related criteria. A first evaluation was based on the overall improvement in translation, that is, an end-to-end evaluation. Since the module can be either incorporated into the system, or turned off, the evaluation shows the system's performance with and without the discourse module. For such purposes, two tests are performed.

A first test determines the translation quality based on a comparison of the input and the output, with the assignment of a grade to the translation: "perfect" for translations where the content and the illocutionary force were conveyed in a perfect way in the target language—Spanish; "okay" for translations that conveyed the content, but which were awkward or not completely perfect; and "bad" for SDUs that were not translated at all or for translations that would not be understood. As all evaluations of the machine translation components in JANUS, the grading was done by independent graders, with native fluency in the input language and a native or near-native fluency in the output language. In this test, the disambiguation process follows simple heuristics incorporated into the parser. If there are more than one possibilities, the parser picks the one that skipped less words of the input and has a shallower parse tree.

The second test includes the discourse module, and uses that module to pick among the possibilities that the parser returns. The idea behind this evaluation is that since the choice of parse will be more appropriate for the situation, the translation will also be more contextually accurate. The results are shown in Figure 8.

As the results show, there is no improvement in translation quality for translations that were "bad" in the first place. That comes as no surprise, since the discourse module could not possibly improve faulty speech acts returned by the parser. However, when we look at the "perfect" and "okay" grades, we can see an increase in perfect translations when the discourse processor is introduced. The increase is produced by a shift of about four percentage points from the "okay" to the "perfect" category, which means that the translation accuracy was, in fact, improved by choosing the right speech act in the right context.

The number of "bad" translations was, however, very high. This is due to the poor performance of the grammar itself, because we are dealing with a very varied and wide domain, with lower levels of predictability in what the speakers might say. The grammar had too few development dialogues to train on, and thus the parses returned by the parser were not too informative in the first place. Since I felt that the first evaluation did not provide enough information with respect

Total number of speech acts	Ambiguous cases	Successfully disambiguated
398	108 (27.1 %)	73 (67.6 %)

Figure 9: Ambiguity Evaluation

to the performance of the discourse processor, a second type of evaluation was performed on the parsing side, looking at the ambiguous cases returned by the parser, and at how many of those were resolved by the discourse processor. The idea behind this evaluation is to abstract away from the performance of the grammar and the parser, to focus on the performance of the discourse processor in isolation.

The procedure was to manually annotate each SDU with the possible speech acts it could fill, and to let the discourse processor choose among them. Then, the choice was checked against the input, to decide whether the discourse processor had made an appropriate choice. The same 5 dialogues were used in this evaluation. Figure 9 shows the number of ambiguous cases, and how many of them were successfully disambiguated.

The results here show that the discourse processor can resolve 67.6% of the ambiguous cases it encountered, which would eventually yield a better translation in all those cases.

7 Conclusion

In this paper I have presented a model of dialogue structure in two layers, which processes the sequence of subdialogues and speech acts in task-oriented dialogues in order to select the most appropriate from the ambiguous parses returned by the Phoenix parser. The model structures dialogue in two levels of finite state machines, with the final goal of improving translation quality.

The discourse processor handles global and local structures within a single component, a finite state machine, that can easily be modified and extended to cover different types of dialogues. Its compactness in a single component avoids the cost of exchanging information between two systems to deal with global and local structure. The system is robust with respect to unpredicted input, since it can jump to a different place in the finite state machine, creating a new context for the unexpected input.

A possible extension to the work here described would be to generalize the two-layer model to other, less homogeneous domains. The use of statistical information in different parts of the processing, such as the arcs of the FSM—which would be the equivalent of bigram probabilities—, could enhance performance.

8 Acknowledgements

My stay at Carnegie Mellon was made possible by a grant by "la Caixa" Fellowship Program. My work in this project is part of the JANUS Project, funded by different agencies in the United States, Germany, and Japan. I am also indebted to my advisors and fellow students for helpful comments and suggestions.

A Subdialogues in the English Travel Planning Task

A corpus analysis of 19 dialogues in the English Travel Planning Task—dialogues where a client calls a travel agent to ask for information on a particular trip, make reservations or cancellations—has yielded the following preliminary classification of subdialogues:

- Opening
- Problem_statement
- Information_seeking
- Confirmation
- Clarification
- Closing

Except for Closing and Opening, the rest of the subdialogues are annotated with the speaker-hearer ordering[4], thus Information_seeking-12 involves the travel agent asking a question to the customer, such as her name[5], whereas Information_seeking-21 involves a question from the customer, such as the location of the hotels she is inquiring about. In the following, I will describe in more detail the typical contents of these dialogues and the usual exchanges they involve.

A.1 Opening

The number of exchanges here is very limited, usually two or three, starting with an `identification` speech act on the part of the travel agent, which acts as a prompt to the customer. The response by the customer usually includes some acknowledgment, plus the actual request for information, or problem statement. Following is an exchange from a dialogue.

(28) S1 Pittsburgh Travel. This is Rob. How can I help you?
S2 Hi. I'd like to make a trip to Pittsburgh...

A.2 Problem_Statement

Although these subdialogues often do not differ from Information_seeking ones in their format, the reason why I decided to make them an independent category is that they usually occupy a specific position in the dialogue, thus determining the type of exchanges to follow.

Problem-statement subdialogues occur after the opening phase, when the customer states the reason for the call. Whether it is a request for information of a request for action, the content of this phase will determine the possible subdialogues to follow. After a request for information, such

[4] '1' is usually the travel agent, and '2' is the customer, because usually the travel agent iniciates the conversation with a prompt to the customer.

[5] I use the convention of assigning male to the travel agent and female to the customer.

as the prices of rooms in certain hotels, we will expect some providing of information on the part of the travel agent, maybe interleaved with questions about specifics (when does the customer want the rooms, what type of rooms,etc.). After a request for action, we expect immediate clarification questions and information-seeking subdialogues from the travel agent (if it is a cancellation, what kind of reservation it was, the name of the customer, etc.). Another characteristic of these subdialogues is the presence of some background information that the customer provides, such as the dates in both examples (29) and (30).

As stated above, the statement of the problem to be addressed includes both requests for information and for action, as illustrated in the examples below.

(29) S1 Pittsburgh Travel. This is Rob. How can I help you?
S2 Hi. I need to get information. Cause I want to go to Pittsburgh around June seventh, for an Arts Festival down there. I'm in Washington DC. I've just wondering what diferent kinds of options I have.
S1 Well, the best bets < from >[6] into Pittsburgh are to fly USAir or to take a train.

(30) S1 Pittsburgh Travel. How may I help you?
S2 Yeah. I'm going to be making a trip to Pittsburgh. I'll be arriving on the sixth of June. And I'll be leaving in the middle of the day on the twenty third of June, and I would like to make some reservations at the downtown Ramada.
S1 Okay. What's your name?

A.3 Information_seeking

In this category we can include a number of different subdialogues where the speakers exchange information. The direction alternates, between the travel agent and the customer, as to who is the iniciator of the subdialogue, and the subdialogues are consequently labeled *Information_seeking-12* or *Information_seeking-21*.

A.3.1 Information_seeking-12

These refer to all requests for information coming from the travel agent. They involve questions to clarify the request of the customer—preferred dates of travel or types of hotel rooms, as in (31)—or questions necessary to complete the request—name to make the reservation, credit card number or other personal information, as in (32).

(31) S1 Pittsburgh Travel. This is Rob. How can I help you?
S2 Hi. I'm calling to reserve or find out information about hotel and rooms for a trip I'm making down there in about two months.
S1 Okay. When exactly is your trip?
S2 I'll be arriving Pittsburgh on the seventh of June.

[6] In the dialogue annotation, angle brackets indicate a false start.

S1 Okay. And how long < you > will you be staying?
S2 I'm leaving on the morning of the seventeenth Monday.
S1 Okay, so you're looking for a ten night stay. I have information on two hotels in the Pittsburgh area...

(32) S1 Sure, if that's the next < best > best thing. Yeah I'll go for that.
S2 Okay, the total cost for the week then will be seven hundred fifty dollars. Or for your ten day stary will be seven hundred fifty dollars. Can I have your name, please?
S1 The last name is Vaidya, V A I D Y A. And the first name is Anuj, A N U J.
S2 Okay, and how would you like to pay for this?

A.3.2 Information_seeking-21

Excluding the very first request for information that the customer expresses[7], these subdialogues include general requests for information and clarifications of specific points.

(33) S1 Okay, well, at the Ramada we can get you rooms with two double beds and we can also get you suites with two double beds. At the Best Western you pretty much only have one option, which is a room with two double beds.
S2 What is the advantage of a suite over a normal room?
S1 Well, at the Ramada they have suites, which have < a > a bedroom. And then they have the junior suite, which has a kitchen. And then the senior suite, which has < a > a kitchen and a living room.
S2 And what would the regular room be like then at the Ramada Inn?
S1 Well, that's just one room with the beds.

A.4 Confirmation

This is a request for confirmation of some of the information mentioned earlier in the conversation. It is very frequent in the dialogues involving prices, where the customer asks the travel agent for confirmation of the information she just heard.

(34) S1 Okay, at the Ramada, which is downtown, the rates are one fifteen a night for a room with two double beds or one ten a night for a room with one king size bed. You can also get suites for seventy five dollars a night or eighty five dollars a night. And that's a minimum stay of a week.
At the Best Western, which is in Greentree, which is about three miles away, you can get a two double bedroom for eighty four dollars, a one double bedroom for fifty dollars and a one queen size bedroom for sixty five dollars.
S2 So I have one fifteen for two double bed at the Ramada Inn, < and one > and one ten for one king size at the Ramada Inn. Is that right?

[7] In the cases where that happens, because in some other cases it is a request for action.

S1 That's correct.

A.5 Clarification

There is not a very clear distinction between a confirmation and a clarification in many cases.

A.6 Closing

Final part of the conversation. It typically involves a 'thank you' phase, plus a 'goodbye' phase, with maybe a promise to exchange further calls.

(35) S2 Okay, thanks a lot for your information. I'll give you a call back if I'm ready to make a reservation.
S1 Okay, thank you very much. Thanks for calling.

B Speech Acts in the English Travel Planning Task

The set of speech acts consists of the following.

- Identification
- Offer_help
- Greeting
- Provide_info
- Request_info
- Acknowledge
- Affirm
- Promise
- Thank
- Farewell
- Negate
- Request_confirmation

B.1 Identification

The speaker identifies himself or herself. It is usually the travel agenty who identifies the company name and his name. This speech act, in the corpus analyzed, always appears in the opening subdialogues.

(36) S1 Pittsburgh Travel. This is Rob.

(37) S1 This Pittsburgh Travel Agent.

B.2 Offer_help

As the previous one, this speech act appears mostly in the opening phase of the conversation, where the travel agent prompts the speaker to determine the reason of her call.

(38) S1 How can I help you?

B.3 Greeting

Just a regular greeting, also happening during the opening phase. It can be uttered by both speakers.

(39) S2 Hi

B.4 Provide_info

Under this label we could include most of the speech acts in the negotiation phase of the dialogues, and it can be uttered by any of the speakers. It is usually followed by some acknowledgment ('okay', 'alright'), but it could also be followed by another request for information (42).

(40) S1 Every day flights leave DC at seven thirty, ten o'clock, twelve twenty five, three fifty nine, five fifty nine, and seven forty five pm.
S2 Okay...

(41) S2 Sure. It's Sondra S O N D R A Ahlen A H L E N.
S1 Alright.

(42) S1 Well < the shuttle will be > the last shuttle that you'd be interested in would be getting back to the airport approximately ten after five.
S2 And what time does it leave the hotel?

B.5 Request_info

As the previous one, this speech act can appear in many of the different subdialogues. It does not always have the surface form of a question (45)

(43) S2 What are the departure times?

(44) S2 Do I need to make reservations for that?

(45) S1 And if I can have a credit card I will reserve that under your name.

B.6 Acknowledge

An acknowledgment commonly follows both a request for information and the providing of information. In the case of the request, it is often present before the information is delivered (47), or before a further clarification is sought (47). In the case of the providing of information, it seems to serve a more conventional purpose, to mark the switching to a different topic or question (48).

(46) S1 And I was wondering if you could give me information on what kind of rooms would be available for all of us and what the cost would be.
S2 Okay, well at the Ramada we can get you rooms...

(47) S2 I'd [like] to make a trip to Pittsburgh. I'll be travelling with my family and we're looking for hotel reservations.
S1 Okay, when do you plan on arriving in Pittsburgh?

(48) S1 I'll be arriving Pittsburgh on the seventh of June.
S2 Okay, and how long < you > will you be staying?

B.7 Affirm

A 'yes' answer to a yes/no question.

(49) S2 Do you know what day of the week the sixth is?
S1 Yes, in fact I do. It's a Saturday.

B.8 Promise

The speaker commits to some course of action, like returning a call, or checking some information.

(50) S2 I'll give you a call back if I'm ready to make a reservation.

B.9 Thank

The speaker expresses his or her thanks, usually in an exchange where both speakers utter the speech act in succession. It appears in the closing phase of the dialogue.

(51) S1 Okay, thanks a lot for your information. [...]
S2 Okay, thank you very much. Thanks for calling.

B.10 Farewell

The speaker says goodbye. It is used by both speakers, and always in the closing phase of the conversation.

(52) S1 Alright. Thanks for calling Pittsburgh Travel.
S2 Alright. Goodbye.

B.11 Negate

A 'no' answer to a yes/no question.

(53) S1 Is there anything else we can help you with?
S2 Nope.

B.12 Request_confirmation

Requests a confirmation from the other speaker.

(54) S1 Is that right?

References

[Allen and Core 97] J. Allen and M. Core. Draft of DAMSL: Dialog Act Markup in Several Layers. Draft produced by the Multiparty Discourse Group at the Discourse Research Initiative (DRI) meetings at the University of Pennsylvania and at Schloss Dagstuhl, 1997.

[Alexandersson et al. 97] J. Alexandersson, N. Reithinger and E. Maier. Insights into the Dialogue Processing of VerbMobil. *VerbMobil Technical Report 191,* 1997.

[Austin 62] J. Austin. *How to Do Things with Words*, Harvard University Press, Harvard, 1962 (2nd edition).

[Coulthard and Brazil 92] . M. Coulthard and D. Brazil. Exchange Structure. In M. Coulthard (ed.), *Advances in Spoken Discourse Analysis*, Routledge, New York, 1992.

[Halliday 94] M.A.K. Halliday. *An Introduction to Functional Grammar*, Edward Arnold, London, 1994 (2nd edition).

[Jekat et al. 95] S. Jekat, A. Klein, E. Maier, I. Maleck, M. Mast, J. J. Quantz. Dialogue Acts in Verbmobil, Verbmobil Technical Report, 65, 1995.

[Lambert 93] L. Lambert. *Recognizing Complex Discourse Acts: A Tripartite Plan-Based Model of Dialogue*. PhD Thesis, University of Delaware, 1993.

[Lambert and Carberry 92] L. Lambert and S. Carberry. Modeling negotiation subdialogues. In *Proceedings of 32nd Annual Meeting of the ACL*, 1992.

[Lavie 95] A. Lavie. *A Grammar Based Robust Parser for Spontaneous Speech*, PhD Thesis, Carnegie Mellon University, 1995.

[Lavie et al. 96a] A. Lavie, D. Gates, N. Coccaro, L. Levin. Input Segmentation of Spontaneous Speech in JANUS: A Speech-to-Speech Translation System. In *Proceedings of ECAI 96*, Budapest, Hungary, 1996.

[Lavie et al. 96b] A. Lavie, D. Gates, M. Gavaldà, L. Mayfield, A. Waibel, L. Levin. Multi-lingual Translation of Spontaneously Spoken Language in a Limited Domain, In *Proceedings of COLING 96*, Copenhagen, 1996.

[Lavie and Tomita 93] A. Lavie and M. Tomita. GLR*: An Efficient Noise Skipping Parsing Algorithm for Context Free Grammars. *Proceedings of the Third International Workshop on Parsing Technologies, IWPT 93*, Tilburg, The Netherlands, 1993.

[Levin et al 95] L. Levin, O. Glickman, Y. Qu, D. Gates, A. Lavie, C. P. Rose', C. Van Ess-Dykema, A. Waibel. Using Context in Machine Translation of Spoken Language, In *Proceedings of the Theoretical and Methodological Issues in Machine Translation Conference*, 1995.

[Maier 96] E. Maier. Context Construction as Subtask of Dialogue Processing: The Verbmobil Case. In *Proceedings of the Eleventh Twente Workshop on Language Technology, TWLT 11*, 1996.

[Martin 92] J. Martin. *English Text: System and Structure*, John Benjamins, Philadelphia/Amsterdam, 1992.

[Mayfield et al. 1995] L. Mayfield, M. Gavaldà, Y-H. Seo, B. Suhm, W. Ward, A. Waibel. Parsing Real Input in JANUS: A Concept-Based Approach, In *Proceedings of TMI 95*, 1995.

[Qu et al. 96a] Y. Qu, B. Di Eugenio, A. Lavie, L. Levin, and C. P. Rosé. Minimizing Cumulative Error in Discourse Context, In *Proceedings of ECAI 96*, Budapest, Hungary, 1996.

[Qu et al. 96b] Y. Qu, C. P. Rosé, and B. Di Eugenio. Using Discourse Predictions for Ambiguity Resolution. In *Proceedings of COLING 96*, Copenhagen, Denmark, 1996.

[Reithinger 95] N. Reithinger, E. Maier, J. Alexandersson. Treatment of Incomplete Dialogues in a Speech-to-Speech Translation System, In *Proceedings of the ESCA workshop on Spoken Dialogue Systems*, Denmark, 1995.

[Reithinger 96] N. Reithinger, R. Engel, M. Kipp, M. Klesen. Predicting Dialogue Acts for a Speech-to-Speech Translation System, Verbmobil Technical Report 151, 1996.

[Reithinger and Maier 95] N. Reithinger and E. Maier. Utilizing Statistical Dialogue Act Processing in Verbmobil. In *Proceedings of ACL*, 1995.

[Rosé and Qu 96] C. P. Rosé and Y. Qu. Discourse Information for Disambiguation, Paper submitted to *Computational Linguistics.* Carnegie Mellon University, 1996.

[Rosé et al. 95] C. P. Rosé, B. Di Eugenio, L. Levin, and C. Van Ess-Dykema. Discourse Processing of Dialogues with Multiple Threads, In *Proceedings of ACL 95*, Boston, MA, 1995.

[Sacks n.d.] H. Sacks. Aspects of the sequential organization of conversation. Unpublished MS, no date.

[Schegloff 72] E. Schegloff. Notes on a conversational practice: formulating place. In Sudnow, D (ed.) *Studies in Social Interaction.* Free Press, New York, 1972.

[Schegloff and Sacks 73] E. A. Schegloff and H. Sacks. Opening up Closings, *Semiotica 7*, 289-327, 1973.

[Searle 79] J. Searle. A Taxonomy of Illocutionary Acts. Reprinted in A. Martinich (ed.) *The Philosophy of Language*, Oxford University Press, New York, 1996. (3rd edition).

[Sinclair 66] J. Sinclair. Indescribable English. Inaugural Lecture, University of Birmingham, 1966.

[Sinclair and Coulthard 75] J. Sinclair and R. M. Coulthard *Towards an Analysis of Discourse: The English Used by Teachers and Pupils*. Oxford University Press, Oxford, 1975.

[Schmitz and Quantz 95] B. Schmitz and J. Quantz. Dialogue Acts in Automatic Dialogue Interpreting. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, Leuven, 1995, pp. 33-47.

[Traum and Hinkelman 92] D. Traum and E. Hinkelman. Conversation Acts in Task-Oriented Spoken Dialogue. *Computational Intelligence* 8 (3), pp. 575-599, 1992.

[Waibel 96] A. Waibel. Interactive Translation of Conversational Speech. In *IEEE Computer Society*. Volume 29, Number 7. 1996.

[Ward 94] W. Ward. Extracting Information in Spontaneous Speech. In *Proceedings of ICSLP 94*, 1994.

[Ward 91] W. Ward. Understanding Spontaneous Speech: the Phoenix System. In *Proceedings of ICASSP 91*, 1991.

[Young 91] S. Young. Use of Dialogue, Pragmatics and Semantics to Enhance Speech Recognition. *Speech Communication* 9, pp. 551-564, 1991.