# Discourse Segmentation Of Spoken Dialogue: An Empirical Approach

by

Giovanni Flammia

Laurea, Universitá di Roma, La Sapienza (1988)
S.M., McGill University (1991)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1998



Author .....
Department of Electrical Engineering and Computer Science
May 20, 1998

Certified by .....
Victor W. Zue
Senior Research Scientist, Laboratory for Computer Science
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

# Discourse Segmentation Of Spoken Dialogue: An Empirical Approach

by

Giovanni Flammia

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 1998, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

## Abstract

This thesis is an empirical exploration of one aspect of human-to-human telephone conversations that can be applicable to building human-to-machine spoken language systems. Some cognitive and computational models assert that human-to-human dialogue can be modeled as a joint activity decomposable as a sequence of discourse segments. Detecting segment boundaries has potential practical benefits in building spoken language applications. Unfortunately, the discourse structure of spontaneous dialogue can be quite variable. In this thesis, we seek answers to two questions. First, is it possible to obtain consistent annotations of segments from many coders with no specific prior training in discourse analysis? Second, once a corpus has been annotated, what are the regular patterns and irregularities found by analysis of the data?

The contributions of this thesis are twofold. Firstly, we developed and evaluated the performance of a novel annotation tool called *Nb* and associated discourse segmentation instructions. *Nb* and the instructions have proven to be instrumental in obtaining reliable annotations from many subjects. Extensive inter-coder agreement experiments indicate that it is possible to obtain reliable discourse segmentation when the instructions are specific about the task and when the annotation task is limited to choosing among few independent alternatives. Reliability (measured by the kappa coefficient) is competitive with other published work.

Secondly, the analysis of an annotated corpus of information-seeking dialogues provides substantial empirical evidence about the differences between human-to-human conversation and current interactive voice response systems (IVRs) and question-answer systems (QAs). In IVRs and QAs interaction is limited by the design of the system. In natural conversation either speaker can take the initiative *at any time*. The data analysis indicates that even simple information-seeking dialogues have a rich structure. The data support theories of dialogue as a joint activity in which discourse segments are initiated by either speaker with the purpose of either finding a solution to the task at hand or repairing and preventing misunderstandings. In addition, we demonstrate how a stack data structure is sufficient to model segment transitions including preliminaries, repairs, fresh starts and switches between multiple active purposes.

Thesis Supervisor: Victor W. Zue
Title: Senior Research Scientist, Laboratory for Computer Science

# Acknowledgments

This thesis has been completed thanks to the continuous stream of energy, inspiration, enlightenment, support, encouragement, feedback, corrections and arguments provided by Victor Zue, my thesis supervisor. The work reported here is one fruit of his constant and enthusiastic support for conducting empirical research in all aspects of spoken language.

Four contingent factors have determined the choice of discourse and dialogue modeling as a topic. Early discussions with Victor pointed to dialogue modeling as the next frontier of research for developing widely usable spoken language systems. The Wizard-of-Oz studies for Italian Voyager confirmed that developing a conversational system is essentially a problem in dialogue design, and less of a problem in natural language processing or speech recognition. Many frequent enlightening discussions of dialogue strategies with Stephanie Seneff (a member of the thesis committee) and Joe Polifroni clarified issues and provided inspirations for research. At the same time, at Harvard University, Barbara Grosz (another member of the committee) and Christine Nakatani kindly accepted me as a participant in their discussion group about empirical methods in discourse segmentation. Their contributions to the discussion group also provided inspiration for this work. Since then, Barbara has offered much advice and suggestions about research. Jonathan Allen (a member of the committee) also provided technical advice and comments in the series of thesis committee meetings.

The work reported in this thesis has also been inspired and influenced by collaborating with other present and past members of the Spoken Language Systems Group at MIT. Apart from this thesis, I had the honor to contribute to the development of parts of Voyager and Web-Galaxy, two experimental conversational information services. In particular, eight other members of the SLS group have provided technical advice: Raymond Lau, Lee Hetherington, Michael McCandless, Jim Glass, Helen Meng, Karen Livescu, Christine Pao and Drew Halberstadt. I wish to thank Karen for editing this document with patience and skill. I particularly enjoyed Ray's astute and sometimes humorous observations about the complexity of parsing natural language and of predicting trends in high technology.

This thesis would not have been completed in time without the unlimited and constant love, confidence, support and patience of my wife Katy during these past years. In addition, my parents Nino and Angela have provided support and encouragement to conduct research in an international context.

This research was supported financially by DARPA under contracts N66001-94-C-6040 and N66001-96-C-8526 monitored through Naval Command, Control and Ocean Surveillance Center. The text transcriptions of the telephone conversations used for this thesis were kindly provided by BellSouth Intelliventures and American Airlines.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivations

Advances in human language technology enable building conversational applications that are more usable and flexible than simple menu-based interactive voice response systems (IVRs) and question-answer systems. In an IVR application, the system prompts the user to speak specific words or phrases. In a question-answer system, interaction is limited to the system answering a series of direct questions from the user. In contrast, a *conversational* system is designed to conduct a dialogue in which moves can be initiated by either the user or the system, using a large vocabulary and syntactic constructs that are more similar to everyday conversation [112].

Designing conversational applications is challenging for at least three reasons. Firstly, the design should incorporate gracious and quick recovery from the inevitable speech recognition errors and natural language misunderstandings. Secondly, speech is an ephemeral medium. Users can only remember the last few words and phrases of each sentence, so it is impractical for the system to speak long lists of information. Thirdly, the functionality of conversational interfaces is hidden. Unlike graphical user interfaces (GUIs) and IVRs, in which all of the choices are either visible or spoken to the user, it is impractical for conversational systems to list to the user all the words they can use to get the information they want. To overcome these challenges, the design of better spoken applications may be based on analyses of human-to-human dialogues [112, 7]. However, designing user interfaces based on human-to-human interaction is a controversial issue. In a debate between direct manipulation vs. interface agents, B. Schneiderman stated ([95], page 56):

> *I am concerned about the confusion of human and machine capabilities. I make the basic assertion that people are not machines and machines are not people. I do not think that human-to-human interaction is a good model for the design of user interfaces.*

On the other hand, dialogue system designers such as N.Yankelovich have a different opinion ([111], forthcoming):

> *Our experience has shown that natural dialogs can serve as an effective starting point for a speech user interface design. Not only do they help in the design of grammars, feedback, and prompts, but they also point out instances where speech technology cannot be effectively applied.*

Although it is not expected that users will talk to a machine as they would to a fellow human, we believe that a machine that shares some of the human communication conventions may be more usable than a machine which requires the user to learn and remember new speaking conventions by trial and error. After all, many of the conventions of human communication makes it a very efficient medium. According to D. Norman ([75], forthcoming):

> *Human language serves as a good example of the evolution of a robust, redundant, and relatively noise-insensitive means of social communication. Errors are corrected so effortlessly that often neither party is aware of the error or the correction. The communication relies heavily upon a shared knowledge base, intentions, and goals...*

This thesis is an empirical exploration of one aspect of human-to-human telephone conversations that can be applicable to the design of human-to-machine conversational systems. We would like to test the hypothesis set forth by theories in cognitive and computer sciences that natural task-oriented dialogue is a highly structured goal-oriented activity. In particular, some models assume that human-to-human task-oriented dialogue can be modeled as a sequence of related discourse segments. Discourse segments may be initiated by any participant in the conversation with the purpose either of finding a mutually satisfactory solution to a task [39, 37, 57, 58] or of repairing and preventing misunderstanding [21, 22, 23]. An example of a discourse segmentation is displayed in Figure 1-1. Segments are sequences of one or more related communicative acts which accomplish a specific purpose (or goal) in common between the conversation participants. In the figure, segment

| *Segment purposes and text transcription* | *Communicative acts* |
|---|---|
| **Segment 1: List Theater Showing Movie** | |
| 1 C: I was trying to find out what time the Specialist is playing, and where | Request |
| **SubSegment 2: Clarify Location** | |
| 2 A: What part of town? | Request Clarification |
| 3 C: Mansell Crossing. | Clarification |
| 4 A: Okay, it's showing at Mansell Crossing at Northpoint Mall | Inform |
| 5 C: [Uh-huh] | Acknowledgment |
| **Segment 3: List Show Times For Movie** | |
| 6 C: What are the shows after two thirty? | Request |
| 7 A: The next show is at four fifty | Inform |
| 8 C: [Uh-huh] | Acknowledgment |
| 9 A: And then I have six forty and nine fifty | Inform |
| **SubSegment 4:: Confirm Show Times** | |
| 10 C: Was that four fifty? | Request Confirm |
| 11 A: That's correct. | Confirm |
| 12 C: Ok. Thanks. | Closing |

Figure 1-1: Example of an information retrieval dialogue with corresponding segmentation indicated by a horizontal line. Segment purposes are indicated in boldface above the text transcription. To the right of the transcription is the sequence of communicative acts.

purposes are indicated in boldface above the text transcription, and the sequence of communicative acts is indicated to the right of the transcription. In the text transcription, C stands for Customer and A for agent (or operator). Typically, a segment opens with a request from the customer C and contains the information delivered by the agent A. Optionally, a segment may contain one or more clarification sub-dialogues before the delivery of the information, and confirmation and closing sub-dialogues after the delivery. Detecting segment boundaries has many potential practical benefits in building spoken language applications (e.g., audio indexing, designing effective system dialogue strategies for each discourse segment and dynamically changing the system lexicon at segment boundaries).

Unfortunately, drawing conclusions from studying human-to-human conversations may be difficult because spontaneous dialogue can be quite variable, containing frequent interruptions, incomplete sentences and discourse segments structured quite differently from written text and spoken monologue. For example, consider these two exchanges from our corpus of telephone conversations:

| | |
|---|---|
| C: Okay, [ah] what's playing around nine forty?<br>A: [humming]<br>C: Well what's playing period? I mean<br>A: Hey, that'd be a better question<br>C: [Yeah] [Laughter]<br>A: Frankenstein is playing at ten | A: I have it at the Cobb Place Eight<br>C: Is that it?<br>A: They're at Parkway near Highway Forty One<br>C: Is that the only one?<br>C: Is it at Galleria?<br>A: Yes, sir, it's next show time is at [uh] four thirty |

Spontaneous dialogue variabilities make it difficult to hypothesize unambiguous discourse segment boundaries at specific dialogue turns. While the underlying structure may be clearly specified as a hierarchy of topics and goals, the surface linguistic realization may be ambiguous. R. Hopper is among the conversation analysts who warn against the pitfalls of analyzing discourse segments in natural telephone conversations ([48], page 155):

> *How many topics appear in this segment? How do partners accomplish the transition points between topics? Cannot multiple topics be considered at once? At the end of the segment what topic is on the floor? ... These questions illustrate the futility of counting topics, or treating them as displaying clear boundaries between them.*

Because discourse segment structure is such a controversial issue, it is necessary to conduct empirical studies that test linguistic theories against annotated corpora. With the help of properly annotated data, researchers understand may the regular and the variable aspects of the linguistic phenomena under investigation [107]. This thesis takes this approach, by annotating and analyzing hundreds of conversation transcriptions with their discourse segment structure.

For a corpus to be truly useful, it must be properly annotated. Corpus annotation involves defining the inventory of constituent units (e.g., phonemes, syntactic categories, and intentional categories), together with a set of annotation conventions. The annotation units and the conventions form what is called a *coding scheme*. For example, at the syntactic level, pronouns might be annotated along with the definite noun phrases they refer to [19, 44] and at the intentional level, sentences might be annotated with the speaker intentions (e.g. whether the sentence is a request for information, an acknowledgment) [92, 93, 1].

Annotation of some linguistic phenomena such as phonetic variants and disfluencies are relatively straightforward, since agreement on the choices of units and conventions can often be reached [55, 96]. As a result, the task of annotation can often be shared across site, and the aggregate corpora are larger and more useful to a wider community. As we move up the linguistic chain, however, the

picture can rapidly deteriorate. While the study of larger linguistic units (e.g., sentences, discourse segments) necessitates a corresponding increase in the amount of annotated data, this need for resource sharing is difficult to implement in reality. In most cases, the controversy stems from the fact that the choices of units and conventions are often tied to linguistic theories that are not universally subscribed. Therefore, corpora annotated by one site may not be useful to researchers from other sites, leading to duplication of effort and inhibiting cross-system comparisons. One approach to dealing with this problem is to provide a set of minimal, theory-neutral evaluation metrics. The Penn Treebank [64] is an excellent example of linguistic data annotated using this approach. Syntactic structure of sentences is implicitly described by *bracketing* major constituents without actually attaching labels to them. While there were some initial doubts regarding the ultimate utility of such an annotation scheme, they were largely put to rest once researchers had a chance to make use of the corpus. The Penn Treebank has been instrumental in facilitating the comparison of several general English parsers [9, 10]. In this thesis we apply the same paradigm to evaluate agreement between different segmentations of the same text.

## 1.2 Overview of the Corpus

In this section, we briefly present some characteristics of natural dialogues that have been extracted from the corpus used for this thesis. To carry out our research, we are making use of a corpus of 1532 orthographically transcribed telephone conversations. The text data are faithful transcriptions of actual telephone conversations between customers and telephone operators collected by BellSouth Intelliventures and American Airlines. The data have been collected and transcribed in 1994 for quality of service purposes, and not for the purposes of speech technology research. A majority of conversations lasted between one and two minutes, and consisted of between 25 and 45 different dialogue turns (a turn is a set of a few clauses that are spoken by the same speaker without being interrupted).

Figure 1-2 lists five representative initial sections of transcriptions from the corpus. The subjects of the conversations range among five topics: looking up movie schedules and restaurant listings in Atlanta, looking up automobile and job classified ads, and planning for air travel on American Airlines flights. While not shown in the figure, the text transcriptions include punctuation marks, sentence segmentation, speaker change markers and other markers for long pauses, overlapped and interrupted speech, non-speech events, and unclear speech.

The left plot in Figure 1-3 displays the average length of the conversations by topics. The movie listing domain and the restaurant guide domains appear to have the shortest dialogues. The tasks for these two domains are few and the information reported by the agent is relatively short in size (e.g., listing show times for selected movies or locating particular movie theaters and restaurants

| *Movie Schedules* | *Restaurant Guide* |
|---|---|
| C: Is there a [ah] number that you dial to just get all the different theaters?<br>A: I can give you that information.<br>C: You can?<br>C: Okay, in Snellville, Septum Movies.<br>A: Sure, just one moment please...<br>A: And that was the Septum Theater?<br>C: [Yeah].<br>A: Okay, I have a Cineplex Odion in Snellville.<br>... | C: [um] Could you tell me the nearest [um] place, pizza place that<br>C: delivers in, Where we live on?... Powder Springs.<br>A: OK, one moment.<br>C: How much does this phone call cost?<br>A: The first three calls are free.<br>A: After that it's fifty cents a call.<br>C: Oh, OK.<br>A: And you on Powder Springs?<br>... |

| *Flight Booking and Information* | *Automobile Classifieds* |
|---|---|
| C: I'm wondering if you can give me a fare<br>C: from Albuquerque to Detroit on the twenty-seventh of July?<br>A: Okay. I'll sure check for you.<br>C: Thank you.<br>A: Will this be a one-way or round trip for you?<br>C: Uh, round trip–<br>A: Right.<br>C: –Returning the first of August<br>... | C: Could you check something [uh] in the [uh] automobile [uh] used autos section of the classifieds?<br>A: OK, which?<br>C: I'm looking for [um] Do I tell you what kind of car I'm looking for or<br>A: OK, first, do you think it's the one that you're looking for is over or under two thousand?<br>C: Over.<br>... |

| *Job Classifieds* |
|---|
| C: I'm looking for [inhale] employment in the management field.<br>A: OK. And any particular type of management, sir?<br>C: [uh] Retail management.<br>A: OK, just a moment please.<br>A: And you're looking for full-time, sir?<br>C: Yes.<br>A: Just a moment please.<br>A: Let's see what we have in yesterday's paper.<br>C: All right.<br>... |

Figure 1-2: Five representative initial sections from the corpus of telephone conversation transcriptions. In the text, C stands for the customer, and A stands for the telephone operator, or agent.

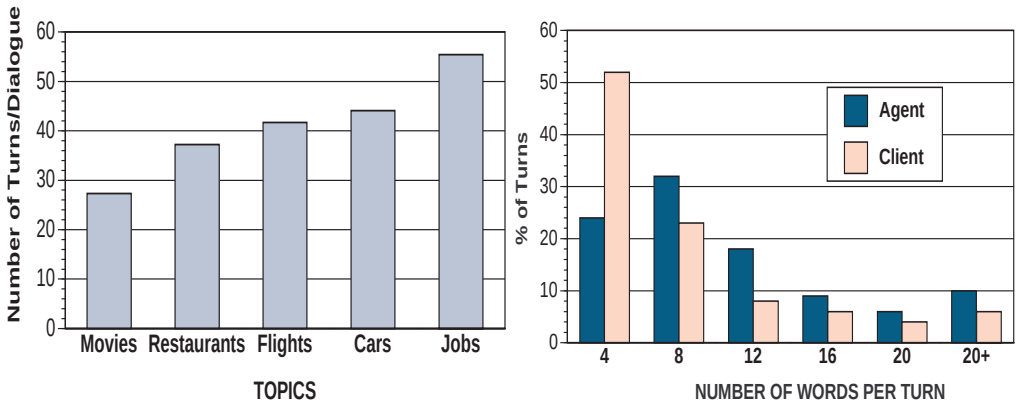



Figure 1-3: Left: A plot of the average number of turns as a function of the topics for over 1000 human-human dialogues. Right: Fraction of turns as a function of dialogue turn length for agents and customers.

based on the location). Flight information, car classifieds and job searches appear to have the longest dialogues. They are also more complex application domains in which the agent may report long list of information. The flight domain involves booking a flight with many possibly conflicting constraints based on dates, availability and fare type. In addition, American Airlines agents can also report flight and ticket status information. The last two domains, used car and job classifieds, often require that the agent report many different ads to the caller. Selecting a car classified depends on several constraints such as make, model, year, asking price, engine type, conditions and optional features. Finally, in the job classified domain the agent may report many ads for different positions and with different requirements before finding one that is appropriate for the caller.

The right plot in Figure 1-3 indicates that the agent speaks in longer turns, and that over 50% of the customers' turns were of 4 words or less. On average, the agent contributions tend to be longer because she is the one who reports lists of information to the customer. Most of the short utterances spoken by the customer are either acknowledgments or confirmations of the information provided by the agent, as in the following example:

A: It is at the Movies at Gwinnett Peachtree Corners and Roswell Mall .
C: Roswell Mall.
A: And the next show time at Roswell would be four thirty five
A: seven oh five and nine twenty five .
C: Seven oh five.
C: Okay.
A: And at Gwinnett the next show times are four fifty seven fifteen and nine forty five .
C: thank you.

The dialogues in the information service corpus share a similar task structure. In all of the dialogues, the customer is the information seeker, and the agent reports the information by accessing

| *Subtask* | *Clarification* | *Diversion* |
|---|---|---|
| A: Will this be a one-way or round trip for you?<br>C: Uh, round trip–<br>A: Right.<br>C: –Returning the first of August.<br>A: First of August. | A: Do you know that there is one called the Septum?<br>C: [Yeah], it's called Septum,<br>C: but Cineplex Odion is probably the theater who owns it, C: but it's called the Septum Theater,<br>C: but [yuh], [yeah] that's it.<br>A: Okay | C: How much does this phone call cost?<br>A: The first three calls are free.<br>A: After that it's fifty cents a call.<br>C: Oh, OK. |

Figure 1-4: Three representative examples of discourse segments.

a database using a graphical user interface. In general, after the initial greetings, the task-specific section of the dialogue is opened by the customer's request for information (see Figure 1-2). The request for information sets the initial purpose, or goal, that motivates the speakers' actions for the remaining sections of the dialogue. The request for information is usually further specified by one or more discourse segments. Typically, the following segments are related to the initial request by being either a *subtask*, a *clarification* of the customer's request, or a *diversion*, as illustrated in Figure 1-4. A subtask segment has the purpose of specifying additional information that is needed by the agent before she can access the database. A clarification segment is needed to set a common ground between the conversation participants and to ensure that they understand each other and are talking about the same topic, such as a location. Finally, the purpose of the dialogue can be momentarily diverted by a digression segment.

When a request is mutually understood and fully specified, the agent can access the database and report only the information of interest to the customer. Reporting the information mat require one or more discourse segments and more than two dialogue turns. Consider the following example:

A: OK, there is a [uh] an ad for Marshall's.
C: [mm-hmm]
A: They're looking for retail managers.
C: Yeah, I've already applied for that one....
A: OK.
A: Family Dollar Stores?
C: I applied for that one.
A: OK.

In the example, the agent reports the information interactively, breaking down the information into one or more short installments, and she proceeds only after the customer has explicitly acknowledged or confirmed the information with an appropriate response.

In summary, the task oriented telephone conversations in our corpus are highly interactive, and quite different from IVRs or question-answer systems. Both the request phase and the response

phase of a database query may require several segments, each one containing many dialogue turns. In particular, the high percentage of confirmations and acknowledgments indicate that setting a common ground between the speakers is paramount in telephone conversations.

## 1.3 Contributions

In the rest of the thesis, we provide detailed quantitative evidence about the discourse structure of the conversations in our corpus, focusing in particular on the movie schedule dialogues. We provide some specific quantitative evidence about the internal turn-taking organization of discourse segments and we discuss what computational devices are appropriate to model sequences of communicative acts within segments and transitions between segments.

We decided to focus primarily on the movie schedule domain because it is within reach of current state-of-the-art spoken dialogue systems. The analysis of this relatively simple domain reveals fundamental differences between human-to-human conversations and interactive voice response systems or question-answer systems. The analyses of the corpus annotated with discourse segment units provide substantial evidence to the assertion that natural human-to-human conversations have a rich discourse structure, even in simple information-seeking application domains. In addition, obtaining reliable annotations of discourse segment units is a controversial issue. We address this problem using a strict incremental approach. Firstly, we would like to annotate reliably simple application domains. Only after that milestone has been reached, we may look at more complex domains, and possibly more complex discourse annotation coding schemes.

The specific objectives of this thesis are to seek answers to three related questions. First, once a corpus of text transcriptions is available for conducting empirical analysis, is it possible to obtain consistent annotations from many coders with no particular prior knowledge of discourse analysis? Second, what are the regular vs. irregular discourse patterns found by the analysis of the annotated conversations? Third, to what extent the annotated data support (or does not support) cognitive and computational theories that view task-oriented dialogues as a highly structured co-operative process?

The contributions of this thesis are threefold. Firstly, we developed and evaluated the performance of a novel annotation tool called *Nb* and associated discourse segmentation instructions. *Nb* has been a pioneering effort in providing the necessary infrastructure for rapid development of discourse annotation coding schemes. The tool and the instructions have proven to be instrumental in obtaining reliable annotations from many subjects. The tool is freely available on-line. It has been designed iteratively over the course of three releases, and has been tested extensively with a variety of discourse annotation coding schemes. It has been used for this thesis as well as for many other discourse segmentation studies in other institutions. *Nb* has proven to be an efficient editing tool

for annotating hierarchical discourse structures in text transcriptions, including discourse segments and their purposes and communicative acts.

Secondly, with *Nb* we conducted four annotation experiments to assess under which condition we can obtain reliable inter-coder agreement in placing discourse segment boundaries in text transcriptions. The dialogue transcriptions have each been annotated by several people, using different annotation instructions. The segmentation task ranged from being unconstrained to being directed. Although unconstrained annotations were less reliable than directed annotations, the analysis of the unconstrained annotations were an essential step in determining effective directives for later experiments. Our findings indicate that it is possible to obtain reliable and efficient discourse segmentation when the instructions are specific about the task and the annotators have few degrees of freedom - i.e., when the annotation task is limited to choosing among few independent alternatives. We have assessed inter-coder agreement using the metrics of precision, recall and the kappa coefficient. The reliability results are competitive with other published work on discourse segmentation [107, 41, 45, 80, 16], and lend empirical support to the notion that discourse segmentation can be done as reliably as annotating other types of discourse units, such as communicative acts. Coders agreed in placing discourse segment boundaries at locations where a speaker would either initiate a new task-related purpose or initiate a clarification or confirmation segment. Coders disagreed in placing segment boundaries at specific locations. These locations mostly corresponded to speech and dialogue repairs and sections with two different purposes active at the same time.

Thirdly, we analyzed in detail the discourse structure of one particular information-seeking domain, the movie schedule task. The data annotated with *Nb* have provided substantial empirical evidence about the discourse segment structure of natural dialogue and the internal organization of segments. The annotated data provide substantial support for the theories of intentional structure of discourse [39, 37, 57, 58] and of contributions to discourse [21, 22, 23] which view dialogue as a co-operative or joint activity driven by the speakers' intentions. The annotated data have also provided the basis for evaluating empirically which computational device is appropriate to process natural dialogues. The data indicate that probabilistic finite state models (i.e., trigrams of communicative acts) may be appropriate to help in predicting the customer's communicative acts, but they lack the data structures necessary to determine the most appropriate agent's responses. On the other hand, we argue that a relatively simple hierarchical data structure (i.e., a stack, as proposed by [35]) is sufficient to model segment purpose switches in natural dialogues, including apparently irregular phenomena such as repairs, fresh starts and multiple concurrent active purposes.

## 1.4 Overview of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, we review some issues in annotating discourse units in conversations, and we define the evaluation metrics (precision, recall and the kappa coefficient) which have been used in this thesis, and we report the state of the art in inter-coder agreement for a variety of discourse coding schemes.

In Chapter 3, we report on how we developed and evaluated the three releases of *Nb* , the annotation tool that we have used for designing and assessing the reliability of discourse segmentation schemes. The first release of *Nb* is also described in [32].

In Chapter 4, we discuss in detail four different discourse segmentation experiments which have involved many different coders, and we report under which condition we have obtained reliable discourse segmentations from many trained coders. This chapter substantially revises and extends some of the results that we reported in [31, 33].

In Chapter 5 we present a case study of conversation analysis which has been made possible by annotating a corpus of 190 different dialogue transcriptions using *Nb* . We have annotated a set of movie schedule dialogues with intentional units such as discourse segments and communicative acts. In this chapter, we analyze turn transitions within segments using a probabilistic finite state model, and segment transitions using a stack model.

In Chapter 6, we summarize the contributions of the thesis, we list some of the open problems which have not been addressed in this work, and we suggest directions for future research in empirical analysis of conversations. In particular, this thesis is focused on the text content of conversations, leaving out issues about intonation contours and timing to future work.

In Appendix A, we demonstrate how to compute the group-wise kappa coefficient for assessing inter-coder agreement in the case of more than two coders annotating more than two categories, with possibly missing data. Finally, in Appendix B we include the full text of the annotation instructions for one of the experiments reported in Chapter 4.

# Chapter 2

# Annotating Discourse Units in Conversations

This thesis analyses the discourse structure that guide what people say when they are engaged in a task-oriented dialogue. A task-oriented dialogue is a dialogue in which the participants co-operate to solve a specific problem, such as reporting movie schedules. Consider the following telephone conversation between a customer C and an agent A:

| | |
|---|---|
| 1 | C: I'm trying to find out where the Lion King is located. |
| 2 | A: The Disney movie? |
| 3 | C: Yes. |
| 4 | A: OK, I don't think it's playing in the theaters anymore. |
| 5 | A: They're supposed to re-release it [um] around the Thanksgiving Holidays. |

Syntactic and semantic analysis is focused primarily on interpreting the grammatical structure and the meaning of phrases and clauses in sentences. For example, a syntactic parser will detect that *the Lion King* is a noun phrase, and semantic analysis would interpret this noun phrase as a specific semantic entity: a movie title. This type of analysis is mainly concerned with determining relations between words in sentences. In contrast, discourse structure is concerned with determining the relations that exist between sentences in text and speech. The type and size of relations depend on the genre of the text. For example, the analysis of a monologue may be different from the analysis of a casual conversation between two friends, and both may be quite different from the analysis of a *task-oriented* conversation between a customer and a telephone operator. In this thesis, we focus on task-oriented spoken dialogue because it is a genre applicable to building spoken language systems. In this case, discourse analysis is mostly focused on determining relations between sentences spoken by different speakers across dialogue turns. In particular, we focus on *intentional* relations (e.g., [1, 38, 39, 92, 93]). For example, in this framework the first sentence of the above dialogue exchange

(*I'm trying to find out where the Lion King is located*) is not interpreted solely as a declarative, but rather as a communicative act: a request for some specific information. Similarly, the fourth sentence spoken by the agent (*I don't think it's playing in the theaters anymore*), is interpreted as the communicative act that responds to such a request. Discourse segment structure is concerned with determining the sequence of communicative acts that took the participants from requesting the information to reporting it.

In empirical studies, discourse structure must be encoded by a set of labels - such as *request for information* - which are attached to one or more clauses in the text transcription. The set of labels and the annotation convention constitute the *coding scheme*, which is an application of a particular discourse theory. Once a reasonably large corpus of transcriptions is annotated with such labels, the data can provide evidence supporting or contradicting hypotheses set forth by theories of discourse. One crucial issue of discourse annotation is whether or not it can be performed reliably by trained coders who do not necessarily have extensive prior knowledge of the discourse theories. Whereas the reliability of annotating phonological, syntactic and intonation units in sentences has been extensively studied [54, 64, 96], a discussion of the issues in annotating discourse units in dialogue has only begun to emerge in last three years [107, 16, 28].

In the rest of this chapter, we define the concept of discourse segment structure and we compare it to other proposed units of discourse analysis. We list the empirical methods that can be applied to assess the inter-coder agreement in annotating discourse units, and we report on the state of the art in assessing how reliably different types of units can be annotated.

## 2.1 The Types of Discourse Units

In this chapter, we consider the discourse units that have been proposed by discourse analysis researchers at various levels of detail. The smallest unit that we consider is the *conversational clause*, or *dialogue move* such as *"The Disney movie?"*. Whereas a clause in text usually contains both a subject and a predicate, in this thesis we define conversational clause (clause for short) as either a full sentence or an elliptical phrase (e.g., a noun phrase or a prepositional phrase) spoken as one cohesive intentional unit. A *dialogue turn* is a set of a few clauses (typically between one and three) that are spoken by the same speaker without being interrupted. For example, the following dialogue turn contains three clauses:

| |
|---|
| A: OK, |
| A: I don't think it's playing in the theaters anymore. |
| A: They're supposed to re-release it [um] around the Thanksgiving Holidays. |

A *communicative act* is an intentional unit (such as a request for information) which can be attached to one or more clauses within a dialogue turn. For example, the above dialogue turn may

be annotated as a sequence of three communicative acts: *acknowledge, inform, explain.* Most of the time, a communicative act is attached to a complete dialogue turn. *Rhetorical relations* link pairs of clauses. For example, a *support* relation exists between the *inform* core statement and the *explain* satellite contribution. The clauses linked by rhetorical relations may be within the same dialogue turn or may be across two different dialogue turns. *Discourse segments* are paragraph-like units that cluster together one or more clauses, typically spanning several successive dialogue turns. For example, the dialogue exchange listed at the beginning of the chapter may be labeled with a segment purpose entitled: *Find theaters playing the movie The Lion King.*

### 2.1.1 Discourse Segment Structure

The theories of intentional structure of discourse and *shared plans* [38, 39], dialogue games and transactions [17, 16] and contributions to discourse [21, 22] assume that task-oriented dialogue is a joint activity in which participants always select a move that is in the direction of accomplishing some mutually agreed upon purpose. From a computational perspective, speakers seem to behave as if trying to maximize a goal-oriented utility criterion. In the case of a cooperative information retrieval dialogue the utility criterion for the agent is determined by at least three successive steps:

1. Understanding what the customer says.
2. Recognizing her intention.
3. Either attempting to accomplish the intention or providing alternatives or explanations if the intention cannot be accomplished successfully.

The successful completion of each one of the steps require in turn the successful completion of the preceding ones (e.g., the agent needs to understand the customer in order to interpret her intention). For example, according to the theory of shared plans, sequences of one or more dialogue turns that share a common purpose are grouped together into what is called a *discourse segment*, in which ([39], page 442):

> *the initial utterances put on the table a proposal that there be a shared plan developed and carried out to satisfy the initiating conversational participant's desire; the subsequent utterance must somehow address this proposal, either accepting or denying it; assuming the proposal is accepted, subsequent utterances can provide information about any of the beliefs or intentions embedded in the definition of a shared plan.*

Figure 2-1 displays an example dialogue annotated with discourse segments. One feature of intention-based segmentation is that purposes are defined as goals with preconditions, acts, and

**Segment 1: List theater playing movie**
1 C: I'm trying to find out where the Lion King is located.

**SubSegment 2: Agree on movie name**
2 A: The Disney movie?
3 C: Yes.
4 A: Ok.

5 A: I don't think it's playing in the theaters anymore.

**SubSegment 3: Provide explanation**
6 A: They're supposed to re-release it [um] around the Thanksgiving Holidays.
...

Figure 2-1: Example of intention-based segmentation of a dialogue. Segments are labeled with their purposes.

post-conditions. Preconditions determine a partial order of related purposes. For example, the purpose **List theater playing movie** can only be accomplished after **Agree on movie name** has been successful, and the purpose **List show times** can be accomplished only if both **List theater playing movie** and **Agree on movie name** have been accomplished first. The partial ordering of purposes is reflected by the embedding relationship between discourse segments. In the example dialogue, Subsegment 2 is embedded into the top level segment 1 because its purpose is a necessary precondition of the first segment purpose. Some of the relationships between segments in dialogue include dominance, support and diversion. A segment dominates a subsegment if the subsegment purpose satisfies a precondition of the segment purpose. A subsegment supports a segment if the subsegment purpose is to provide additional related information, such as an explanation or further details. A subsegment is a temporary diversion if the purpose is unrelated to the top-level segment purpose. Either speaker may initiate clarifications and repair subsegments if any one of her goal-oriented criteria mentioned above fail (i.e., failing to understand what the other speaker says, failing to interpret the other speaker's intention, or discovering obstacles to the successful completion of her intentions). Later in this chapter, we describe another type of relation which may exist between segments: *rhetorical* relations.

Intention-based segmentation recognizes three related analysis structures [38]. The first one is the *linguistic* structure which corresponds to acoustic and lexical features of individual sentences and phrases. These features include intonation contours and discourse cue words which are correlated with segment transitions [82, 46, 45, 100]. The second one is the *attentional* structure, or focus of attention, which is a data structure that records the salient semantic entities that the dialogue participants can refer to in the dialogue (e.g., movie titles, theater locations, show times). The third one is the *intentional* structure, which lists the relations among the purposes being accomplished by the sequences of dialogue turns called discourse segments. This three-layered model has provided the seed for the model of dialogue as shared planning activity [39, 37, 57, 58].

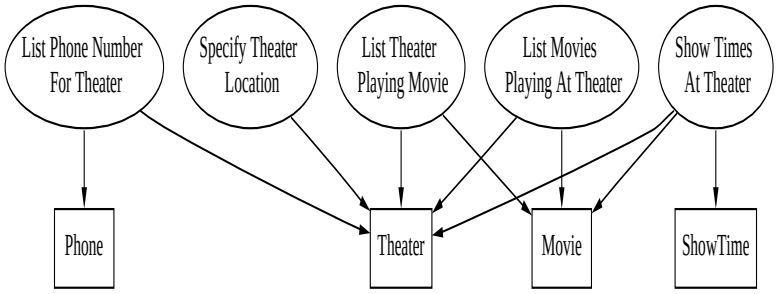



Figure 2-2: Task model for the movie schedule domain. Circles are segment purposes. Boxes are semantic entities that need to be communicated between the conversation participants. Edges are directed from purposes to dependent entities.

For simple information retrieval domains it is possible to quickly develop a list of specific purposes, at least from the agent's perspective. Figure 2-2 is an example task model for the movie schedule domain. Once the list of purposes is drafted, there are two problems in annotating them in text transcriptions. The first one is how to detect segment boundaries in the transcription (i.e., which sentences initiate or complete discourse segments). The second one is to determine exactly the embedding relations between segments, which should mirror, at least in principle, the hierarchical relations between their purposes.

In the following sections we present four other proposed units of analysis - contributions, communicative acts, rhetorical relations, and phrase co-reference - and we discuss their relationship with discourse segment structure.

### 2.1.2 Contributions to Discourse

The psycho-linguistic theory of discourse contributions focuses on explaining how conversation participants manage to *understand* each other's words and intentions, and view this process as a set of joint actions rather than actions accomplished unilaterally by one individual speaker. According to this theory ([22], page 259):

> *Conversations are highly coordinated activities in which the current speaker tries to make sure he or she is being attended to, heard, and understood by the other participants, and they in turn try to let the speaker know when he or she has succeeded.*

Figure 2-3 is an example of discourse contribution analysis drawn from our corpus. Typically, a contribution contains two phases that involve both participants in the conversation: a *presentation* phase, e.g., a statement or a request by one speaker, and an *acceptance* phase, e.g., an acknowledgment or some other appropriate response which demonstrates that the words and the speaker's

| | |
|---|---|
| 6 C: Okay, in Snellville, Septum Movies | Present. |
| 7 A: Sure, just one moment please | Accept. |
| 8 A: And that was the Septum Theater? | |
| 9 C: [Yeah] | |
| 10 A: Okay | |
| 11 A: Okay, I have a Cineplex Odion in Snellville | |
| 12 A: Do you know that there is one called the Septum? | |
| 13 C: [Yeah], it's called Septum, but Cineplex Odion is probably the theater who owns it, but it's called the Septum Theater, but [yuh], [yeah] that's it | |
| 14 A: Okay... *(lists movies playing at the Septum)* | |

Figure 2-3: Example of a presentation-acceptance discourse contribution extracted from our corpus.

intentions have been understood. The acceptance phase is necessary for accomplishing mutual understanding (i.e., *grounding* [11]) between the speakers, and may involve nested discourse contributions. Grounding is a crucial issue for spoken dialogue because speech is a transient medium which is also hidden. Listeners can only recall the last few words or phrases that are spoken, and they need to interpret the intentions of the speaker from sentences which may be elliptical, contain indirect expressions, and may be syntactically or semantically ambiguous. The role of the acceptance phase of a discourse contribution is precisely to manage the inevitable problems that occur in interpreting the speaker's words and intentions.

As in intentional discourse structure theory, the dialogue is viewed as a collective or joint activity driven by the speakers' agreed upon purposes. While intentional discourse theory focuses on explaining how conversation participants proceed in accomplishing a task, the theory of discourse contributions focuses mainly on describing the details of how the speakers successfully solve understanding problems when conveying information from one to the other. A discourse contribution can be viewed as the smallest possible discourse segment unit, whose purpose is to set a common ground of mutual understanding between the dialogue participants, and to prepare for the successful accomplishment of a top-level segment purpose. Larger segment units driven by task-related purposes, such as **List Movies Playing At Theater**, typically contain one or more nested discourse contributions.

### 2.1.3 Communicative Acts

The third type of discourse unit is called a *speech act*, *communicative act*, or *act* for short [92, 93, 1]. An act is an abstract label that is attached to one or more clauses in a dialogue turn. It attempts to summarize the intention that the speaker wants to communicate to the listener. A sequence of acts provides an account of the various steps that accomplish (or divert from) the purpose of the dialogue.

Figure 2-4 list one sequence of communicative acts for the example dialogue. Labels such as

| | |
|---|---|
| 1 | **Request for Information** |
| | C: I'm trying to find out where the Lion King is located. |
| 2 | **Request for Clarification** |
| | A: The Disney movie? |
| 3 | **Clarification Answer** |
| | C: Yes. |
| 4 | **Acknowledgment** |
| | A: OK, |
| 5 | **Inform Statement** |
| | A: I don't think it's playing in the theaters anymore. |
| 6 | **Support Statement** |
| | A: They're supposed to re-release it [um] around the Thanksgiving Holidays. |

Figure 2-4: Example sequence of communicative acts for the example dialogue.

*Request, Inform, Acknowledge* and *Support* are frequently used to classify communicative acts. Discourse segments are a complementary unit of analysis with respect to communicative acts. While the sequence of communicative acts provides a flat annotation structure, the point of view taken in discourse segmentation is top down, with discourse segments containing sequences of one or more related communicative acts. Some discourse segmentation theories correlate specific communicative act labels with discourse segment initiatives [17, 16]. For example, a discourse segment or subsegment may start with specific dialogue acts such as requests for information, requests for clarification and other direct questions.

### 2.1.4 Rhetorical Relations

The fourth type of discourse unit is *rhetorical* relations that exist between pairs of clauses, or pairs of speech acts [60, 49, 62, 61, 69]. Rhetorical relationship theories are motivated by the observation that clauses in text and speech do not occur in isolation. Instead, there is a small set of relationships that can be established between pairs of units (e.g., elaboration, background, motivation).

Figure 2-5 displays the rhetorical relations that can be established between the sentences in our example dialogue. While communicative act labels are attached to clauses without explicitly linking them to their context, rhetorical relations attach labels only to clause pairs. For example the relation **Responds(1,5)** simply states that the Inform statement in line 5 responds to the request for information in line 1, and the relation **Supports(5,6)** states that the statement in line 6 provides support for, or explains, the Inform statement in line 5. Rhetorical relations can be established bottom-up at different levels of detail. They provide a *rhetorical parse tree* of the organization of the text. Interestingly, the clause pairs that propagate from the bottom level to the top level may constitute a summary of the dialogue exchange, by providing an implicit hierarchy of the level of prominence of different sentences. Typically, core clauses that propagate to the top level constitute the presentation phase of discourse contributions, while the ones that are left behind

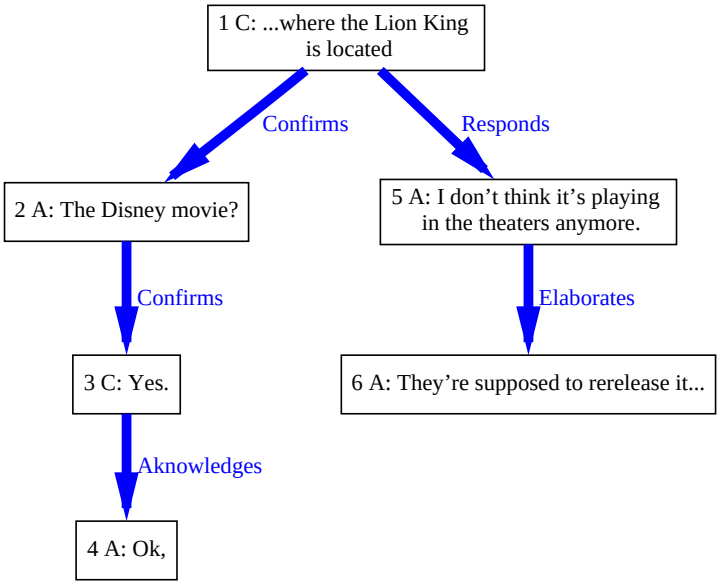



Figure 2-5: Example of rhetorical tree for the same dialogue exchange of Figure 2-1. Edges connect pairs of sentences linked by rhetorical relations.

are the acceptance phase. Rhetorical parsing of text has been used to provide summaries of texts with very encouraging results [63, 62, 61], or to automatically generate coherent explanatory texts [49]. One important contribution of [63] is the development and empirical evaluation of a full-blown discourse-level parser based on the principles of rhetorical relations.

Discourse segments are complementary to rhetorical relations [69]. While rhetorical relations build discourse structures from the bottom up from pairs of clauses, the point of view taken in discourse segmentation is to detect segment boundaries top-down. In particular, discourse segment structure algorithms focus on detecting segment initiatives related to switches in purposes. Typically, a discourse segment should contain certain pairs of clauses that are related by specific rhetorical relations such as: **Responds**, **Clarifies** or **Elaborates** relations. In general, a discourse segment should consist of a rhetorical parse tree, and a subsegment should consist of a rhetorical subtree. In particular, the discourse coding scheme proposed by Moser, Moore, and Glendening [68] is an attempt to unify discourse segment structure with rhetorical relations. In their scheme, a segment must be labeled with a purpose and contain a *nucleus*, or core contribution, and a series of *satellite* contributions related to the focus by rhetorical relations. In the example displayed in Figure 2-5, nodes 1,2, and 5 are nuclei and 3,4, and 6 are satellites. Each segment in Figure 2-1 contains one nucleus and one or more satellites (e.g., the nucleus for segment 1-6 is sentence 1, and the nucleus for segment 2-4 is sentence 2).

Another example of a linguistic theory which combines discourse segmentation with rhetorical relations is the one proposed by Polanyi [83]. In that model, a text is first segmented into elementary

communicative act units called *discourse coherent units* (DCUs). Each DCU has syntactic and semantic features attached to it. A right-branching parse tree may be built incrementally bottom-up, linking DCUs based on rhetorical and logical relations such as **Coordination, Elaboration** and **Interruption**. However, unlike the rhetorical models mentioned above, there are no specific nuclei and satellites. The text is processed sequentially by the parser. When a new DCU needs to be processed, the parser searches along the right edge of the tree for a suitable attachment point. When a suitable node is located, attachment is made. If no suitable node is available, a new node is created and inserted in the tree at an appropriate point.

### 2.1.5 Co-reference

The fifth type of discourse unit that can be annotated is noun phrase and pronoun co-reference. In the following example, four different phrases and pronouns refer to the same movie: *The Lion King*.

| | |
|---|---|
| 1 | C: I'm trying to find out where **the Lion King** is located. |
| 2 | A: **The Disney movie**? |
| 3 | C: Yes. |
| 4 | A: OK, I don't think **it**'s playing in the theaters anymore. |
| 5 | A: They're supposed to re-release **it** [um] around the Thanksgiving Holidays. |

Corpus-based analyses of co-reference try to establish the semantic relations that exist between definite noun phrases and pronouns [19, 44]. Resolving co-reference is an important function of the understanding component of a spoken dialogue system. This function is non trivial. For example, a resolution algorithm should be able to detect that in line 5, the pronoun *they* (in *they're supposed to...*) does not have an antecedent. Co-reference resolution is instrumental in determining *what* a dialogue or text is about. It provides an index of specific entities such as locations, events, people and objects that are mentioned in the course of the dialogue, and the relations that exist between them. In theory, these relations should be a small set of mutually exclusive classes (e.g., *is same as, is element of, contains...*).

The list of active entities that can be mentioned at any point in the dialogue constitutes the *focus of attention*, or the *attentional state* of the dialogue participants [35]. To conduct a cooperative dialogue, it is necessary for the participants to share a common attentional state. The purpose of the clarification segment *The Disney movie?-Yes* is precisely to ensure that both participants are talking about the same movie. When a clarification subsegment is absent, we assume that the participants are talking about the same entities, until a misunderstanding occurs. If a misunderstanding is detected by a participant, he or she may initiate a clarification subsegment, or discourse contribution, to repair the misunderstanding (e.g., *Do you mean X or Y?*). Discourse segment structure can be used in the search for co-references. It is assumed that each discourse segment has an associated

attentional state. When searching for a co-reference, a reasonable heuristic is to start looking in the attentional state of the embedded subsegment, and if a co-referent cannot be found, to look in the attentional state of the embedding top-level segment.

Segments can be defined as sections of the dialogue *about* the same topic or subtopic, independently of the speaker's intentions [13, 67, 41]. This alternative definition of segment is useful for information retrieval and indexing of text and speech, where it is desirable to extract from long documents paragraph-like sections that are about a particular topic, and it is not necessary to infer the speaker's intentions. Algorithms for subtopic segmentation rely on co-occurrence of lexical items that are related to each other by co-reference relations [67], or more simply by co-occurrence relations [41]. These algorithms assume that within a segment there are many lexical terms (e.g., nouns and proper nouns) that are related to each other with co-reference relations. The density of these related terms should be minimal across segment boundaries. Typically, subtopic segmentation produces either a linear segment structure or a graph of segments that may contain several discourse segment purposes.

## 2.2 Current Research Issues

Whether the unit of analysis is discourse segments, communicative acts or rhetorical relations, one important issue is matching the theory against observed data and comparing one theory against the other. The role of empirical studies is precisely to provide quantitative evidence for each discourse analysis unit. Empirical studies try to answer questions such as: How reliably can the units be annotated? What is the frequency of occurrence of each unit? Is it possible to compare theories when they are applied to the same corpus of data?

### 2.2.1 Different Corpora and Genres

Discourse phenomena that occur often in one corpus might be very infrequent or irrelevant when analyzing a corpus in another domain. As a consequence, it is hard to develop and test empirically a comprehensive theory of dialogue. One important issue in empirical studies is to determine which phenomena appear in all corpora, and which phenomena are dependent on the corpus being analyzed. To date, at least four task-oriented human-to-human dialogue corpora and one large corpus of casual conversations have been used for empirical studies in British and American English. In addition, at least one corpus of spoken monologues has been used to conduct discourse segmentation studies (i.e., the Boston Direction Corpus [45]).

The London-Lund corpus has been a pioneering effort in collecting and transcribing spontaneous speech, including some task-oriented telephone conversations [99]. The corpus contains 500,000 words of spoken British English recorded from 1953 to 1987. It consists of 100 text samples. The

text genres include transcriptions of directory assistance conversations between telephone operators and customers as well as monologues, commentaries and public speeches. The corpus used for this thesis is similar in nature to the directory assistance portion of the London-Lund corpus.

Figure 2-6 displays representative sections from four other corpora, and illustrates the differences in domain and conversational style. The Trains corpus was collected at the University of Rochester [2]. It is a collection of 98 dialogues recorded in a laboratory. In each one of the dialogues, one speaker plays the role of user and another one plays the role of the assistant. The assistant helps the user in accomplishing a task involving the manufacturing and shipment of goods in a hypothetical railroad freight system. The Map Task Corpus collected at the University of Edinburgh, England consists of 128 spontaneous face-to-face dialogues also collected in a laboratory setting [3, 16]. In each dialogue, each of the two speakers has a map, one with a route marked and one without. The goal is for the route follower to draw the route from the instructions of the route giver. Verbmobil is a bi-lingual corpus, with some portions in American-English and some in German. The American-English only portion of the Verbmobil corpus was collected at Carnegie Mellon University [50]. The data include 313 dialogues in an appointment scheduling task. Finally, the Switchboard corpus is a collection of 2400 casual telephone conversations between speakers from all areas of the United States [51]. At the start of each conversation, speakers were prompted to talk casually about one of 70 different everyday topics (e.g., how to buy a car).

The dialogue samples in the figure differ in at least two dimensions: one is the participants' roles in the conversations, and the other is the complexity of the task. In Map Task and Trains, the role of each speaker is different: user vs. assistant and route giver vs. route follower. In Verbmobil and Switchboard, the relationship between speakers is peer-to-peer. Some other relationships that have been analyzed in other corpora are: expert vs. apprentice [105] and customer vs. agent (this thesis). In a user vs. assistant setting, for example, the types and size of contributions spoken by the assistant are much more limited than in a customer vs. agent setting or in a peer-to-peer setting. Another dimension which differentiates between corpora is the complexity of the task. Of all the domains, the Trains corpus is the most complex, requiring planning a shipment route under a large number of constraints about quantity and location of goods. Under these conditions, it is not straightforward for the speakers to determine which is the best route. In contrast, in Map Task, a multi-step route is assigned a priori on a map, and the problem is for the route giver to communicate the information (the route) successfully to the follower. In Verbmobil, two peers have to agree on a specific date and a specific time for scheduling a meeting, and the dialogue proceeds by successively proposing and evaluating alternative dates and times. Finally, in Switchboard, only a rather general topic of conversation is given a priori, and the purposes that are accomplished are not immediately apparent, which result in loosely structured dialogues.

The corpus used for this thesis is a collection of recorded telephone conversations between cus-

| | |
|---|---|
| ***Trains***<br>**User:** okay [breathing] so we have the three boxcars at [pause] Dansville so how far is it from Avon to Danville<br>**Assistant:** three hours<br>**U:** three hours then from Dansville to Corning<br>**A:** one hour<br>**U:** okay so we we can actually use those three boxcars right<br>**A:** mm-hm<br>**U:** okay so that's three boxcars okay so that's engine E one from Avon going to Dansville... Pick up the three boxcars go to Corning load them up and then take it to Bath okay so that's... good<br>.... | ***Map Task***<br>**Giver:** Ehm, do you have the start, yeah?<br>**Follower:** uh-huh, right in the –<br>**G:** And the diamond mine?<br>**F:** Up at the ... uh-huh up at the top to the left of the diamond mind?<br>**G:** Yeah<br>**F:** Mine, right.<br>**G:** Right. If you come down to the just below the in the diamond<br>**F:** Okay, straight down?<br>**G:** Yeah<br>.... |
| ***Verbmobil***<br>**JB:** Maybe we should get together to talk further about this how 'bout some time in the next couple of weeks ?<br>**SR:** okay well I will be on vacation for the next two weeks how about Friday the twenty first<br>**JB:** Friday the twenty first is scheduled from early morning to late afternoon could you perhaps choose another day, a morning on a Wednesday or an early afternoon on a Tuesday ?<br>**SR:** mornings on Wednesdays seem bad how 'bout the twenty seventh, twenty eighth or thirty first<br>.... | ***Switchboard***<br>**A:** What kind do you have?<br>**B:** Uh, we have a, a Mazda nine twenty nine and a Ford Crown Victoria and a little two seater CRX.<br>**A:** Oh, okay.<br>**B:** Uh, it's rather difficult to, to project what kind of, uh-<br>**A:** We'd look, always look into, uh, Consumer Reports to see what kind of, uh, report, or, uh, repair records that the various cars have-<br>**B:** So, uh-<br>**A:** And did you find that you like foreign cars better than the domestic?<br>**B:** Uh, yeah. We've been extremely pleased with our Mazdas.<br>.... |

Figure 2-6: Sample sections extracted from four different dialogue corpora, illustrating differences in domain and conversational style.

tomers and telephone operators from BellSouth and American Airlines. This corpus, similar to sections of the London-Lund corpus, differs in several respects from the other corpora. Unlike Switchboard, the information retrieval conversations are focused and goal oriented. In particular, the telephone operators are trained to keep the conversation focused on the task. Unlike Map Task, Trains and Verbmobil, the data were collected on the field for quality of service analysis and not for the purpose of this research. The nature of the information retrieval tasks does not require extensive problem solving knowledge. The complexity of at least three of the information retrieval domains in this corpus (i.e., movie schedules, automobile classifieds and restaurant guide) and of Verbmobil is within reach of current state-of-the-art spoken language systems, while the complexity of Map Task and Trains is beyond the scope of current spoken language systems. The difference between the Verbmobil corpus and the one presented here is that the appointment scheduling task requires *negotiating* a date between two peers, each one with her or his own constraints. In our corpus, only one task, flight booking, requires negotiating dates and fare type based on many constraints. The other tasks are typically information-seeking dialogues in which the roles of the speakers are clearly differentiated between the information seeker (the customer) and the information giver (the agent). Differences between negotiating and informing produce differences in the structure of the conversations. For example, in the information-seeking segments it is possible to differentiate between a *request* contribution, in which the request for information is specified, and the *response* contribution, in which the information is reported. This type of segmentation is not appropriate for the other tasks, such as appointment scheduling.

### 2.2.2 How Many Units, and Which Ones?

To allow researchers to share data and knowledge about discourse units, there is a need for unifying naming conventions for discourse and dialogue level tags across different systems, different linguistic theories, and different languages. For example, it would be very interesting to list which communicative acts open or close discourse segments, which rhetorical relations must be contained in a discourse segment, and which pairs of communicative acts can be linked by rhetorical relations. A related open problem is to determine whether or not communicative acts, rhetorical relations, subtopics and intentions would produce consistent analyses of the same text, i.e., whether or not the implicit or explicit segmentation produced by a theory can be contained in the segmentation produced by another theory without crossing segment boundaries [69].

Most linguistic theories postulate the existence of a set of exhaustive categories, whether they are speech acts, rhetorical relations, or another type of unit. Discourse segment purposes tend to be domain dependent and task specific, such as **List movies playing at theater**. While larger set of units can potentially describe a dialogue with a greater level of detail, a smaller set of units may be desirable to decrease the learning curve for the annotators and increase the consistency and

reliability of the annotated data. Task specific units are more easily learned by annotators who are not experts in linguistics, while abstract linguistic tags might be applicable to many different domains, but might be harder to learn.

The Map Task group at the University of Edinburgh has developed annotation instructions for a small set of communicative act tags and segment tags called transactions, games, and moves [16]. In the Map Task coding scheme, transactions and games correspond to discourse segments and subsegments. The communicative act tags are organized into segment initiatives (commands, questions) and responses (replies and acknowledgments).

In 1996 and 1997, two workshops were organized by the Discourse Resource Initiative [59, 15]. The major outcome of the workshops was an instruction manual for communicative acts called dialogue acts in multiple layers (DAMSL) [28]. The manual specifies independent dimensions for tagging forward looking acts and backward looking acts. Forward looking acts correspond to intentional speech acts, such as *Statement* and *Request*. Backward looking acts indicate the relationship between the current act and the dialogue history, such as *Respond* and *Agreement*. A dialogue turn can be tagged with multiple labels, one per independent dimension. Corpus specific tags can be specified as subclasses of the abstract tags. The DAMSL tag set has been adapted to tag the Switchboard corpus using a set of 42 mutually exclusive speech act tags [51].

The papers by Mann and Thompson [60], by Moser, Moore and Glendening [69] and Marcu's doctoral thesis [62, 61] list a set of rhetorical relations that can be used for rhetorical parsing of text.

In the area of discourse segmentation, the group led by Grosz at Harvard has published a manual for annotating direction giving monologues according to discourse segment purposes [71]. In their approach, annotators are free to choose label names, and the evaluation is conducted by measuring agreement in placing segment boundaries, ignoring domain-specific and theory-specific segment label names. This is a good approach for dealing with the problem of comparing and unifying discourse analysis units, because it focuses on assessing whether one theory is consistent or compatible with another one, without enforcing a particular naming convention for each unit.

### 2.2.3 Ambiguous Data and Subjective Tasks

One problem encountered in tagging discourse units is that the same word string can be mapped into more than one speech act, rhetorical relation, topic or segment purpose. For example, an inform statement might serve as response, implicit acknowledgment or agreement, and at the same time initiate a new segment while a question might initiate two or more distinct purposes. Often, a text may be interpreted in more than one way, and analyzed from different points of views. As a consequence, annotating the discourse structure of a text is a subjective task, unless the instructions state clearly the point of view to be taken. Also, discourse unit boundaries are often difficult to detect because they are not always indicated by unambiguous acoustic correlates or lexical cues.

One of the goals of this tesis is to understand under which conditions it is possible to annotate reliably discourse segment boundaries.

### 2.2.4 The Need for Annotation Tools

To produce a large amount of useful data for empirical research, we must make sure that the annotation of discourse units can be done *easily* and *consistently* by annotators with minimal training in linguistic and pragmatics. The annotation task can be simplified if the annotators are provided with efficient editing tools.

The annotation tool described in this thesis has been a pioneering effort in empirical studies of discourse. However, presently, *Nb* is not the only discourse annotation tool available. Because it is difficult to design an annotation tool that is appropriate to handle many different coding schemes, all tools with one exception have been developed to comply with one specific coding scheme. The series of MUC conferences [19] used the co-reference annotation tool called DDTool developed at SRA labs [5]. The DDTool allows users to display chains of co-referent nouns and noun phrases by linking them with colored straight lines. In addition to the DDTool, Melamed developed at the University of Pennsylvania another word annotation tool called Blinker, with the purpose of annotating corresponding words between English and French translations of the Bible [66]. After the release of *Nb* , the Discourse Resource Initiative project developed a tool for tagging multiple layers of communicative acts called DAT. The tool has been designed to work with one particular coding manual, the DAMSL coding manual [15, 28]. A team of five natural language researchers at MITRE is developing a generic annotation tool called the Alembic Workbench [44]. The goal of the Alembic Workbench is to provide graphical authoring tools to annotate textual data with fully customizable tag sets, machine learning tools to bootstrap the annotation process, and evaluation tools to analyze annotated data using measures such as precision and recall. The Alembic Workbench is currently being developed to annotate nouns, noun phrases and prepositional phrases in multilingual text corpora. The data annotated with the Alembic Workbench is used to foster research and development in text understanding, summarization, and information extraction. Finally, the Human Language Technology group at the University of Edinburgh is developing a generic software toolkit for generating and parsing annotated text using SGML (standard generalized mark-up language) and XML (extensible mark-up language) [101].

## 2.3 Evaluating the Reliability of Annotations

Various metrics have been proposed to measure the agreement among coders for different linguistic annotations. For example, phonetic transcriptions have been compared in terms of a pairwise inter-coder agreement, taking into account insertion/deletion as well as substitution errors. Inter-coder

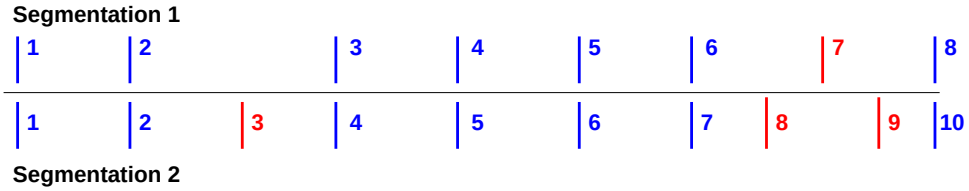



Figure 2-7: Comparing linear segmentations. Boundary 7 in segmentation 1 and boundaries 3, 8, and 9 in segmentation 2 do not agree. The overall agreement depends on the actual length of the text.

agreement has been found to be 80-85% on phonetic transcription tasks [25]. In this section, we review how to compute four metrics for measuring inter-coder agreement: precision, recall, percent agreement and the kappa coefficient. A discussion of evaluation metrics can also be found in [53, 6, 104, 14].

### 2.3.1 Precision and Recall

When comparing two different annotations of the same text, we may select one as the reference and the other one as the test. When the segmentations are linear, the annotation task is reduced to a two-way classification of each clause (*Segment Boundary* if the clause opens a new segment, else *Non Boundary*). The different measures are computed from the confusion table of the different categories:

| | Test | not Test |
|---|---|---|
| Ref | C | D |
| not Ref | I | N |

In the table, $C$ is the number of data samples (clauses) which were proposed to be segment boundaries by both annotators. $D$ counts the reference boundaries that were deleted by *Test*. $I$ counts the number of extra boundaries inserted by *Test*. Finally, $N$ counts the number of data samples that were classified as *non-boundary* by both coders. *Recall* measures the proportion of references boundaries that were correctly identified by *Test*, and *precision* measures the proportion of false alarms. They are derived from the confusion table as follows:

$$Recall = \frac{C}{C+D} \tag{2.1}$$

$$Precision = \frac{C}{C+I} \tag{2.2}$$

Consider the example displayed in Figure 2-7 that shows two possible segmentations for a hypothetical text running from left to right. The two segmentations agree over seven boundary locations. If we take segmentation 1 to be the reference segmentation, then segmentation 2 deletes 1 boundary and inserts 3 boundaries. Recall and precision values are:

$$Recall = \frac{7}{8} = 87.5\% \tag{2.3}$$

$$Precision = \frac{7}{10} = 70.0\% \tag{2.4}$$

If we take segmentation 2 to be the reference segmentation, precision and recall values are reversed (insertions become deletions). Sometimes it is convenient to report only one measure for precision and recall. This measure is the *F-value* and it is a weighted average of precision and recall. Usually, the cost of insertion errors is the same as the cost for deletions, and precision and recall are weighted equally. For this example the F-value is: 78.75%. Precision, recall and F-value are independent of the text length.

### 2.3.2 Percent Agreement

Precision and recall are useful metrics for considering one category at a time (i.e. *Segment Boundaries*). The percent, or observed, agreement $P_o$ measures coder agreement for all the annotated categories. In case of a two-way classification task, it is:

$$\begin{aligned} Text\ size\ T &= C + D + I + N \\ P_o &= \frac{C + N}{T} \end{aligned} \tag{2.5}$$

If the two categories are segment boundary and non-boundary, the observed agreement is:

$$P_o = \frac{boundaries\ correct + nonboundaries\ correct}{text\ size} \tag{2.6}$$

The agreement $P_o$ is proportional to the total number of clauses $T$, i.e., the text length. In the example displayed in Figure 2-7 if the text is short, (e.g., $T = 15$ clauses), the two coders must have agreed on 7 boundaries and 4 non-boundaries, and the agreement is $P_o = \frac{7+4}{15} = 73\%$. If the text is longer (e.g., $T = 40$ clauses), the coders must have agreed on 7 boundaries and 29 non-boundaries, and the agreement is: $P_o = \frac{7+29}{40} = 90\%$.

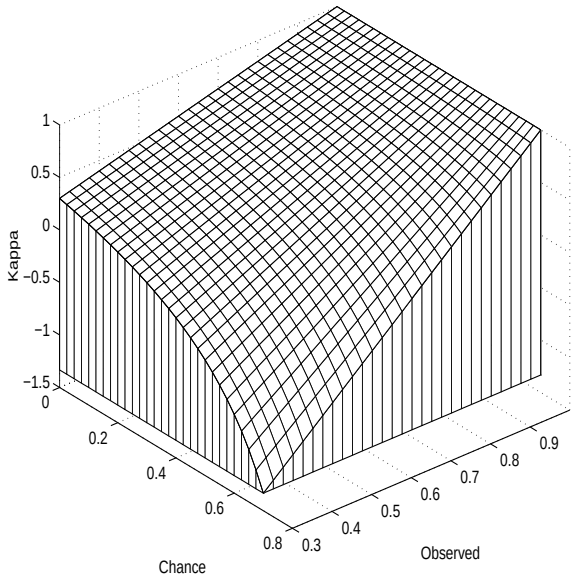



Figure 2-8: The kappa coefficient as a function of observed agreement and chance agreement.

### 2.3.3 The Kappa Coefficient

The problem with reporting only the percent agreement, or only precision and recall, is that they can overestimate the agreement among coders when the distribution of the coding categories is skewed. In particular, in a long text, most of the clauses will be classified as non boundary by both coders.

Recent empirical work on discourse coding has overcome the text size sensitivity problem by either reporting the agreement on the most critical categories (i.e., *Segment Boundary*) [46] or by reporting the kappa coefficient, $\kappa$, a measure of agreement that is used in experimental psychology [53, 6, 14]. This measure corrects the observed agreement by subtracting the estimated *chance agreement* $P_c$ one expects a priori from the marginal distributions of the coded categories. The coefficient is computed as follows:

$$\kappa = \frac{P_o - P_c}{1 - P_c} \tag{2.7}$$

For linear segmentations, the chance agreement $P_c$ is computed by summing the marginal distributions of the two categories (boundary and non-boundary):

$$P_c = \frac{(C + D)(C + I)}{T} + \frac{(N + D)(N + I)}{T} \tag{2.8}$$

Figure 2-8 shows how the kappa coefficient is directly proportional to the observed agreement and inversely proportional to the chance agreement. The coefficient is greater than 0.6 for values of the observed agreement greater than 0.8 and values of the chance agreement less than 0.5. The

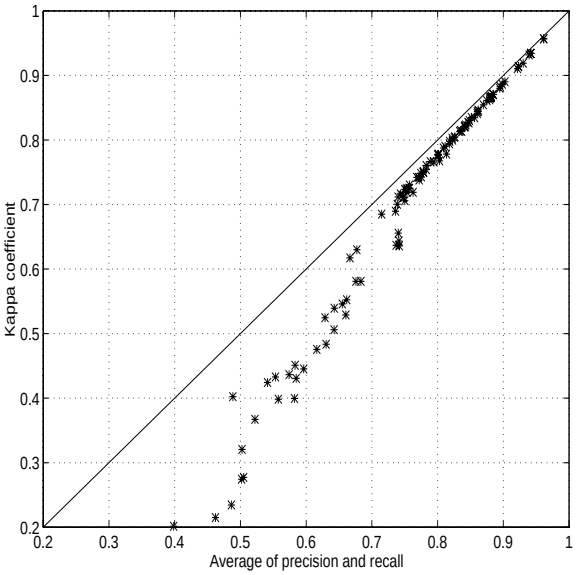



Figure 2-9: The kappa coefficient as a function of precision and recall.

coefficient drops dramatically when the chance agreement increases above 0.5 and the observed agreement decreases below 0.8.

In the example in Figure 2-7, in the short text a large fraction of the clauses is classified as boundaries by both coders, while for the longer text a large fraction of the clauses is classified as non-boundaries by both coders. In particular, the fraction of clauses coded as boundary by coder 1 is $\frac{8}{15} = 0.53$ for the shorter text, and $\frac{8}{40} = 0.2$ for the longer text. The respective fractions for coder 2 are: $\frac{10}{15} = 0.66$ for the shorter text, and $\frac{10}{40} = 0.25$ for the longer text.

For the shorter text, the two coders have an a-priori probability of $0.53 \times 0.66 = 0.35$ of agreeing over boundary locations, and a probability of $0.47 \times 0.33 = 0.15$ of agreement over non-boundary locations. Summing the chance probabilities for the two categories (i.e., 0.35 for boundaries and 0.15 for non-boundaries) gives an overall a priori chance of agreement $P_c = 0.5$.

In the example, the chance agreement for the shorter text is large because a large fraction of the clauses are coded as boundaries by both coders. This results in a low value of the $\kappa$ coefficient: $\kappa = \frac{0.73-0.5}{1-0.5} = 0.46$. In contrast, if we carry out the same computations for the longer text, we obtain $P_c = 0.65$, and the agreement coefficient is significantly higher: $\kappa = \frac{0.9-0.65}{1-0.65} = 0.71$.

The experimental psychology literature reports that a value of $\kappa$ greater than 0.6 indicates statistical correlation among coders and $\kappa$ greater than 0.7 can be interpreted as an indication of replicable agreement among coders [6]. In our example, the values of the $\kappa$ coefficients confirm that the agreement among the two coders is significantly more reliable for the longer text. Figure 2-9 displays the kappa coefficient as a function of the F-value (the average of precision and recall) and

provides a visual explanation for choosing a reliability threshold at 0.7. The plot has been derived from the four annotation experiments described in Chapter 4, by collecting 125 different pairwise agreements among coders in placing discourse segment boundaries. In general, $\kappa$ is found to be always less than the F-value. The plot indicates that $\kappa$ is almost identical to (and less than) the F-value above 0.7, with a very small variance. In contrast, it drops significantly for values below 0.7, with a larger variance.

The agreement measures can be adapted to comparing multi-level segmentations of the same text. Precision and recall can be computed by counting the number of segments or subsegments that appear in both segmentations, and the number of segments inserted or deleted in one of the segmentations.

For evaluation purposes, the segmentation task can be linearized if segment and subsegment initiatives are grouped into one class, and all other sentences are grouped into another class. The two-way classification task can be evaluated using $\kappa$. Alternatively, the segmentation task can be mapped into a three-way classification task: *Segment Initiative, SubSegment Initiative* and *Other* and $\kappa$ can again be computed. Some researchers suggest first to evaluate where coders agree in placing segment initiatives, and then to evaluate if they also agree on segment closings for the segments in which they agree on the initiatives [45, 80].

One important feature of the kappa coefficient is that it can be extended to experimental conditions with multiple coders, more than two annotated categories and missing annotated data. In such cases it becomes the group-wise kappa coefficient. Because it is not easy to find in the literature the exact definition of the chance probability for this extended case, we report in Appendix A the exact formulae that have been used to compute the group-wise kappa coefficient in Chapter 5.

### 2.3.4 State of the Art in Evaluating Inter-Coder Agreement

Satisfactory levels of agreement have been reached for tagging linguistic phenomena in large written and spoken corpora, provided that subjects are trained appropriately (sometimes extensively), using baseline coding schemes that are agreed upon by many researchers and are common to different linguistic theories. Agreed upon units include phonemes and phonetic variants [25], intonation labels [96], and syntactic parse trees [64]. In contrast, reliability studies in discourse analysis have explored more specific contexts [27, 72, 107, 59, 15].

Some of the topics studied by empirical work in discourse analysis include defining and evaluating the consistency of coding schemes for speech act tags (e.g., [14, 27]), frequency analysis and models for speech repairs, grounding contributions and other spontaneous speech phenomena such as repairs (e.g., [103, 42]), correlation analysis between prosodic cues and discourse segment boundaries (e.g., [46, 36, 45]), evaluating the agreement among subjects in placing discourse boundaries in transcriptions of monologues [72], and evaluating human and algorithmic performance in annotating

| *Annotation Task* | *Citation* | *% Agree* | *Recall* | *Precision* | *Kappa* |
|---|---|---|---|---|---|
| *Discourse Segments* | | | | | |
| Subtopic Segmentation of written science articles | Hearst 97 [41] | | 81.6 | 71.4 | 0.67 |
| Discourse Segmentation of spoken directions from text | Hirschberg and Nakatani 96 [45] | | | | 0.63 |
| Discourse Segmentation of directions from text and speech | Hirschberg and Nakatani 96 | | | | 0.80 |
| Discourse Segmentation of spoken stories | Passonneau and Litman 97 [80] | | 63.3 | 70.6 | |
| Map Task Move Boundaries | Carletta et al. 97 [16] | 89.0 | | | 0.92 |
| Map Task Transaction Segments | Carletta at al. 97 | | | | 0.59 |
| *Communicative Acts* | | | | | |
| DAMSL Forward looking Speech Act labels | Core and Allen 97 [28] | 82 - 93 | | | 0.15 - 0.70 |
| DAMSL Backward looking Function labels | Core and Allen 97 | 78 - 95 | | | 0.57 - 0.77 |
| DAMSL Forward Acts adapted to Switchboard | Jurafsky at al. 97 [51] | 84 | | | 0.80 |
| Map Task Move Speech Acts | Carletta et al. 97 | | | | 0.83 |
| *Rhetorical Relations* | | | | | |
| Selection of prominent sentences in written science articles | Marcu 97 [63] | 71 | 55.5 | 66.6 | |
| *Co-referent Words and Phrases* | | | | | |
| Pronouns and noun phrases in newswire text | Hirschman et al 98 [44] | | 80 - 90 | 85 - 90 | |
| Aligning content words in English-to-French translations | Melamed [66] | 90 - 92 | | | |

Table 2.1: Some representative studies in inter-coder agreement in annotating units in text and speech.

co-referent noun phrases in newswire text [19, 44].

Table 2.1 illustrates the state of the art in evaluating text analysis units by listing some representative inter-coder agreement studies that have been published in the last few years. The list is not exhaustive. It is meant to provide the reader with an indication about the numeric range for the various evaluation measures when applied to different tasks and corpora. The values for precision, recall and kappa reported in the rest of this thesis can be evaluated using the values reported in this table as reference points. However, a direct comparison among different units is difficult because they differ along at least four dimensions (it is the famous apples and oranges dilemma). Different dimensions include the number and text size of the units, the number and prior knowledge of coders, the genre of the text to be annotated, and the intrinsic cognitive difficulty of the annotation task. For example, the highest value of precision and recall in annotating co-referent phrases has been obtained in a pilot experiment by two expert coders annotating three texts [44], and the highest agreement for segmental units has been obtained for segmenting individual dialogue turns into one or more communicative acts (i.e., dialogue moves, or conversational clauses). In general, the val-

ues reported in the table indicate that this is an emerging, rather than established, research field in computational linguistics (e.g., [107]). Finally, caution should be taken into selecting one unit of analysis based on comparing reliability, because some scientific issues can only be explored by studying where people disagree.

# Chapter 3

# Efficient Discourse Annotation with *Nb*

Annotated corpora can help researchers understand the regularity and variability of linguistic phenomena under investigation, propose computational models to mimic their behavior, estimate the parameters of the models, and evaluate the effectiveness of either the models or systems that embed these models [43]. When a corpus is annotated by more than one trained coder, assessing where coders disagree is crucial for understanding the difficulty or linguistic ambiguity of the task.

To develop phonetic recognition algorithms, for example, researchers in the US have relied on the TIMIT corpus [54] to understand the acoustic realizations of phonemes under varying phonetic environments and to develop phonetic models to capture such contextual variations [55]. Speech corpora annotated with the ToBI prosodic labels have been crucial in fostering progress in modeling suprasegmental acoustic features [96] and in the study of the correlation between prosody and discourse segment structure [46, 100]. In the area of text processing, text corpora annotated with co-referent nouns and noun phrases have been instrumental in monitoring the progress in automatic information extraction algorithms developed for the series of Message Understanding Conferences (MUC) [19, 44].

Once a corpus is made available for research, a good set of annotation tools can greatly facilitate the annotation process, both in throughput, accuracy, and consistency, thereby leading to useful data that can serve the needs of the research community.

The focus of this thesis is to understand the discourse structure of natural task-oriented dialogues. To achieve this goal, the types of units we wish to annotate are (possibly embedded) discourse segments and communicative acts. Discourse segments typically span several dialogue turns, while the size of communicative acts is the clause. When we started the research work for this thesis, there were no tools publicly available for efficiently annotating these two types of units. As a

consequence, we decided to develop a discourse segmentation tool called *Nb* that would make it possible to annotate discourse segments and communicative acts. We also decided to make it freely available on the Internet.

The following sections provide an overview of *Nb* . We describe how it has been used for developing the four discourse segmentation experiments reported in the next chapter, and how it has been tested with eight other coding schemes. We evaluate the success of the tool and we list its features and limitations.

## 3.1 Purpose of *Nb*

*Nb* has been developed to annotate embedded discourse segments, communicative acts, and phrases in text transcriptions. The tool allows researchers and trained coders to build discourse data structures by using embedded mark-up tags that wrap around the annotated text. For example, given an input text transcription with one clause per line (where A stands for Agent and C for Customer):

C: How about the Town Center Cinema in Lawrenceville?
A: Okay, sure
A: Alright, playing there, we have Love Affair
C: What time is that?
A: Love Affair is twelve, two thirty, five...

*Nb* allows users to mark the text with tags that wrap around phrases (e.g., semantic tags), single lines (communicative acts) or multiple lines (embedded discourse segments). The tags are indicated with simple mark-up conventions, where `<X>` indicates the begin point of a tag and `</X>` indicates the end point. Figure 3-1 is an example of annotated text (the syntax of the mark-up is simplified to improve readability).

*Nb* has been designed to allow the user to define the mark-up units, or coding categories (e.g., **Segment, NP, Act**), to enumerate the unit's possible values, and to indicate syntactic constraints for the units (e.g. whether they can span phrases, one line or multiple lines, whether they can be embedded or not). For example, a segment unit of type **Segment** can have as value a domain-specific purpose such as **List Movies** and **List Times**. Segment units span multiple lines and they can be embedded but they may not cross each other' boundaries. Communicative act units of type **Act** can have values such as **Request, Ack, Inform**. If the text is segmented a priori into one clause per line, then there should be one and only one act per line.

A mark-up language is very efficient for parsing, generating and sharing computer representations of discourse data structures. For example, if the same text has been annotated by multiple coders who produced multiple annotated files with the same syntax, generic programs can be used to evaluate inter-coder agreement with precision, recall and kappa. If a large corpus of hundreds of

```
<Segment 1 List Movies>
   <Act 1 Request> C: How about
      <NP 1 Location>the Town Center Cinema in Lawrenceville </NP 1>
   </Act 1>
   <Act 2 Ack>
      A: Okay, sure
   </Act 2>
   <Act 3 Inform>
      A: Alright, playing <NP 2 Location> there </NP 2>,
      we have <NP 3 Movie>Love Affair </NP 3>
   </Act 3>
   <Segment 2 List Times>
      <Act 4 Request>
         C: <NP 4 Time> What time </NP 4> is <NP 5 Movie>that </NP 5>?
      </Act 4>
      <Act 5 Inform>
         A: <NP 6 Movie>Love Affair</NP 6> is
         <NP 7 Time>twelve, two thirty, five </NP 7>
      </Act 5>
   </Segment 2>
</Segment 1>
```

Figure 3-1: Example of text transcription annotated with embedded mark-up tags which represent discourse segments, communicative acts, and semantic tags.

conversations is annotated, it is possible to automatically derive probabilistic models for segment transitions and communicative act transitions based on the observed frequency counts. Different programs can share a common generic library of functions for parsing and generating tagged text.

However, it is impractical and susceptible to error to have to type the mark-up tags using a text editor. *Nb* has been designed to provide an easy-to-use graphical user interface for producing annotated files efficiently without typing. The user does not need to enter the mark-up tag by typing it. Instead of typing in a mark-up tag, the user highlights the text spanned by a unit, and then chooses the unit type and value either by selecting it from a list or by entering a keyboard shortcut. The text is automatically color coded and indented by *Nb* according to the annotation. In addition, *Nb* allows users to easily delete or rename a tag, to undo editing actions, and to quickly browse the text and go to a specific tag. With *Nb* it is also possible to listen to the speech signal which correspond to annotated sections.

*Nb* has been developed with the goals of being ergonomic and portable. By ergonomic we mean that a tool should present the annotation instructions and the annotated text in a clear-cut way, so that subjects will be able to produce consistent annotations by focusing on the annotation task, rather than having to master the annotation tool. A good tool should also enforce some syntactic constraints compatible with the coding instructions, to ensure that the annotated data are consistent with the syntactic constraints and free from trivial errors. By portable we mean that the interface should be general enough to accommodate a large variety of annotating instructions and linguistic

theories, so that changes in the set of coding units can be incorporated easily, possibly without modifying the internal structure of the software. A portable tool should allow researchers to rapidly prototype and deploy a novel annotation experiment. In addition, a good tool paired with a good set of written instructions should allow coders to work on their own without interacting with an expert.

## 3.2 Iterative Design

*Nb* has been designed iteratively and over a period of three years involving three major releases. The first release, completed in 1995 and reported in [32], was an X/Motif graphical user interface written in C that allowed users to annotate discourse segments and dialogue acts line by line. At that time, *Nb* ran only under the Sun OS Unix operating system. The last two releases have been developed in Tcl/Tk and are available for all flavors of Unix, Windows and Macintosh operating systems. In the following sections, we briefly describe the first two releases of *Nb* , and then we explain in more detail how to annotate text and develop discourse coding schemes using the third release of *Nb* , which is the one that provides the best functionality.

### 3.2.1 Release 1

Figure 3-2 displays a screen shot of the first release of *Nb* . The top left panel indicates the editing action being performed and the segments that are open at the current line. The top right panel has buttons for reading and saving annotated files, editing the text and playing back the speech signal corresponding to the current line. The bottom right panel displays the annotated text centered at the line that is currently being annotated. The bottom left panel has buttons for browsing the text and opening and closing segments. This interface allowed users to annotate discourse segments and give them arbitrary titles corresponding to their purposes.

The main usability problem with the interface was that while segment units are multi-line structures, the editing was done one line at a time. The user would scroll the text to a specific line, and then select to either open a new segment or close any of the segments that were open at that line. Annotating the file line by line tended to slow down the annotation process. In addition, the user interface did not have a means of clearly highlighting segments. Another problem with this first release was that the tool was designed exclusively to annotate discourse segments with purpose names. New coding conventions had to be implemented at the source code level.

### 3.2.2 Release 2

The second and third releases, completed in 1996 and 1997, were a complete redesign of *Nb* using the Tcl/Tk programming language. In the latest two releases, the user interface becomes truly *what*

Figure 3-2: Screen shot of the first version of *Nb* . This release allowed users to annotate discourse segments line by line by typing their purposes.

*you see is what you get* (WYSIWYG).

Figure 3-3 is a screen shot of the graphical user interface for the latest two releases of *Nb* . The main window displays a document containing both the instructions and the annotated text, which is indented and colored automatically by *Nb* according to the annotation. In the figure, a section of dialogue is annotated with two segments and two sub-segments. Two pop-up menus at the bottom left corner of the screen provide choices for annotating either five different discourse segment purposes (top list) or five different communicative acts (bottom list). Annotation is performed by highlighting a section of text and selecting the corresponding unit from one of the lists. The input and output annotated data are formatted using a simple mark-up language which is compatible to standard languages for annotating text such as SGML (standard generalized mark-up language) and XML (extensible mark-up language). In the latest two releases, the input to *Nb* is a collection of files that include all the text to be annotated, and a configuration file that lists all of the coding conventions and syntactic constraints specific to the chosen coding scheme.

### 3.2.3 Release 3

The third release of *Nb* was completed in 1997. In this release, the graphical user interface has the same look and feel of the second release, with direct editing of discourse units and color highlighting

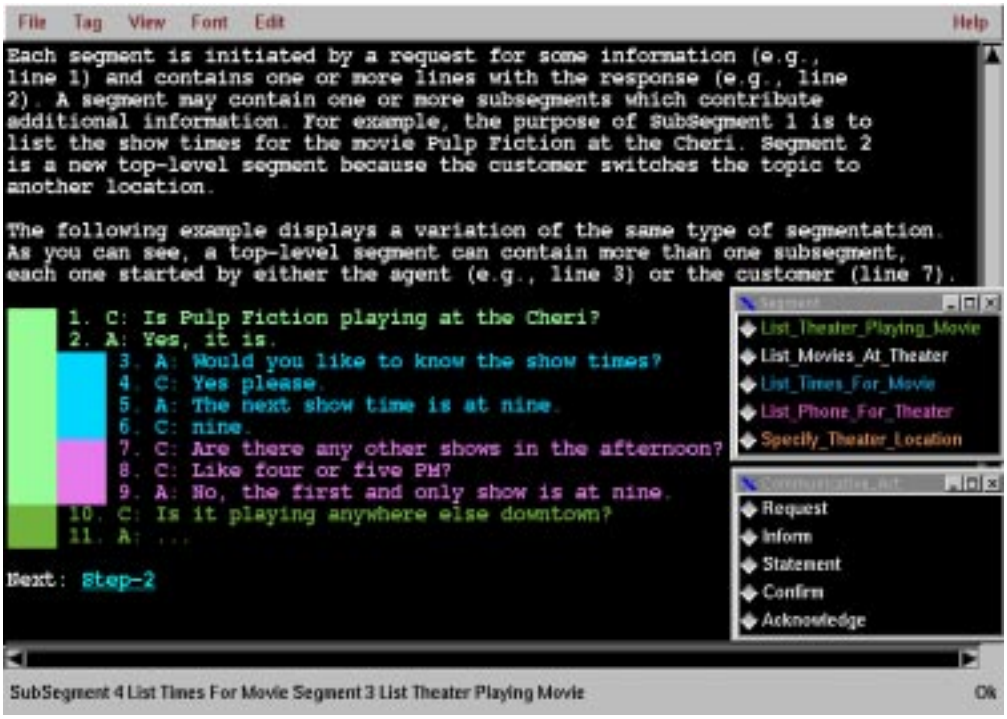



Figure 3-3: Screen shot of the latest version of *Nb* . The visual editing tool allows the user to annotate embedded discourse segments using a point-and-click interface. The user highlights the lines that she wants to annotate as a unit, and then selects the unit name for it from one of the menus at the bottom right corner. *Nb* automatically colors and indents the annotated units.

and indentation. There are two new features in the third release of *Nb* .

Firstly, it is not necessary to load and save files using the File menu. Instead, *Nb* has full hypertext functionality. This new feature allows the user to perform specific actions by simply clicking on highlighted text. Possible actions are displaying some relevant instructions in a separate window, loading the next file to be annotated or reviewed, or automatically sending all the annotated files by electronic mail to the administrator of the annotation experiment. Rather than relying on a printed instruction manual, the annotation instructions can be organized into a set of related hypertext documents and integrated directly into the annotation session.

Secondly, a new *drill* feature has been introduced. A drill is a file that has been annotated with some (hidden) reference mark-up tags, such as discourse segments. The reference tags are not visible to the user. The user can try to discover the tags by trial and error. *Nb* does not let the user tag the text unless the tag matches a reference tag. When a match is found, *Nb* enters the mark-up tag and tells the user how many tags are left. Hints can be integrated into a drill by providing hypertext links to the appropriate instruction screens.

### Using *Nb* to Annotate Text

In the first and second releases of *Nb* , coders read a printed instruction manual and used *Nb* to load, annotate and save files. In the third release of *Nb* , the coders navigate the on-screen instructions organized as a series of linked hypertext screens. The hypertext documents may contain instructions, annotated examples, and text to annotate. The annotated examples are in the same format as that of the actual annotation task (as displayed in Figure 3-3). After browsing the instructions, the user completes the annotation exercises and then annotates a set of text samples. The user can choose to view two *Nb* windows at the same time, one with an instruction screen and one with the exercises and the text to annotate. Finally, one hypertext link allows the user to send the annotated data to the experiment administrator by electronic mail.

*Nb* allows users to edit discourse units directly in the main window, using either the mouse or keyboard shortcuts. In the latest two releases, the user can browse large portions of the text in one screen. Annotations options are only displayed on demand with pop-up menus so as not to occupy a large portion of the screen. In addition, the user can quickly jump to any annotated section and edit or delete it. To annotate segments or communicative acts, the user can highlight the corresponding text with the mouse, and then select a unit type and value from a list (e.g., choose **Segment: List Times For Movies** or type the corresponding keyboard shortcut in Figure 3-3). For some coding schemes, the user can also type new tag units and values online while the annotation is in progress. It is also possible to listen to the speech signal in a time window corresponding to the highlighted text, if the speech waveform file is available.

The latest two releases of *Nb* include easy-to-use editing features for renaming and deleting mark-up tags. For example, Figure 3-4 demonstrates how to change the value of an existing tag. First, the user selects some text and clicks on the right mouse button. A window named "Edit Tags" pops up with a list of all of the mark-up tags in the selected text, and the user may change or delete one of them. In addition, *Nb* allows users to undo one or more of the most recently entered mark-up tags.

With *Nb* it is also possible to browse the text indexed by annotated units. For example, Figure 3-5 displays a pop-up window with a list of all the annotated units. If the user clicks on a unit in the list, the *Nb* main text window will display the corresponding text. Finally, to view very long segment units in one screen (up to 50 lines) the user can maximize the window and select a very small font size for display.

*Nb* enforces syntax constraints specified by the coding conventions. By default, segment boundaries are extended to the next end-of-line. Typically, the input text should have been already formatted into one clause per line. In addition, *Nb* can be set so as to optionally perform syntax checking while the annotation is in progress. For example, *Nb* can be set not to allow users to annotate units that cross each others boundaries. *Nb* may limit the level of embedding (e.g., only one or two levels

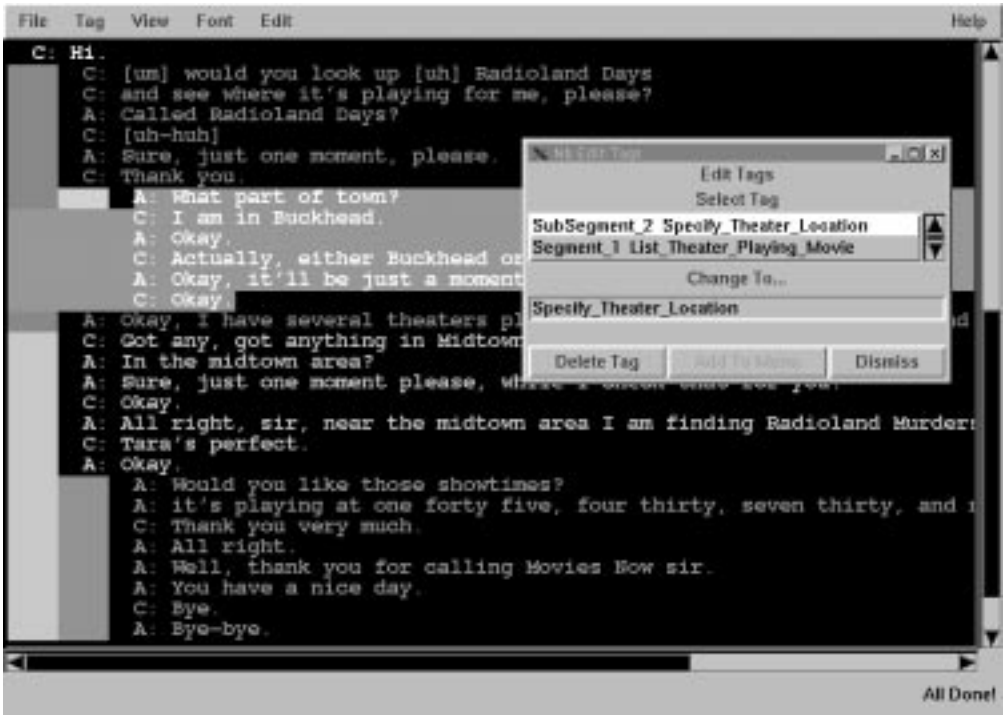



Figure 3-4: Editing tags with *Nb* . The user highlights some text with the mouse. A window pops up (on the right) with the list of all the mark-up tags in the highlighted text. The user can then select one and either change its value or delete it.

of subsegments) or it may force the annotator to tag top-level segments before tagging embedded subsegments. Another optional constraint is that a line may start only one segment or subsegment but not more than one. Finally, *Nb* can be set not to allow users to annotate a trivial top-level segment that contains the entire dialogue. While these constraints decrease the expressive power and the detail of the annotation, they simplify the annotation task for the coders. For example, if only one level of embedding is allowed, if a line can either start a segment or a subsegment but not both, and if the evaluation is mostly concerned with locating segment initiatives, the cognitive task for each line is essentially reduced to choosing among three independent alternatives: (1) starting a top-level segment, (2) starting a new embedded subsegment, or (3) continuing the current discourse structure. In addition, the annotated discourse structures are easy to display, and it is possible to evaluate inter-coder agreement using measures such as precision, recall, and the kappa coefficient (the kappa coefficient can only be computed for a finite set of mutually exclusive categories).

### Using *Nb* to Develop Coding Schemes

A coding scheme usually consists of three related specifications. First, the scheme must provide the list of all possible tags and their values. Second, it must provide the syntactic constraints (e.g.,

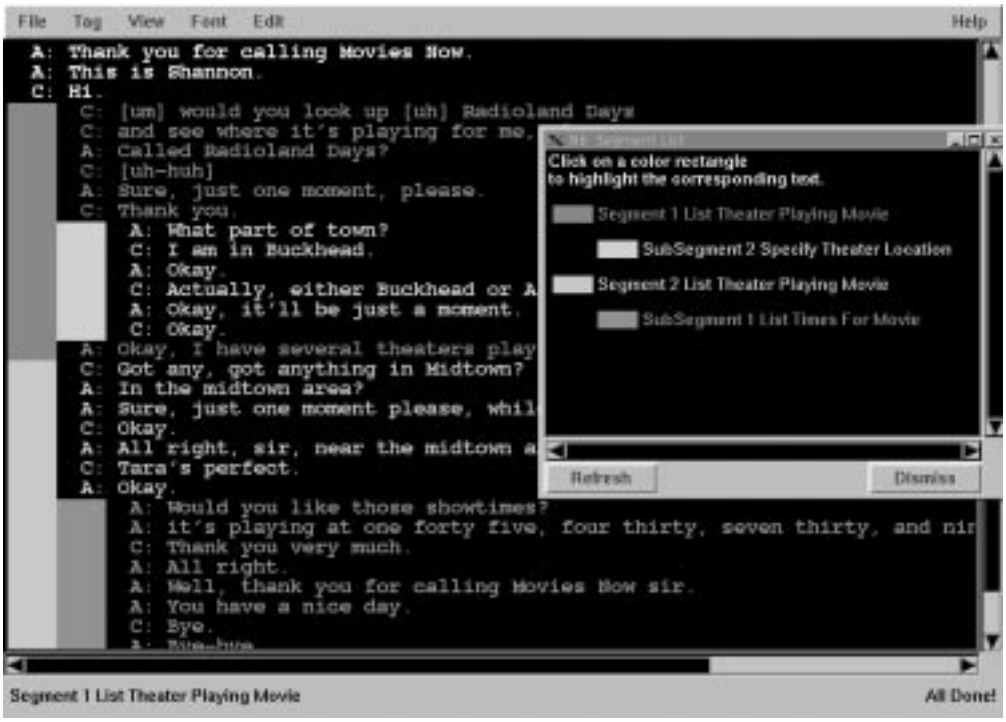



Figure 3-5: With *Nb* , it is possible to quickly jump to the text of an annotated segment by selecting it in the pop-up window (to the right) which lists all the annotated units. The main window will then display the corresponding text.

whether they can span words, phrases, lines, or multiple lines). Thirdly, the scheme must include an instruction manual which describes the meaning and usage of each tag, and provide examples of annotated text.

In the latest two releases of *Nb* , developing a new annotation experiment for *Nb* does not require the experimenter to update the source code or write the coding manual in a separate paper document. Instead, the instructor defines all of the coding scheme parameters and constraints in a configuration text file, and embeds instructions, annotated examples and text to annotate all together in a set of hypertext documents. The coding scheme must specify the units to annotate, such as discourse segment purposes and communicative act types. The coding units and values will be displayed to the coder by *Nb* in the various pop-up menus. For each unit, the configuration file should also specify a keyboard shortcut and a color in which the unit is to be displayed.

The designer of the coding scheme is free to organize the hypertext documents as she wishes. For the annotation experiments reported in Chapter 4, we found that the following blue print to provide easy-to-understand instructions that produced reliable annotations. First, the user is presented with a short tutorial about how to use *Nb* to enter and edit tags, and is introduced to the keyboard shortcuts and the color coding conventions. Second, the instructions provide many examples of

annotated text using the coding units specified in the configuration files. For each unit, we usually provide a series of examples as well as counter-examples drawn from real data similar to the data to be annotated. We also found useful to provide at least one fully annotated dialogue as a concrete example. The advantage of integrating all annotated examples into *Nb* is that the user sees the annotated examples exactly in the same display format as if she entered the annotation herself. Third, the coding scheme designer should write three to five exercise files. These are hypertext files that are annotated with reference tags that are recognized by *Nb* but not displayed.

## 3.3 Evaluation

In the following sections, we provide concrete examples of how *Nb* has been used for annotating text using different coding schemes. Collectively, the three releases of *Nb* have been used in combination with at least ten different coding schemes for annotating communicative acts and discourse segment units. We then summarize how well the different releases of *Nb* performed in terms of usability (how easy or difficult was it to annotate or display units?) and portability (how easy or difficult was it to develop different coding schemes?).

### 3.3.1 Examples of Coding Schemes Used with *Nb*

#### First Release

The first release of *Nb* has been used in the first experiment of this thesis, described in greater detail in Chapter 4. The experiment involved 18 text transcriptions, each annotated by five different coders. The task was to annotate embedded discourse segments. The segment purpose labels were subjective. The task was unconstrained, and allowed users to annotate segments to any level of embedding, provided that they did not cross each others' boundaries.

#### Second Release

The second release of *Nb* was downloaded by many researchers around the globe. It has been used for discourse segmentation experiments at Harvard University and at the first Discourse Resource Initiative (DRI) workshop at the University of Pennsylvania in March 1996.

At Harvard University, *Nb* has been used to annotate the discourse of direction giving monologues, by examining text transcriptions as well as listening to the corresponding speech signals. This coding scheme was rooted in the intentional theory of discourse structure proposed by Grosz and Sidner [38, 71, 45]. Although coders were free to choose the segment purpose labels, the instructions specified that these labels should reflect the *task* - *subtask* - *digress* structure of a dialogue, in which top-level tasks could be realized as a sequence of sub-tasks, with possibly embedded digression segments which would momentarily deviate from the purpose of the top-level segment.

| | | |
|---|---|---|
| Accept | Confirm | Deliberate |
| Digress | Feedback | Greet |
| Init | Introduce | Motivate |
| Reject | Request | Thank |

Table 3.1: List of twelve abstract communicative act types used in the Verbmobil coding scheme.

At the first DRI workshop, *Nb* was used to annotate five transcriptions of natural dialogues and two transcriptions of monologues, using seven different discourse coding schemes [59]. The five dialogues were extracted from the following corpora: Verbmobil (an appointment scheduling dialogue), Map Task (route finding on a map), Trains (shipping and transportation problem), and the corpus used for this thesis (a flight reservation dialogue and a yellow pages inquiry). The two monologues were extracted from the Boston Direction Corpus (walking directions to a landmark building in Boston) [72] and from the instruction corpus developed at the University of Pittsburgh (directions for assembling some electronic parts). Each text was annotated by three to eleven coders. While no formal inter-coder agreement results were reported, the annotated data served the purpose of informally assessing differences and commonalities between the coding schemes, and to initiate a discussion about how to unify and standardize all the different approaches. This was a pioneering workshop because, for the first time, researchers were gathered together to discuss and evaluate empirically different coding schemes on the same data.

The first coding scheme was the segmentation scheme mentioned above, proposed by Nakatani et al. from Harvard University.

The second coding scheme was the Verbmobil set of 12 speech acts, or dialogue moves appropriate for describing meeting scheduling dialogues [50]. The dialogue moves were organized in a hierarchy. At the top level, there were abstract dialogue moves appropriate for negotiating dialogues, which are listed in Table 3.1. Each one of the domain-independent moves were further specified with domain dependent values. For example, the moves **Introduce** and **Request** could have the following values:

- Introduce Name
- Introduce Position
- Introduce Location
- Request Suggest Data
- Request Suggest Duration
- Request Suggest Location

The third coding scheme was the Map Task coding scheme, which was developed for analyzing co-operative route finding dialogues based on the theory of dialogue games [16]. In this coding scheme, dialogues were annotated with two layers of related tags. Embedded segments were annotated

| *Move* | *Response* | *Other* |
|---|---|---|
| Suggests Action | Agrees with Suggestion | Discourse Marker |
| Requests Action | Disagrees with Suggestion | Metalanguage |
| Requests Validation | Complies with Request | Orientation of Suggestion |
| Requests Information | Acknowledges Only | Requests Refers To Personal |
| Elaborates | | Jokes Exaggerates |

Table 3.2: The three independent functions used to annotate each communicative act in the Condon and Chech coding scheme.

with the labels called *Transactions* and *Games*. Typically, transactions corresponded to top-level tasks and games to individual sub-tasks. In the Map Task route-finding dialogues, top-level tasks corresponded to goals such as going from one specific landmark to another. A transaction contained one or more embedded games. A game corresponded with one cohesive step in the route-finding process, such as crossing a bridge. Individual clauses in dialogue turns were also annotated by a set of 12 different communicative acts, or conversational moves, that were also conceptually organized as a hierarchy, depending on the intention of the speaker. At the top level, moves were divided into initiatives, responses and preparations. Initiatives were commands, statements and questions. Commands were marked as *Instruct* moves and statements as *Explain* moves. Questions, or requests, could be marked as one of *Align, Check, Query-yn, and Query-w*. The *Align* and *Check* moves were used for implicit requests for confirmation and clarification. The *Query-yn* and *Query-w* were used for explicit requests that required either a yes-no answer or a more complex answer. Responses were divided into *Acknowledgment, Clarify, Reply-y, Reply-n* and *Reply-w*. Preparation moves were tagged as *Ready*. A preparation move was a sentence such as *let's see* and *just a minute*.

The fourth coding scheme, developed by Condon and Chech [26], was organized into three independent functions. In this scheme, each clause was to be annotated using one, two or three different mark-up tags, called *Move, Response,* and *Other*. The motivation for this coding scheme was that a clause may simultaneously perform more than one intentional function in the dialogue context. It may initiate a new purpose (*Move* tag), be an appropriate response to some past move (*Response* tag), or be a meta-communication signal that is useful for setting a common ground and mutual understanding between the dialogue participants (*Other* tag). Table 3.2 lists all the possible annotation tags organized by function.

The fifth coding scheme was proposed by Traum and involved four different layers of mark-up [102]. In his proposed coding scheme, *Discourse Units* were discourse segment units that tagged sequences of one or more clauses that were related because they accomplished the same purpose, or task. Traum scheme, like Condon's scheme, recognized that a clause may perform multiple intentional functions. For this reason, a clause could be tagged with between one and four different tags. The different layers, displayed in Table 3.3 were: *Relatedness, Grounding Acts, Surface Form*

| *Relatedness* | *Grounding Acts* | *Surface Form* | *Speech Acts* |
|---|---|---|---|
| Explicit | Continue | Declarative | Inform |
| Related | Acknowledge | Imperative | Request |
| Unrelated | Repair | Question | Accept |

Table 3.3: Examples of communicative act labels organized in multiple layers as proposed by the Traum coding scheme.

and *Speech Acts*. *Relatedness* specified whether or not the clause was related to (or responded to) the immediate dialogue history. *Grounding Acts* were communicative acts that served the purpose of setting a common ground and mutual understanding between the dialogue participants. *Surface Form* specified how the communicative act was realized (e.g., Declarative vs. Interrogative vs. Imperative). Finally, *Speech Acts* were communicative acts common to the other coding schemes, such as *Request, Accept, Reject* and *Confirm*.

The sixth coding scheme, proposed by Moser, Moore and Glendening [68], was a segmentation scheme that was designed to combine the top-down intentional theory of discourse structure with the bottom-up approach of rhetorical structure theory. In this scheme, segments were labeled with their intentional purposes. Each segment contained one *Core* clause, which better summarized the purpose of the segment, and a set of related *Contribution* clauses which provided additional support for the core clause.

The seventh coding scheme was proposed by the author and it was a relatively unconstrained segmentation scheme, in which coders could annotate each text with embedded segments by giving a short title to each segment. In this coding scheme, the instructions were not as specific in the definition of a discourse segment purpose as in the coding scheme developed at Harvard, allowing coders to segment either according to intentional structure (what was the task that the speaker wanted to accomplish) or according to the topic structure (what were the topics - people, places, events - that were discussed by the dialogue participants).

### Third Release

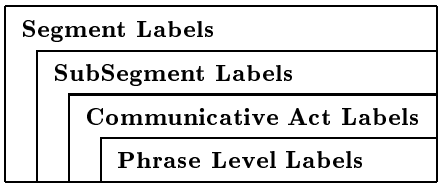
The third release of *Nb* has been used for the last three experiments described in Chapter 4 and the data analysis reported in Chapter 5. All experiments involved discourse segment units. The data were 23 text transcriptions of conversations about movie schedule listings in Atlanta. Each transcription was annotated by between five and nine different coders. In this set of experiments, discourse segment units reflected the *task-subtask* structure of a dialogue, rather than the topic structure. The units in the first experiment were linear segments with domain-specific purpose labels fixed a priori. The units in the second experiment were segments and subsegments with purpose labels also fixed a priori. The units in the third experiments were linear segments with no

labels. This release of *Nb* has also been used to annotate 190 movie schedule dialogues by an expert coder with segment units and communicative acts.

### 3.3.2 Usability

We evaluated the usability of *Nb* by conducting informal feedback interviews with coders after they completed each experiment. The first release of *Nb* was judged difficult to use because the editing was line-oriented while the task was oriented towards clustering larger units into embedded discourse segments. The latest two releases have been praised as extremely usable by the coders, when the task involved annotating either embedded discourse segment units or one layer of communicative acts. Users were able to concentrate on the difficult cognitive task of discourse annotation without spending much time learning to use *Nb* . For example, in the third release of *Nb* , it took two hours for a novice user to complete an annotation experiment with three exercises and ten dialogues of average length 50 lines. Users spent one third of the time respectively browsing the instructions, completing the exercises and performing the annotations. Users spent most of their time thinking about the discourse segment structure and only a small fraction editing the segmentations. The display and editing features of *Nb* allowed users to quickly test and correct different segmentation hypotheses.

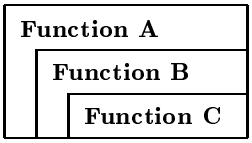
The feedback received from distributing *Nb* at the DRI workshop was in general very positive because for the first time, one editing tool allowed users to annotate text transcriptions using many different coding schemes. It also pointed out some limitations of the *Nb* display when tagging multiple layers of communicative acts. Two of the seven coding schemes (proposed by Condon and by Traum) specified that one clause could be tagged with up to three different units in any order. The problem in the display was that *Nb* assumes that mark-up tags are organized into a meaningful hierarchy of embedded units. *Nb* automatically indents and color-codes layered tags. This display is optimal for embedded non-overlapping tags that are organized as follows:




This assumption is very useful for automatically enforcing syntactic constraints and producing annotated data that are free from trivial errors. The display is not appropriate when the order and the layering of the labels is not important. This is the case for the Condon and the Traum coding schemes, which conceptually has a flat organization with no a priori order:

| Function A | Function B | Function C |
|---|---|---|

The display of *Nb* imposes a hierarchy in the different functions, which is not what the coding scheme intended:




Finally, it is not possible to edit and display co-reference relationships between two or more segments or communicative acts. The only way to tag relationships between two units is by giving them the same value (e.g., the same segment purpose label).

### 3.3.3 Portability

Over three releases, *Nb* has been used for developing at least nine different coding scheme, and the software, publicly available on the Internet, can run on any platform that supports the Tcl/Tk language. The experiments reported in this thesis as well as the ones developed at Harvard University demonstrate that it is relatively straightforward to develop new instructions for discourse segmentation according to many different theories, such as sub-topic structure vs. intentional structure. The annotation experiments for this thesis produced reliable results with an average kappa coefficient of 0.45 for the first release of *Nb* , and 0.82 for the latest release of *Nb* . At Harvard, *Nb* has been used for discourse segmentation using text and acoustic cues from written instructions, with very encouraging reliability results [71, 45]. When users were able to listen to the corresponding speech signal as well as browse the corresponding text, the kappa coefficient was found to be 0.83.

Although the feedback received from distributing *Nb* over the Internet was in general very positive, it also pointed out some limitations of *Nb* . The display limitations of *Nb* make it inappropriate for coding schemes that involve many different overlapping layers of mark-up tags, or tags that are linked to each other because they refer to the same semantic or intentional entity. Also, while *Nb* can read and write annotated files that are compatible with SGML, it does not parse generic SGML and XML tags.

### 3.3.4 Toward a Generic Annotation Tool

While *Nb* has allowed us to quickly develop and deploy discourse segmentation experiments for this thesis, it is not a generic discourse annotation tool. Building a generic discourse annotation tool that accommodates all possible coding schemes is beyond the scope of this thesis. The experience gained from developing *Nb* provided us with very valuable insights about the requirements for a generic annotation tool. A generic annotation tool should be able to process and display generic files annotated using SGML, or at least XML, and provide multiple alternative displays of the same data. One important open problem is that one coding scheme may involve multiple layers of mark-up

tags that overlap with each other. For comparison purposes, it may be useful to display different coding schemes applied to the same data. For these reasons, we feel that a generic tool should provide for more than one type of possibly overlapping display. Each display can be tailored to one specific coding scheme, such as discourse segmentation, dialogue acts, relationships between segments and acts, and co-reference. Finally, it may be desirable to tag segments and clauses with entities that have more than one property such as syntactic, semantic and intentional features, and co-reference pointers to other entities.

# Chapter 4

# Producing Reliable Segmentations

In this chapter we report on four different discourse segmentation experiments with many different trained coders. The results of the first experiment have been reported in [31] and the results for second experiment have been reported in [33]. This chapter is a substantial revision and extension of the work reported there. In each experiment, the training consisted of reading some instructions and annotating the dialogue transcriptions using the *Nb* annotation tool. The objectives of the inter-coder agreement experiments are twofold. First, we want to assess what type of segmentation coding scheme is appropriate to reach a satisfactory level of agreement among coders, defined by a kappa coefficient of at least 0.7. Segmenting text reliably is a difficult problem in computational linguistics. Recall from Chapter 2 that while there are many valuable theories of spoken dialogue, the problems in segmenting it reliably have largely remained unexplored. Only one coding scheme has reported a reliability score (0.59) for segmenting the Map Task dialogues [16]. In contrast, two communicative act coding schemes report reliability levels of better than 0.80 for annotating the Map Task and the Switchboard corpora [16, 51]. Three out of four published segmentation schemes, applied to monologues and science articles, report lower reliability scores, between 0.59 and 0.67 [45, 16, 41]. The best reliability score (0.80) has been obtained by three expert coders annotating spoken monologues using intentional discourse structure theory, by looking at the text transcription as well as listening to the corresponding speech signal [45]. Second, we want to provide concrete examples of where coders disagree and where they agree in placing segment boundaries. We would like to determine empirically the extent to which these areas of agreement and disagreement support the hypotheses set forth by discourse structure theories based on intentions and discourse contributions.

Table 4.1 provides a road map of the four experiments. While we believe that discourse annotation can be done more reliably using text *and* speech, two important practical considerations limited the experimental conditions to using the text transcriptions only. First, the acoustic signal corresponding

| *Experiment* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| *Dialogues* | 18 | 22 | 12 | 12 |
| *Coders per Dialogue* | 5 | 7-9 | 5 | 5 |
| *Domains* | Many | One | One | One |
| *Detailed Instructions* | No | Yes | Yes | Yes |
| *A Priori Labels* | No | Yes | Yes | No |
| *Embedded Segments* | Yes | No | Limited | No |
| *% Recall* | 61.5 | 83.9 | 74.9 | 75.0 |
| *% Precision* | 57.7 | 85.0 | 73.4 | 78.0 |
| *Kappa* | 0.45 | 0.82 | 0.70 | 0.71 |

Table 4.1: Comparison of the experimental conditions and summary of the results for the four inter-coder agreement experiments discussed in this chapter.

to each dialogue was not available when we started the experiments. A considerable effort was going to be necessary to provide a segmented and indexed database of all of the acoustic signals time-aligned to the corresponding text transcriptions. Second, we wanted to reach as many potential users as possible. Users were able to quickly download and install *Nb* on any Unix and Windows desktop machine. They could immediately start an annotation session and *Nb* would automatically send the annotated data to the experiment administrator at the end of the session. Providing acoustic waveform playback on multiple platforms and supporting multiple sound cards could have increased considerably the complexity of developing *Nb* . By keeping the *Nb* software package small and multi-platform, we were able to enroll 46 coders annotating 40 dialogues.

In all of the experiments, each dialogue was annotated by at least five and at most nine coders. Our goal was to assess reliability by enrolling as many paid volunteers as possible. We consider that five different segmentations per dialogue are sufficient to identify trends of agreement and disagreement among coders and provide significant results. In all of the experiments, coders were able to browse the entire dialogue and annotate segments with no predermined order. The first experiment was the least constrained, allowing coders to annotate embedded discourse segments according to their best judgment, without providing them with a specific definition of a discourse segment. That experiment was designed to explore the complexity of the problem of obtaining reliable segmentations. We specifically decided *not* to guide coders toward a discourse segment theory. Our goal in this experiment was to find out if the segmentations produced by a majority of coders provided support for or against a particular theory of discourse. The three subsequent experiments were more focused and explored different ways of directing the annotations toward an intentional discourse structure, applied to one particular information-seeking domain. We decided

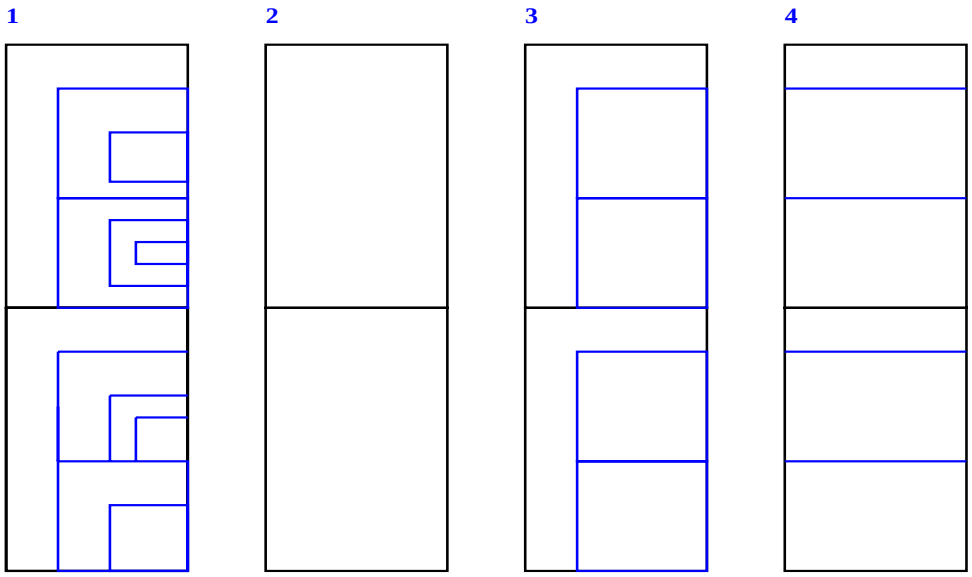



Figure 4-1: Diagram illustrating the differences in type of segmentation explored in the four annotation experiments. From left to right: 1: Unconstrained segmentation. 2: Linear segmentation with a priori labels. 3: Limited embedded segmentation. 4: Linear segmentation with no a priori labels.

to focus on the movie schedule domain for two reasons. First, we are interested in building spoken language systems for domains of the same complexity as the movie schedule domain. Second, we argue that conversations in this relatively simple information-seeking domain have a rich turn-taking structure which is appropriate for studying fundamental issues in discourse analysis. The constraints in the instructions varied among the last three experiments: the structure was either linear or hierarchical, and the segment labels were either chosen from a small set of predefined purpose names or freely defined by the coders.

Figure 4-1 illustrates the difference in the number of layers of the segmentations in the four experiments. Going from left to right, in the first experiment coders were free to choose the level of detail in the segmentation and the segment labels. In the second experiment, the segmentation was linear. It captured only switches in purposes using a predefined set of purpose labels related to the task domain. In the third experiment, coders were allowed to use at most one level of embedding using the same set of purpose labels as in Experiment 2. Finally, in the fourth experiment, coders were asked only to place segment boundaries at switches in segment purposes, without specifying the purpose. In the rest of the chapter, we describe in detail and discuss the outcome of each experiment.

## 4.1 Experiment 1: Unconstrained Segmentations

### 4.1.1 Data and Task

We selected 18 representative recorded telephone dialogues from our corpus. The domains of the conversations are air travel planning, Yellow Pages inquiries and classified ads. The subjects of the conversations range among the following topics: confirming ticket reservations, obtaining air fares, seeking addresses and phone numbers of business listings in the Atlanta area, looking up the job offers section and the car classified ads of the *Atlanta Journal and Constitution*, and looking up restaurant and movie listings. The dialogues ranged in length from 20 to 50 dialogue turns, with an average of 34 dialogue turns. It was formatted a priori into one clause per line (one dialogue turn typically included one to three clauses). The average number of clauses per dialogue was 50. Clause end-points were determined automatically based on punctuation marks. Special markers were used in the transcriptions to indicate speech overlap. Pauses were indicated by three or more periods. The number of periods was loosely correlated with the perceived pause duration in seconds. All annotations were completed from the text alone, without listening to the corresponding speech signal.

This experiment was unconstrained in nature. The task was to bracket each dialogue with possibly embedded segment boundaries placed between clauses, using the first release of *Nb* . Our goal was to determine by cluster analysis techniques what types of discourse patterns were agreed upon by a majority of coders without extensive instructions. This data-driven approach was proposed by Rotondo in 1984 to assess inter-coder agreement in segmenting text [84]. In addition, we wanted to determine empirically where coders would disagree and what types of instructions and tutorial examples were needed to achieve more reliable results. We did not expect to obtain substantial agreement among coders without giving them specific directions. The annotation instructions were minimal and were not biased towards any particular discourse theory. Each segment was annotated by giving it a title. Coders could type in any title they wished. They were not given the definition of a segment, except that a segment should correspond to one individual topic. Coders were free to define the number and the level of embedding of the segments, and a clause could open or close one or more segments. The only syntactic constraint enforced by *Nb* was that segments could not cross each others' boundaries. Each coder completed the task with no feedback from others.

### 4.1.2 Coders

The coders were researchers and graduate students in electrical engineering, computer science, and cognitive sciences. Fifteen paid volunteers participated in the experiment. Four of them were previously exposed to discourse analysis literature. Each coder annotated 6 dialogues for a total of 90 annotated texts. Thus each one of the dialogues has been annotated by 5 different coders. Each

text transcription, averaging 34 dialogue turns, was annotated in 15 to 20 minutes.

### 4.1.3 Agreement Statistics

To allow comparisons with other annotation studies, we decided to report average pairwise precision, recall and kappa coefficient in placing segment boundaries. We decided to focus on assessing the agreement in placing the starting clause of a new segment, because this task is crucial in linear discourse segmentation (i.e., in the evaluation we did not consider where segments ended or differences in segment purpose labels). The statistics have been computed by clustering each clause into one of two categories: *boundary* - if the clause was annotated to start one or more segments, or *not boundary* - in all other cases.

The total number of clause samples used to gather statistics was 902 and the total number of dialogue turn samples was 626. On average, the most prolific coder opened a new segment every 3.5 clauses, while the least prolific coder opened a new segment every 5 clauses. At the clause level, the average pairwise precision was found to be 57.7% and the recall was 61.5%. At the dialogue turn level, precision increased to 67.0% and recall increased to 71.0%.

The kappa coefficient can be used to assess reliability when each text unit is classified into one of a finite set of categories. Under this condition, it is possible to compute the chance agreement as the product of marginal distributions for each individual category. The coefficient is not directly applicable to assessing agreement between embedded bracketings of the same text. When applied to the two way *boundary vs. non boundary* classification task, the group-wise kappa was 0.45 for clause units and 0.54 for dialogue turn units.

### 4.1.4 Agreement Trends: Openings, Closings, Tasks and Contributions

For each annotated dialogue, we selected empirically the majority rule segmentation as well as the segments proposed by a minority of coders by using a simple splitting algorithm. All hypothesized segments by all coders were put in an input list. This list was used to produce two different lists. In the first one (the majority list), we only kept the segments for which both end-points were proposed by at least three out of five coders, and that did not cross boundaries with other segments in the majority list. Figures 4-2, 4-3, 4-4 and 4-5 display the output of this procedure for four representative dialogues. The second list was used to collect the segments that were proposed by only one or two coders. The majority list was used to determine patterns of agreement among coders, while the minority list was used to determine patterns of disagreement. We examined each majority and minority list along three discourse relations: openings and closings, task-subtask structure, and contributions to discourse.

[ 1] A: How can we help you?
[ 2] C: Hello.

[ 3] C: I'm wondering if you can give me a fare from Albuquerque to Detroit on the twenty-seventh of July?
[ 4] A: Okay.
[ 5] A: I'll sure check for you.
[ 6] C: Thank you.

[ 7] A: Will this be a one-way or round trip for you?
[ 8] C: Uh, round trip--
[ 9] A: Right.
[ 10] C: --Returning the first of August.
[ 11] A: First of August.

[ 12] A: How many will be going with you?
[ 13] C: Uh, one person.
[ 14] A: At one person.
[ 15] A: Let me check.
[ 16] A: Okay.
[ 17] A: That will be a Wednesday, back on a Monday.

[ 18] A: I've got a promotional fare on that one available.
[ 19] A: And that will be a total ticket of three-hundred-ninety-one dollars round trip.
[ 20] A: And that includes all airport charges and all taxes.
[ 21] C: Okay.

[ 22] A: Did you wish to leave in the morning or afternoon?
[ 23] A: And I'll check what schedules are available for you.
[ 24] C: Uh, no, um, let me get back to you, sir.

[ 25] C: Thank you very much.
[ 26] A: Alright.
[ 27] A: Well, let us know as soon as possible,

Figure 4-2: Experiment 1: majority segmentation of a flight reservation dialogue. Embedded segments are represented by embedded boxes that enclose sections of text.

[ 1] A: Thank you for calling Movies Now.
[ 2] A: This is Shannon.
[ 3] C: [Yuh] do

[ 4] C: is there a [ah] number that you dial to just get all the different theaters?
[ 5] A: I can give you that information.
[ 6] C: You can?

[ 7] C: Okay, in Snellville, Septum Movies.
[ 8] A: Sure, just one moment please...
[ 9] A: And that was the Septum Theater?
[ 10] C: [Yeah].
[ 11] A: Okay...
[ 12] A: Okay, I have a Cineplex Odion in Snellville.
[ 13] A: Do you know that there is one called the Septum?
[ 14] C: [Yeah], it's called Septum, but Cineplex Odion is probably the theater who owns it, but it's called the Septum Theater, but [yuh],
[ 15] C: [yeah] that's it.
[ 16] A: Okay [heavy_breathing] playing there, I have...
[ 17] A: this would be the one at Highway seventy eight at Walton Court?
[ 18] C: Yes.
[ 19] A: Okay.
[ 20] A: Playing there is Exit to Eden, Little Giant, Only You, The River Wild, The Specialist, Wes Craven's New Nightmare,
[ 21] A: and that'll do it.
[ 22] C: Okay, thanks.

[ 23] C: How about [um], the Town Center Cinema's in Lawrenceville.
[ 24] A: Okay, sure...
[ 25] A: Alright, playing there, we have Love Affair

[ 26] C: What time is that?
[ 27] A: Love Affair is twelve, two thirty, five, seven thirty, and ten.
[ 28] C: Okay, two thirty and five.
[ 29] A: Right.
[ 30] C: Okay.

[ 31] A: Okay, do you want me to read the rest of the movies?
[ 32] C: [Um] no, that's fine.

[ 33] A: Okay?
[ 34] C: Thank you.
[ 35] A: Thank you for calling.
[ 36] C: Bye.

Figure 4-3: Experiment 1: majority segmentation of a movie schedule dialogue.

[ 1] A: This is Ashley.
[ 2] C: [um] Hi, Ashley.

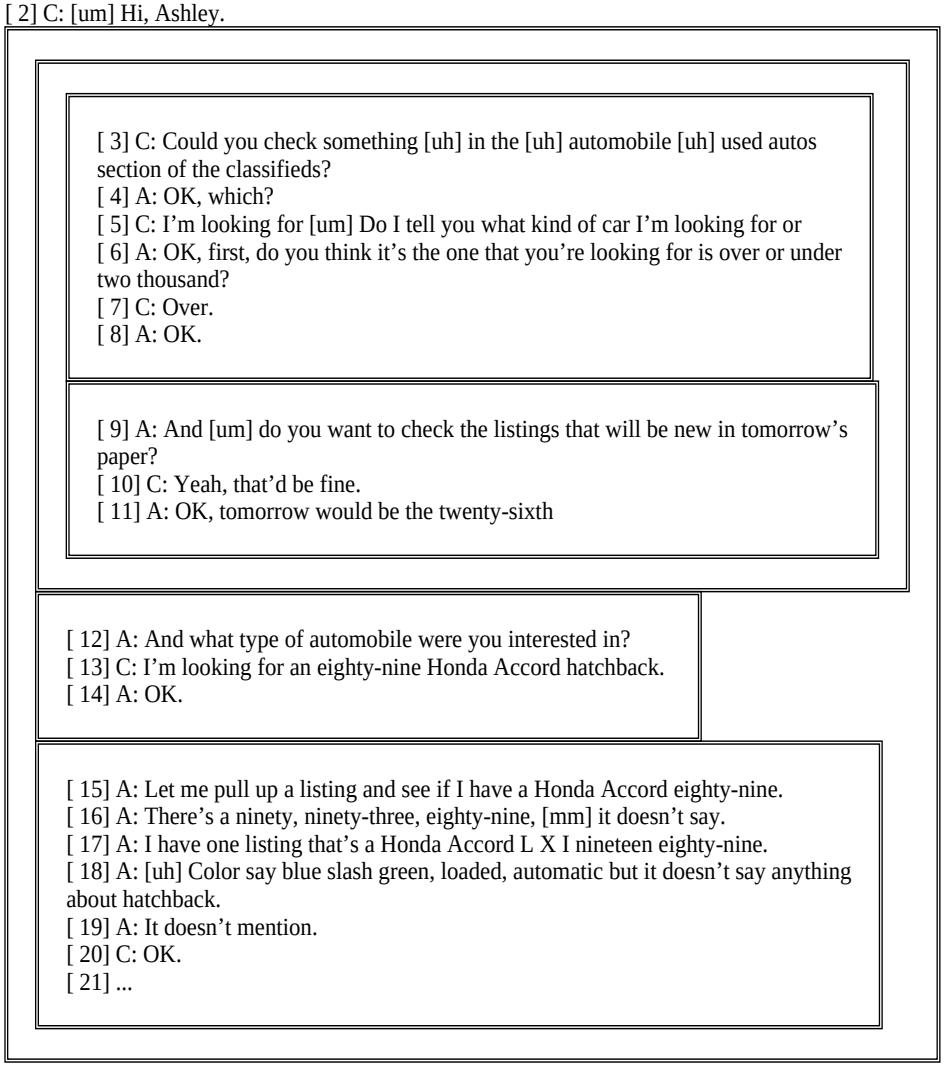



Figure 4-4: Experiment 1: majority segmentation of the first section of a automobile classified dialogue.

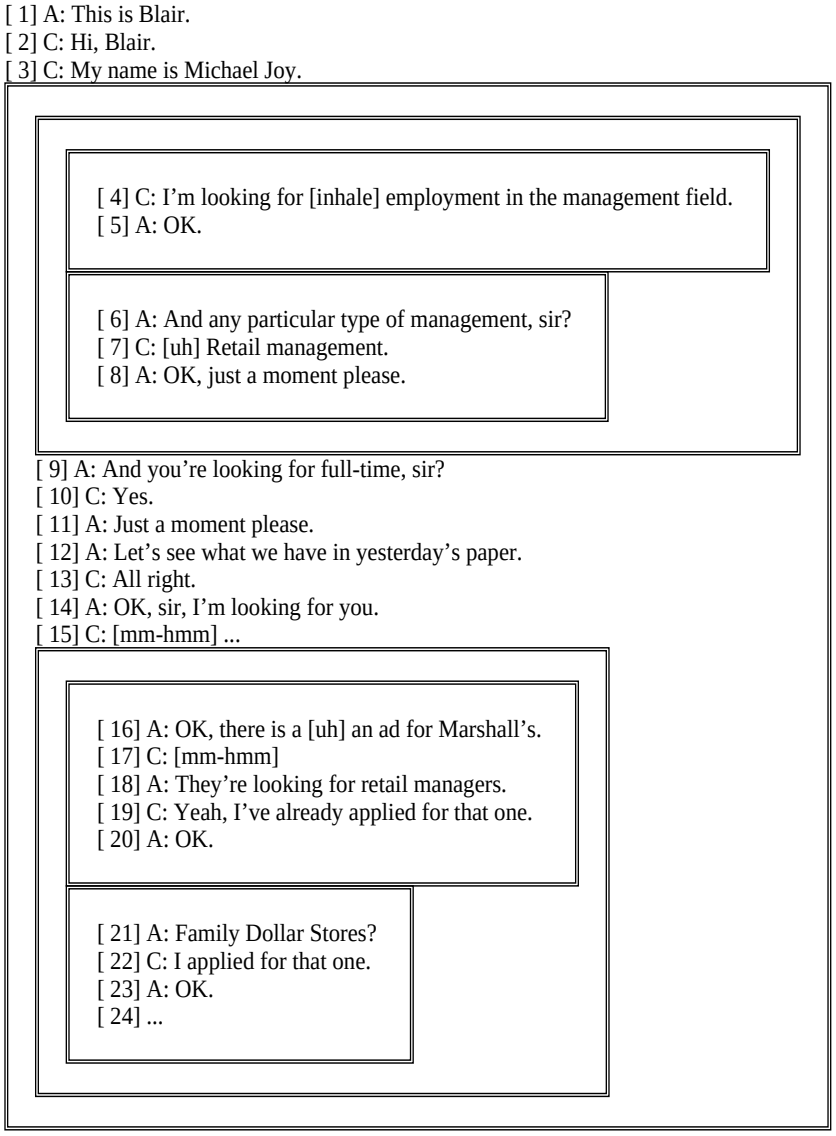



Figure 4-5: Experiment 1: majority segmentation of the first section of a job classified dialogue.

### Openings and Closings

All of the segmentations had a common conventional structure: *greetings - body - closings*. The dialogues were always introduced by an open-ended introduction by the agent (e.g., *This is Demita, how can we help you?*) and concluded with at least two turns of salutations. The majority segmentations reflected this pattern by including an initial and a final segment with dialogue turns that had no task-specific content. However, we also found that at least 15% of the minority segments included openings and closings as well. In the following example, some coders started a task-related segment at line 4, while other coders started it at line 7:

| | |
|---|---|
| 1 | A: Thank you for calling movies Now |
| 2 | A: This is Shannon |
| 3 | C: [Yuh], do |
| **Preliminaries (pre-sequence)** | |
| 4 | C: is there a [ah] number that you dial to just get all the different theaters? |
| 5 | A: I can give you that information |
| 6 | C: You can? |
| 7 | C: Okay, in Snellville, Septum movies |

The variability in determining the exact location of the start of the task-related conversation can be explained by what some researchers have called preliminaries, or pre-sequences [91, 85, 21, 48]. Pre-sequences are discourse transitions that are used by conversation participants to set a general common ground before committing to a more specific task. Transitional pre-closing statements also posed some problems in locating the end of the task-oriented part of the conversation. In the following example, some coders let the closing sequence start at clause 24, while some others placed the start at clauses 25 or 26:

| | |
|---|---|
| 24 | C: Uh, no let me get back to you sir |
| 25 | C: Thank you very much. |
| 26 | A: All right |
| 27 | A: Well, let us know as soon as possible |

Clause 24 serves two purposes. On the one hand, it closes the preceding segment with a negative response to the agent's request. On the other hand, it initiates a closing segment by signaling the customer's wish to end the conversation. Polite closings are an instance of communicative acts that serve two purposes: one is of acknowledging the information received, and the other one is to signal a switch in the task structure of the dialogue. This double purpose structure introduces some ambiguity in the segmentation. The acknowledgment should be part of the preceding segment, while the signal serves as an opening for the next segment.

### Task-Subtask Structure

The segment boundaries produced by the majority lists were located before changes in the task structure of the dialogue, as predicted by discourse structure theory [38, 39]. For example, the majority segmentation of the flight information dialogue displayed in Figure 4-2 is consistent with the following task-subtask structure:

1. Get fare information *(clauses 3-21)*:
   (a) Request fare information *(3-17)*:
      - Select departure city, arrival city, and departure date *(3-6)*.
      - Select round trip vs. one way and return date *(7-11)*.
      - Select number of tickets.
   (b) Report fare information *(18-21)*.
2. Book flight:
   (a) Select time of departure.

The correlation between discourse segment boundaries and changes in the task-subtask structure is present in all of the different domains of the annotated dialogues. However some of the majority segmentations are sequential in nature and do not display a clearly defined nested hierarchical structure. For example, the majority segmentation for the car classified dialogue displayed in Figure 4-4 is consistent with the following flat structure:

- Select price range (over or under $ 2,000) *(pre-sequence at lines 3-5, content at lines 6-8)*.
- Select publication date *(lines 9-11)*.
- Select year, make and model *(12-14)*.
- Report the classified ads *(15-20)*.

Many coders tended to annotate with two separate subsegments the initial part of a task and the final part of a task. Although segment boundaries were correlated with changes in the task structure of a dialogue, there was not always an exact one-to-one mapping between annotated discourse segments and task-subtask structure. Sometimes, a single segment would accomplish two or more related subtasks, such as selecting the year, make and model of a car. Some other times a single task would be accomplished over multiple subsegments, each containing one or more related subtasks. However, such subsegments were not be annotated by all coders as individual segments. Insertions and omissions of embedded subtask and digression segments accounted for about 30% of the minority lists, indicating that different coders applied different levels of detail in annotating segments and subsegments.

### Contributions to Discourse

According to Clark and Schaefer's theory of dialogue [22] conversations are organized into nested sequences of contributions. A contribution can be viewed as the smallest possible discourse segment unit. Typically, a contribution contains two phases: a presentation phase (e.g., a statement or a request by one speaker) and an acceptance phase (e.g., an acknowledgment, a confirmation or some other appropriate response). The annotated discourse segments in the majority lists consistently corresponded to one or more discourse contributions. Consider the following two annotated segments:

| | | | | |
|---|---|---|---|---|
| 16 | A: OK, there is a [uh] an ad for Marshall's. | | Present. | Segment 1 |
| 17 | C: [mm-hmm] | | Accept. | |
| 18 | A: They're looking for retail managers. | | Present. | |
| 19 | C: Yeah, I've already applied for that one. | Present. | Accept. | |
| 20 | A: OK. | Accept. | | |
| 21 | A: Family Dollar Stores? | | Present. | Segment 2 |
| 22 | C: I applied for that one. | Present. | Accept. | |
| 23 | A: OK. | Accept. | | |

The segments above are a good illustration of the fact that reporting some information may require several steps to be coordinated between the agent and the customer. The first segment, or contribution, is from turn 16 to turn 20. At turn 16, the agent presents an ad for evaluation to the customer. At this time, the information presented is generic. At turn 17, the customer accepts the presentation by indicating his understanding of the agent's sentence, and implicitly invites the agent to provide more specific information. At turn 18 the agent specifies more information about the ad. At turn 19, the customer not only acknowledges understanding, but also signals to the agent that this is not new information, implying that it is not necessary to continue reading the ad. At turn 20, the agent acknowledges understanding the customer statement, and complies with the implicit request by initiating a new contribution.

The third contribution is from turn 21 to turn 23. At turn 21 the agent proposes for evaluation another ad, by simply naming a company and presenting it as a question. Then, turns 22 and 23 mirror the sequence of turns 19 and 20. The second segment has the same nested presentation - acceptance structure as the first segment. Incidentally, it is not by chance that each turn in the second segment is much shorter than the corresponding turn in the first segment. According to Clark, contributions are made with the participants obeying the rule of least effort [21]. This rule is consistent with Grice's conversational maxim stating that dialogue participants should avoid unnecessary verbosity [34]. Since the second segment has the same intentional structure as the first one, the speaker only needs to speak enough words to signal to the listener what type of contribution he is making, omitting redundant information.

The majority of coders tended to annotate discourse segments that contained at least one discourse contribution (i.e., one presentation dialogue turn by one speaker followed by one acceptance phase spoken by both speakers). In addition, no segmentation proposed by the majority of coders crossed boundaries with a discourse contribution. However, different coders applied different levels of detail in the analysis of discourse contributions. Task-related segments may contain one or more discourse contribution and contributions can be nested. Some coders annotated each contribution separately, while others merged more than one contribution into a single segment. Omissions or insertions of individual contributions accounted for at least 20% of the disagreements among coders. Finally, some coders annotated sections of the acceptance phase of a contribution as a separate embedded segment, as in the following example:

| |
|---|
| 14 A: At one person |
| 15 A: Let me check |
| 16 A: Okay. |
| 17 A: That will be a Wednesday, back on a Monday. |

### 4.1.5 Discussion

We consider the results of the first experiment to be encouraging. The patterns of agreement in the unconstrained annotations provided some empirical evidence of the theories of discourse contributions and of discourse segment organization along tasks and subtasks. The patterns of disagreements provided some very useful insights about future experiments. The low scores for the agreement metrics (precision, recall, and kappa) suggested that in order to obtain more reliable results, specific instructions were needed. In particular, three problems that emerged from this study were the following. Firstly, different coders tended to annotate dialogues with different levels of detail. Secondly, some coders tended to separate the presentation and acceptance phases of a contribution into different segments, while others kept them in the same unit. Thirdly, coders tended to disagree about where exactly the task specific body of the conversation started, and where exactly it ended.

These problems have motivated a set of three guidelines that have been used for the three subsequent experiments. First, we decided to fix a priori the level of detail in the segmentation by constraining the task to be either linear segmentation with no embedded segments, or by limiting the maximum allowed depth in the segmentation. Second, to avoid splitting discourse contributions into separate segments, the instructions would provide extensive examples and counter examples of clauses that could start a segment. For example, direct and indirect requests for information may start a new segment, while acknowledgments, repetitions, confirmations and responses may not start a new segment. The guidelines would specify how to include the presentation and acceptance phase of a contribution in the same segment. Third, to avoid disagreements around greetings and

closings, we would provide specific guidelines about how to determine where the task-specific part of the conversation would start and end. In particular, we decided that greetings and open-ended or generic prelimiary sequences would belong to the first segment, while the second segment would not start until a specific request for information was spoken. Closings should simply continue the last open segment, rather than starting a separate closing segment. In addition, since the patterns of agreement were strongly correlated with the task-subtask structure of the conversations, we decided to introduce task oriented labels in the coding scheme.

The following sections present and discuss three additional annotation experiments. In all of these experiments, we decided to use conversations from one particular application domain (movie schedules) to eliminate sources of variability in the data. In Experiment 2, we limited the task to linear segmentation using a predefined set of task specific segment purpose labels. In experiment 3, we weakened the linear segmentation constraint by allowing at most one level of nesting, and in Experiment 4 we eliminated the constraint of working from a predefined set of segment purpose labels, but reinstated the linear segmentation constraint.

## 4.2 Experiment 2: Directed Linear Segmentation

### 4.2.1 Data and Task

For this experiment we selected 23 dialogues from the movie schedule domain. One dialogue was the same as in Experiment 1, but all the others were new. Using release 2 of *Nb* , coders were asked to segment linearly the transcription of the conversations that was presented to them one clause per line. The average text length was 40 clauses. The coders could choose among five different segment purpose labels:

- **List Theater Showing Movie**
- **Specify Theater Location**
- **List Movies At Theater**
- **List Show Times**
- **List Phone Number**

The complete text of the instructions is reported in Appendix A. Unlike the first experiment, this experiment included extensive online instructions. In particular, the instructions included many examples of opening clauses for the five segment types, as well as many examples of acknowledgments and other responses that should not start a new segment. Specific directions were given about greetings and closings. In particular, greetings should not be annotated, and the first segment should start at the most specific initial request for information. Closings should always belong to the last open segment and should not be annotated separately.

Another innovation in this experiment was that the instructions were followed by a set of four online annotation exercises. In each exercise, a movie schedule dialogue had been annotated a priori with some segments that were not visible to the coders. The goal of each exercise was to discover the segmentation by trial and error. *Nb* would provide minimal hints and only display an annotated segment when it matched the underlying reference segmentation. All coders found the set of exercises extremely useful for learning the segmentation model proposed in the instructions. The exercises were very useful in providing examples of the desired level of detail of the annotations.

After browsing the instructions and completing the four exercises, coders could annotate an assigned set of dialogues. When the annotations were completed, coders could select to send all annotated data by electronic mail directly from the annotation tool to the administrator.

### 4.2.2 Coders

As with Experiment 1, the coders were 23 paid volunteers. All except one were graduate students in electrical engineering and computer science. No one had participated in Experiment 1. Six were members of the Spoken Language Systems group and had some experience with spoken corpus collection and transcription. One of them was a member of the MIT Artificial Intelligence laboratory and was conducting a doctoral thesis on speech repairs. Another one was member of the Boston University Speech Group. Finally, one coder was working at a local speech recognition company, and had experience in collecting and transcribing telephone speech corpora. All but two coders had no previous knowledge of theories of discourse and dialogue.

Interestingly, we also attempted to open the enrollment to any willing MIT undergraduate student. Three students immediately responded to our public notice by sending the annotated data late that same night. We estimated the reliability of their coding by computing the group-wise kappa coefficient and found it to be less than 0.6. We decided to limit the enrollment to graduate students.

The coders were organized into three groups. The first group, consisting of nine coders, annotated seven different dialogues. The two other groups of seven coders each annotated eight different dialogues, for a total of 23 different dialogues, each annotated by at least seven coders. On average, each annotation session lasted two hours, with the last 30 minutes dedicated to annotating the assigned set of dialogues.

### 4.2.3 Agreement Statistics

For each dialogue and each pair of coders, we have evaluated the agreement in placing segment boundaries and labeling segments with segment purpose labels. Agreement in placing segment boundaries has been measured using precision and recall. For each pair of coders, we counted the number of boundaries that were proposed by both of the coders as well as the boundaries proposed by only one of them. We then computed precision and recall values using each coder in turn as

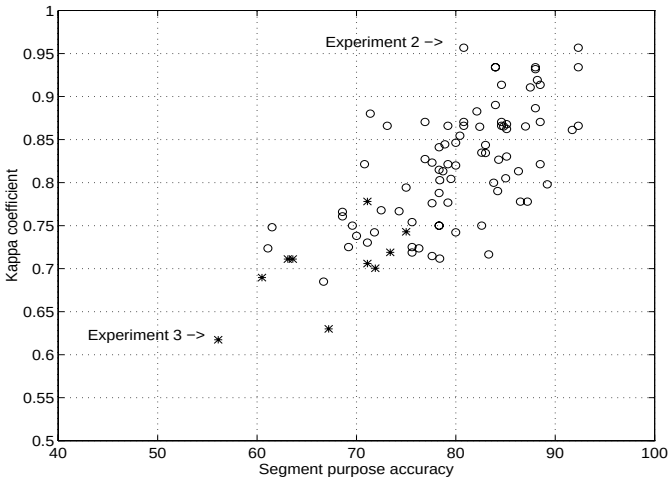



Figure 4-6: The pairwise kappa coefficient as a function of the segment purpose accuracy, for Experiment 2 (circles) and Experiment 3 (stars).

reference. Allowing a segment boundary to be placed at any clause unit, over all pairs of coders, the average precision was 85% and the average recall was 83.9%. The average group-wise kappa coefficient (0.82) confirmed the reliability of this result. Statistics for dialogue turns were somewhat higher, but not significantly different. The average precision was 85.1%, the average recall was 84.7%, and the kappa coefficient was 0.824.

Agreement in labeling segment purposes has been computed by extracting the sequence of segment purpose symbols and running the NIST alignment program on each pair of symbol sequences. Agreement between two coders has been evaluated as the symbol accuracy, defined as the difference between the number of matched symbols and the number of inserted symbols. The average pairwise symbol accuracy for segment purposes was 80.1%.

Figure 4-6 is a scatter plot of the pairwise kappa statistics as a function of the segment purpose accuracy for Experiment 2 and Experiment 3. Each point in the plot is the agreement between two different coders on the dialogues they both annotated. The horizontal axis is the segment purpose label accuracy, while the vertical axis is the kappa coefficient. The two dimensions are positively correlated because inserting two boundaries corresponds to inserting a segment purpose label. Except for one outlier in Experiment 2 and three outliers in Experiment 3, all of the kappa coefficients are above the reliability threshold of 0.70, with maximum values reaching 0.95. The segment purpose accuracy ranged between 62% and 93% for Experiment 2.

### 4.2.4 Agreement Displays

We applied a visual clustering technique to display the patterns of agreement in segmenting the dialogues. The technique is based upon the discourse analysis study by Rotondo [84]. Let $N$ be the

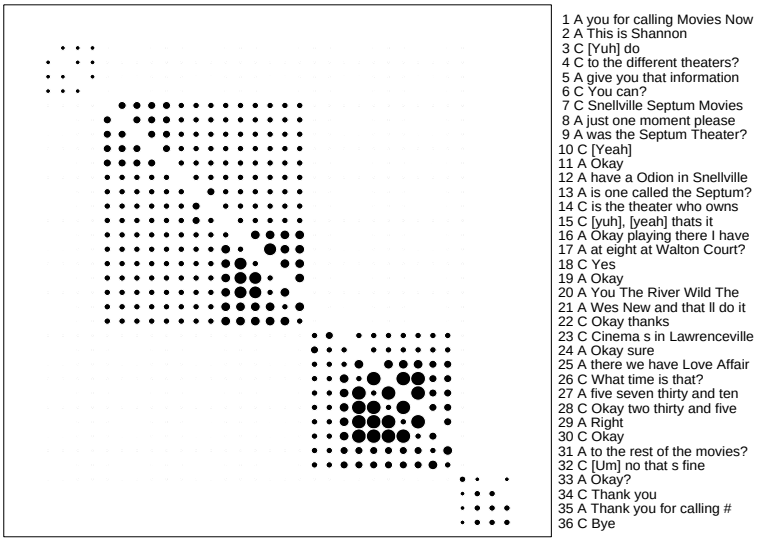



Figure 4-7: Experiment 1: Bubble plot of a movie schedule dialogue. The size of the bubble at clauses $(i, j)$ is proportional to the fraction of coders that places clause $i$ and clause $j$ in the same segment. The last few words of each clause are aligned to the right of the bubble plot.

number of coders annotating a text transcription (i.e. in our case N ranged between 7 and 9). Let $i$ and $j$ be two integers representing two clauses in the text, ranging from 1 to $T$ (the text size). Rotondo defines the correlation coefficient $R(i, j)$ between the two clauses as the fraction of coders who placed the two clauses in the same segment:

$$R(i, j) = \frac{N(i, j)}{N} \tag{4.1}$$

$R(i, j)$ ranges between 0 and 1. The coefficient is 0 if all coders agree that $i$ and $j$ are not in the same segment, and it is equal to 1 only if all coders agree that $i$ and $j$ are in the same segment (which segment it is, however, can vary from coder to coder). If all coders agree, the coefficient jumps from 0 to 1 at the boundary.

$R$ is a square matrix of size $T$. Figures 4-7 and 4-8 are two bubble plots of the $R$ matrix for the same dialogue annotated in two different experiments. Figure 4-7 has been produced by the first experiment while Figure 4-8 by the second one. The bubbles are proportional to the coefficient $R$. The last few words of each clause are aligned to the right of each bubble plot.

Perceptually, the more the plot looks block diagonal, the higher the agreement among coders. For example, the kappa coefficient for the dialogue in Figure 4-7 is equal to 0.43, while the kappa coefficient for the dialogue in Figure 4-8 is equal to 0.80. The second bubble plot is more sharply block diagonal, indicating stronger agreement among coders.

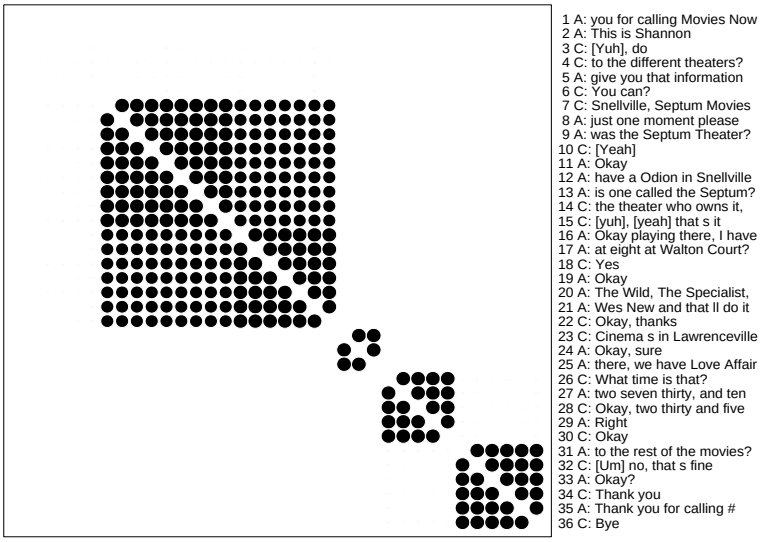



Figure 4-8: Experiment 2: Bubble plot for the movie schedule annotated also in Experiment 1. The plot is more sharply block diagonal, indicating stronger agreement among coders.

### 4.2.5 Agreements: Task Structure and Contributions to Discourse

Figure 4-9 is the segmentation proposed by the majority of coders for the dialogue displayed in Figure 4-8. It should be compared with Figure 4-3. It is interesting to note that it essentially displays the same segment boundaries, except that there are no embedded segments, and that there is no final closing segment. Figure 4-10 is the display of the majority segmentation for another dialogue. The two dialogues illustrate several features about discourse segments. The first segment typically starts with a request for information that needs to be further specified before the agent can answer. The subsequent segments can be initiated either by the agent or by the customer. Each segment contains one or more agent statement reporting some new information. The most common annotated segment was composed of two levels of contributions. At the top level, the segment was composed of a request phase followed by a response phase. At the second, more detailed level, each request and response was composed of one or more discourse contributions. The organization of each contribution followed the principles of cooperative dialogue: the content was presented by one speaker (i.e., a statement or a question) and evaluated by the other speaker with an acknowledgment or some other appropriate response [22]. Here is a typical example:

[ 4] C: is there a [ah] number that you dial to just get all the different theaters?
[ 5] A: I can give you that information.
[ 6] C: You can?

[ 7] C: Okay, in Snellville, Septum Movies.
[ 8] A: Sure, just one moment please...
[ 9] A: And that was the Septum Theater?
[ 10] C: [Yeah].
[ 11] A: Okay...
[ 12] A: Okay, I have a Cineplex Odion in Snellville.
[ 13] A: Do you know that there is one called the Septum?
[ 14] C: [Yeah], it's called Septum, but Cineplex Odion is probably the theater who owns it, but it's called the Septum Theater, but [yuh],
[ 15] C: [yeah] that's it.
[ 16] A: Okay [heavy_breathing] playing there, I have...
[ 17] A: this would be the one at Highway seventy eight at Walton Court?
[ 18] C: Yes.
[ 19] A: Okay.
[ 20] A: Playing there is Exit to Eden, Little Giant, Only You, The River Wild, The Specialist, Wes Craven's New Nightmare,
[ 21] A: and that'll do it.
[ 22] C: Okay, thanks.

[ 23] C: How about [um], the Town Center Cinema's in Lawrenceville.
[ 24] A: Okay, sure...
[ 25] A: Alright, playing there, we have Love Affair

[ 26] C: What time is that?
[ 27] A: Love Affair is twelve, two thirty, five, seven thirty, and ten.
[ 28] C: Okay, two thirty and five.
[ 29] A: Right.
[ 30] C: Okay.

[ 31] A: Okay, do you want me to read the rest of the movies?
[ 32] C: [Um] no, that's fine.
[ 33] A: Okay?
[ 34] C: Thank you.
[ 35] A: Thank you for calling.
[ 36] C: Bye.

Figure 4-9: Experiment 2: Majority segmentation of a movie schedule dialogue. Segment boundaries are placed at changes in the task structure of the dialogue.

[1] A: Thank you for calling Movies Now
[2] A: This is Demita

[3] C: I'm looking for Priscilla, Queen of the Desert
[4] A: In what area, sir?
[5] C: Anywhere
[6] A: Anywhere OK One moment
[7] C: Any part in the Atlanta area
[8] A: OK
[9] A: OK, sir, I have Priscilla in at Midtown Eight and Fitz only
[10] C: Midtown?
[11] A: Eight [uh-huh] and Fitz Plaza
[12] C: And Fitz

[13] A: [uh-huh] Did you need times for matinee or evening showings?
[14] C: Evening
[15] A: OK, for which theater, Midtown or Fitz?
[16] C: Fitz please
[17] A: OK, that one is showing at four fifty-five and ten fifteen only
[18] C: Ten fifteen
[19] A: Yes, sir
[20] C: OK, all right
[21] A: OK?

[22] C: What about Pulp Fiction?
[23] A: Same location?
[24] C: Yes
[25] A: OK OK, I don't have Pulp Fiction at Fitz Let's see Or Lennox
[26] A: The only Buckhead theater showing Pulp Fiction is the Cinema or Cheshire Bridge
[27] C: [uh] Let me get it somewhere in [um] Stone Mountain, Decatur
[28] A: Stone Mountain Festival?
[29] C: And Dekalb
[30] A: OK, Stone Mountain Festival i- [uh] P- Pulp Fiction has one showing and it's at eight o'clock
[31] C: Eight o'clock
[32] A: [uh-huh]
[33] C: OK
[34] A: OK?
[35] C: Yeah Thank you
[36] A: Thanks for calling Movies Now

Figure 4-10: Experiment 2: Majority segmentation of another movie schedule dialogue.

| | | | |
|---|---|---|---|
| 13 | A: [uh-uh] Did you need times for matinee or evening showings? | Present. | Request |
| 14 | C: Evenings. | Accept. | |
| 15 | A: OK, for which theater, Midtown or Fitz? | Present. | |
| 16 | C: Fitz, please. | Accept. | |
| 17 | A: Ok, that one is showing at four fifty-five and ten fifteen only. | Present. | Response |
| 18 | C: Ten fifteen. | Accept. | |
| 19 | A: Yes, sir. | Accept. | |
| 20 | C: Ok, all right. | Accept. | |
| 21 | A: Ok? | Accept | |

The above discourse segment is composed of a request section from line 13 to line 16, and a response section from line 17 to line 21. The request section is composed of two question-answer contributions (lines 13-14 and lines 15-16). The response section is composed of one contribution with a presentation phase (line 17) and an acceptance phase (lines 18 to 21). Note the length of the acceptance phase in the response section. It takes 4 dialogue turns for both speakers to ground the conversation and convince one another that the information has been correctly transmitted and understood.

### 4.2.6 Disagreements: Repairs and Multiple Purposes

Disagreement in placing segment boundaries mostly occurred around events such as incomplete sentences, restarts, repeated questions, and speech repairs. Disagreements in segment purposes mostly occurred when a surface linear segmentation represented an underlying hierarchical segmentation. Other disagreements in assuming a linear segmentation occurred when the conversation was switching back and forth between two purposes before completing either of them. This made the linear annotation task somewhat more difficult, with coders deleting or inserting segments. In general, coders tended to disagree in placing segment boundaries when the spontaneous dialogue structure would violate the linear organization of discourse contributions.

#### Repairs

Repairs can be self-repairs, involving only one speaker and one dialogue turn, or they may involve both participants for a few dialogue turns. They are also local discourse phenomena: as soon as the speaker or hearer detects an error in the speech stream, she tries to repair it [90, 56, 47, 89, 73, 20]. The instructions specified that in case of speech repairs, segment boundaries should be placed before complete clauses that could be fully understood as switches in segment purposes. However, some spontaneous repairs could not be handled consistently by this rule.

The five examples listed in Figure 4-11 are representative sequences of where coders disagreed on the exact location of a segment boundary. The first one is a fresh start which develops over a

**1.** C: Let me–
C: What about–
C: What time is True–
C: [uh] I'm sorry
C: What time is The Client playing?

**2.** A: Any particular movie?
C: What?
A: Any particular movie?
C: [uh] Stargate

**3.** C: Okay, [ah] what's playing around nine forty?
A: [humming]
C: Well what's playing period? I mean
A: Hey, that'd be a better question
C: [Yeah] [Laughter]
A: Frankenstein is playing at ten

**4.** A: On Highway Eighty Five
C: Right, okay
A: [um] currently playing there are, let's see
A: Would you like today's show times?
C: [uh] ye- well, just give me a listing of
what's playing first, and I'll tell you what I want

**5.** A: I have it at the Cobb Place Eight
C: Is that it?
A: They're at Parkway near Highway Forty One
C: Is that the only one?
C: Is it at Galleria?
A: Yes, sir, it's next show time is at [uh] four thirty

Figure 4-11: Five examples of speech and dialogue repairs and fresh starts. Coders tended to disagree about where to place segment boundaries around these locations.

sequence of clauses internal to one dialogue turn. In a fresh start, the first two partial questions and their intention are abandoned to be replaced by the last one. Researchers have pointed out that one effective repair strategy is using the same syntax (i.e. *what time*) to signal to the listener exactly what she is replacing [56, 20]. Some coders placed a segment boundary at the first or second question, while others placed it at the complete question *What time is The Client playing?*. The second and the third examples are fresh starts which involve both speakers, and can also be interpreted as a replacement of the first question by the second question. In the second example, the words of the agent's question *Any particular movie?* were not recognized by the listener, while in the third example the question *what's playing around nine forty?* is too constraining and cannot be answered by the agent unless the time constraint is relaxed.

The fourth and fifth examples pose more complex segmentation problems. The fourth example illustrates how the agent and the customer negotiate how the movie schedule information should be reported. Some coders started a new segment before the agent's incomplete statement *currently playing there are...* while other coders started a new segment at the question *Would you like today's show times?*, and other coders started a new segment at the customer response *just give a listing of what's playing first*. The agent's question starts a new task (listing show times) which is a subtask of the incomplete purpose (listing movie titles) and not a replacement or a digression. The agent self repair signals to the listener that, in addition to listing movie titles, she can also report all of the show times. In this case, the fresh start does not replace the incomplete purpose, instead it contributes to it, and serves as a proposal offered by the agent that the customer can either accept or reject. The customer's response serves two functions. Firstly, it answers the agent's question. Secondly, it also provides a specific request about how the next dialogue moves should be organized sequentially: titles first without show times (resuming the incomplete purpose of listing movie titles), then optionally show times (but only if explicitly requested). In the fifth example, coders proposed placing segment boundaries at any of the three customer dialogue turns. The segmentation is complicated by two events. Firstly the agent seems to ignore or to reject the customer's intention (*Is that it?*), effectively replacing it with another one (specifying the theater location). Secondly, the customer repairs her unanswered request (*Is that the only one?*) by replacing it with a more specific one (*Is it at Galleria?*).

### Multiple Purposes

The linear segmentation constraints did not allow coders to annotate multiple active purpose labels. As a consequence, coders tended to disagree when the underlying task structure involved two concurrent purposes.

Figure 4-12 displays two sections in which there are two active purposes at the same time. The first example includes two tasks, listing the show times for Angels in the Outfield and for True Lies.

| *Examples of multiple active purposes* | |
|---|---|
| **1.** 6 A: You wanted Angels In The Outfield<br>7 C: And True Lies, I need times<br>8 A: [uh] Seven forty and nine fifty<br>9 C: Yeah<br>10 A: And True Lies would be<br>11 A: seven o'clock and nine forty-five<br>12 C: Thank you | **2.** 3 C: [um] I'm looking for the number to a National Seven<br>4 A: [mm-hmm]<br>5 C: Do you know what movies are playing?<br>6 A: Yeah, I could tell you what's playing there.<br>7 C: OK.<br>8 A: OK, [uh] the number first is seven two si- I'm sorry.<br>9 A: Seven six two, nine six three six. |

Figure 4-12: Two examples of multiple active purposes. When two purposes are active at the same time, it is difficult to annotate segments without embedding them into each other.

Some coders annotated two different segments at lines 7 and lines 8 and 9, while others annotated one segment from line 6 to line 12. At line 8, the agent postpones responding immediately to the customer request *And True Lies, I need times*, and proceeds to report first the times for "Angels in the Outfield" and then for "True Lies". The second example illustrates the case when two consecutive dialogue turns start two different segment purposes, and the second one is put on hold until the first one is completed. In the example, some coders annotated lines 3-4 as a separate segment, while others did not because the response did not follow the telephone number request until much later in the dialogue.

### 4.2.7 Discussion

In this experiment, we were able to reach a very reliable level of agreement by constraining the task along two dimensions. Firstly, the task was limited to linear segmentation. Secondly, the level of detail was determined by the segment labels, which were defined a priori to be task specific. The constrained annotation task greatly reduced the cognitive load for the coders.

The patterns of agreement among coders indicate that it is possible to reliably annotate discourse segments that are defined by their intentional purpose [38]. In addition, the internal structure of each annotated segment is consistent with the theory of cooperative discourse contributions [22].

The patterns of disagreement were located around dialogue phenomena that could not be covered adequately by a simple linear segment model, such as speech and dialogue repairs and multiple concurrent purposes. Nevertheless, such events constituted the exception rather than the rule in the dialogues.

The next two experiments were designed to better understand which constraints were necessary to obtain reliable annotations, and which could be relaxed in favor of more expressive segmentations without substantially penalizing reliability. Experiment 3 relaxed the linear segmentation constraint by allowing nested segments. Experiment 4 relaxed the constraint on segment purpose labels by not

limiting the choice of labels to a predetermined set.

## 4.3 Experiment 3: Directed Embedded Segmentation

### 4.3.1 Data and Task

We selected a representative sample of 12 dialogues from the movie schedule domain used in Experiment 2. The structure of the online instructions was very similar to that used in Experiment 2, with a set of examples for segment and subsegment initiatives, several counter-examples, and a set of three annotation exercises that preceded the test annotations. This task was similar in nature to Experiment 2, with the same set of five segment purpose labels. The only difference was that coders were allowed to annotate embedded segments, within the syntactic constraints imposed by *Nb* . The segment-subsegment structure was intended to mirror the task-subtask structure of each dialogue.

The annotation tool enforced the following syntactic constraints. First, a clause could either open a top-level segment or a sub-segment, but not both. Second, only one level of embedding was allowed (i.e., any section of the dialogue could at most belong to a segment and a subsegment). Third, no segment or subsegment could cross boundaries. The constraints imposed by *Nb* limited the cognitive load for each clause to the following choice: should the clause (1) open a top-level segment? (2) open a subsegment? (3) continue the currently open segment structure? (4) close a segment?

### 4.3.2 Coders

Five paid volunteers participated in this experiment, each one annotating the set of 12 dialogues. Two coders had already participated in Experiment 2 and the three other coders were graduate students members of the Spoken Language Systems group. The second experiment took place one year after the first one. As a consequence, it is very unlikely that some coders' participation in Experiment 2 biased the results. On average, each annotation session lasted one and a half hours. The coders divided their time equally between reading the instructions, completing the exercises, and annotating the set of text transcriptions. All coders found the task challenging but praised the user interface, the instructions, and the exercises.

### 4.3.3 Agreement Statistics

#### Segment Boundary Placement

The complete data sample included 513 clauses, annotated by 5 coders. On average, each dialogue included 48 clauses. To compare segment boundary placement, we clustered segment and subsegment initiatives into the class *boundary* and all other clauses into the *non-boundary* class. On average,

coders hypothesized a new segment or subsegment boundary every 7 clauses. The average pairwise precision was 73.4% and the avrerage pairwise recall was 74.9%. The group-wise kappa coefficient was 0.70. The value of the kappa coefficient indicates that the inter-coder agreement is on the threshold of reliability. The agreement statistics were about 10 units lower than the agreement statistics for the second experiment (for the second experiment, we found precision of 85%, recall of 83.9% and kappa of 0.82.)

**Comparison with Experiment 2**

Figure 4-6 compares the pairwise agreement statistics for Experiment 2 and Experiment 3. In the latter experiment, three out of ten pair of coders agreed with a kappa coefficient below the reliability threshold of 0.70, and the highest value of kappa was 0.78. To further compare the third experiment with the second one, we evaluated the agreement between the boundaries proposed by the majority of coders in the third experiment with the boundaries proposed by the majority of coders in the second experiment. There was a significant overlap between the boundaries proposed by the second and third experiment majorities. The third experiment coders produced a more detailed segmentation with 47% more hypothesized boundaries, corresponding to subtask and digression segments. We found that 88% of the majority boundaries proposed in the second experiment were also proposed in the third majority, while 68% of the boundaries proposed in the third experiment were also proposed by the second experiment majority. The kappa coefficient between the two majority segmentations was 0.73, confirming the statistical significance of the overlap between the two experiments.

**Segments and Subsegments**

The instructions for Experiment 3 allowed coders to annotate embedded subsegments. To evaluate whether coders agreed in placing segment and subsegment boundaries, we clustered each clause into one of three categories: *segment* for segment initiatives, *subsegment* for subsegment initiatives, and *other* for all other clauses. By constraining the task to a three-way classification problem, we were able to assess reliability with the kappa coefficient. The table displayed in Figure 4-13 reports precision, recall and kappa coefficient. Coders disagreed the most in annotating subsegments, either deleting them, inserting them, or substituting them with top-level segment labels. The kappa coefficient was 0.62. While it was lower than the coefficient found for linear segmentation (0.71-0.82), it was also substantially higher than the kappa coefficient for unconstrained subjective segmentations (0.45).

**Segment Purpose Labels**

For each pair of coders and each annotated dialogue, we extracted the sequence of segment purpose labels. We then ran the NIST dynamic programming symbol alignment procedure on each pair of

*Experiment 3: Embedded Segmentation*

| *Class* | *Prec.* | *Rec.* |
|---|---|---|
| Subsegments | 52.9 | 50.7 |
| Segments | 62.6 | 69.8 |
| Kappa | 0.62 | |

Figure 4-13: Experiment 3: agreement statistics for segment and subsegment boundaries.

segment purpose label sequences. On average, the symbol accuracy was 67.3% (average number of substitutions: 6.3%, insertions: 12.5% and deletions: 6.95%). The disagreements in assigning segment purpose labels is correlated with the insertions and deletions of subsegment labels.

### 4.3.4 Agreements: Task Structure

The segmentations agreed upon by the majority of coders (3 out of 5) mirrored the task-subtask structure of each dialogue. Figure 4-14 lists the majority rule segmentation for a representative dialogue. The task structure that corresponds to this dialogue is the following:

1. Select theaters playing: Priscilla Queen of the Desert *(lines 3-21)*.
   (a) Specify theater location: any parts of Atlanta *(4-8)*.
   (b) List show times at Fitz *(13-21)*.
2. Select theaters playing: Pulp Fiction *(22-36)*.
   (a) Specify theater location: Fitz *(23-24)*.
   (b) Specify other theater location: Stone Mountain or Dekalb *(27-29)*.
   (c) List show times at Stone Mountain Festival *(30-36)*.

This segmentation can be compared with the majority linear segmentation in the second experiment for the same dialogue, displayed in Figure 4-10. By comparing the two displays, it can be seen that all segment boundaries in Experiment 2 correspond to either segment or subsegment boundaries in Experiment 3.

The two top-level segments have the same intentional structure: finding theaters playing a movie, and listing the show times. It is interesting to notice the differences in the two segments. In the first segment, the agent takes the initiative of prompting for show times (lines 13 to 21). In the second segment, the dialogue history provides a blueprint for the segment purpose structure, and the intentions of the speakers can be inferred rather than explicitly stated. The customer takes the initiative of selecting the location, without specifically asking for show times. The agent infers from the dialogue history that the customer wants to know the show times.

1 A: Thank you for calling Movies Now.
2 A: This is Demita.

3 C: I'm looking for Priscilla, Queen of the Desert.

4 A: In what area, sir?
5 C: Anywhere.
6 A: Anywhere. OK. One moment.
7 C: Any part in the Atlanta area.
8 A: OK. .....

9 A: OK, sir, I have Priscilla in at Midtown Eight and Fitz only.
10 C: Midtown?
11 A: Eight [uh-huh] and Fitz Plaza.
12 C: And Fitz.

13 A: [uh-huh] Did you need times for matinee or evening showings?
14 C: Evening.
15 A: OK, for which theater, Midtown or Fitz?
16 C: Fitz please.
17 A: OK, that one is showing at four fifty-five and ten fifteen only.
18 C: Ten fifteen.
19 A: Yes, sir.
20 C: OK, all right.
21 A: OK?

22 C: What about Pulp Fiction?

23 A: Same location?
24 C: Yes.

25 A: OK. OK, I don't have Pulp Fiction at Fitz. Let's see. Or Lennox.
26 A: The only Buckhead theater showing Pulp Fiction is the Cinema or Cheshire Bridge.

27 C: [uh] Let me get it somewhere in [um] Stone Mountain, Decatur.
28 A: Stone Mountain Festival?
29 C: And Dekalb.

30 A: OK, Stone Mountain Festival i- [uh] P- Pulp Fiction has one showing and it's at eight o'clock.
31 C: Eight o'clock.
32 A: [uh-huh]
33 C: OK.
34 A: OK?
35 C: Yeah. Thank you.
36 A: Thanks for calling Movies Now.

Figure 4-14: Experiment 3: Majority segmentation for a movie schedule dialogue.

1 A: Thank you for calling Movies Now.
2 A: This is Demita.

3 C: I'm looking for Priscilla, Queen of the Desert.

4 A: In what area, sir?
5 C: Anywhere.
6 A: Anywhere. OK. One moment.
7 C: Any part in the Atlanta area.
8 A: OK. .....

9 A: OK, sir, I have Priscilla in at Midtown Eight and Fitz only.
10 C: Midtown?
11 A: Eight [uh-huh] and Fitz Plaza.
12 C: And Fitz.

13 A: [uh-huh] Did you need times for matinee or evening showings?
14 C: Evening.

15 A: OK, for which theater, Midtown or Fitz?
16 C: Fitz please.

17 A: OK, that one is showing at four fifty-five and ten fifteen only.
18 C: Ten fifteen.
19 A: Yes, sir.
20 C: OK, all right.
21 A: OK?

22 C: What about Pulp Fiction?

23 A: Same location?
24 C: Yes.

25 A: OK. OK, I don't have Pulp Fiction at Fitz. Let's see. Or Lennox.
26 A: The only Buckhead theater showing Pulp Fiction is the Cinema or Cheshire Bridge.

27 C: [uh] Let me get it somewhere in [um] Stone Mountain, Decatur.
28 A: Stone Mountain Festival?
29 C: And Dekalb.
30 A: OK, Stone Mountain Festival i- [uh] P- Pulp Fiction has one showing and it's at eight o'clock.
31 C: Eight o'clock.
32 A: [uh-huh]
33 C: OK.
34 A: OK?
35 C: Yeah. Thank you.
36 A: Thanks for calling Movies Now.

Figure 4-15: Experiment 3: Segmentation proposed by one of the five coders. Six out of seven segment boundaries are the same as in majority segmentation. The hierarchical structure of the segmentation is different from the majority.

### 4.3.5 Disagreements: Segment-Subsegment Structure

Coders tended to disagree about the hierarchical structure of the segmentations, especially insertions or deletions of subsegments. For example, compare the majority segmentation displayed in Figure 4-14 with the segmentation proposed by one coder displayed in Figure 4-15. The two segmentations agree on six out of seven segment boundaries, resulting in 85% precision and recall. The major difference between the two segmentations is in the hierarchical structure, especially between lines 13 and 21. Both segmentations appear to be plausible, with individual differences in the level of detail. In general, more than one hierarchical segmentation may be hypothesized for a given section of a dialogue, unless a precise hierarchical model is specified for the task-subtask structure of a dialogue. For example, consider the following linear segmentation:

| | **Segment 1** |
|---|---|
| 1 | A: And Frankenstein is playing at ten. |
| | **Segment 2** |
| 2 | C: No Pulp Fiction? |
| 3 | A: No. |
| | **Segment 3** |
| 4 | C: What about Marietta? |
| 5 | C: Check the Marietta Mall. |

The problem is to assign the roles of segments 2 and 3. One possible segmentation embeds segment 2 in segment 1, because it is about listing show times at the same theater of segment 1:

| **Segment 1** | |
|---|---|
| 1 A: And Frankenstein is playing at ten. | |
| | **SubSegment 2 contained in Segment 1** |
| | 2 C: No Pulp Fiction? |
| | 3 A: No. |
| **Segment 3** | |
| 4 C: What about Marietta? | |
| 5 C: Check the Marietta Mall. | |

Another segmentation embeds segment 3 in segment 2, because it is about listing the show times for the movie "Pulp Fiction":

| **Segment 1** |
| --- |
| A: And Frankenstein is playing at ten. |
| **Segment 2** |
| 2 C: No Pulp Fiction? |
| 3 A: No. |
| **SubSegment 3 contained in Segment 2** |
| 4 C: What about Marietta? |
| 5 C: Check the Marietta Mall. |

The instructions did not specify which interpretation would be more appropriate. For this particular case, since both embedded segmentations are plausible, perhaps the most appropriate segmentation would be the linear one.

### 4.3.6 Discussion

The results of Experiment 2 and Experiment 3 indicate that coders tend to agree in placing segment boundaries at changes in the task structure, while they may disagree about the hierarchy of the segmentation. Often, more than one hierarchy is plausible, and the instructions did not give precise guidelines about resolving ambiguity in the hierarchical segmentation. The hierarchical ambiguity may be due to the fact that the five purpose labels that we have chosen do not have a specific hierarchical relationship among them. It is possible that a different set of labels for subsegments and more extensive instructions might yield different results. For example, Grosz and Sidner [38, 39] and Clark [21] among others, point out that subsegments tend to play a specific role in the task structure. Typically, a subsegment might represent either a *sub-task* or a *digression*. A sub-task contributes a clarification, an explanation or a confirmation necessary for the completion of the top-level segment purpose. A digression is a section that is not directly related to the task of the top-level segment that contains it. A coding scheme that would use subtask and digression for subsegment labels might produce more reliable results.

## 4.4 Experiment 4: Linear Segmentation With No Labels

### 4.4.1 Data and Task

We selected the same sample of 12 movie schedule dialogues used in experiment 3. The structure of the online instructions was very similar to the structure of the instructions for Experiment 2, with a set of examples for segment initiatives, examples of sentences that should not open a segment, and a set of three annotation exercises that preceded the annotations. The exercises stressed the fact that the segmentation should follow the task structure of the dialogue: segments should be open

when a new task or subtask would start. The only difference from Experiment 2 was that coders were instructed not to give purpose labels to the segments, instead of choosing from a set of labels.

### 4.4.2 Coders

For this experiment, five new paid volunteers participated. They were all graduate students at MIT. None of them participated in any of the previous text annotation experiments. To our knowledge, they did not have experience with discourse analysis or conversation analysis. Once again, each coder completed the annotation session in one round of approximately two hours, with the actual annotation lasting between 30 and 40 minutes. The coders completed the annotations without feedback from others. The only contact with the instructor was by electronic mail after the annotation experiment was concluded.

### 4.4.3 Agreement Statistics

On average, four coders annotated a segment boundary every 7.5 to 9 clauses, while one coder was more prolific and annotated one segment boundary every 5.5 clauses. The average pairwise precision was 78%, the average pairwise recall was 75% and the group-wise kappa coefficient was 0.715. The statistics are only slightly better than for Experiment 3, and about 10 points lower than for Experiment 2.

#### Comparison with Experiment 2

We compared the majority segmentations of Experiment 2 with the majority segmentations produced by Experiment 4 on the same dialogue. We found that 90% of the boundaries proposed by Experiment 2 were also proposed by a majority of coders in Experiment 4. This is a very encouraging result, indicating that the segment purpose labels chosen for Experiment 2 were appropriate for describing the vast majority of purposes in the dialogues. Coders in Experiment 4 were more prolific in assigning segment boundaries than for Experiment 2. We found that the boundaries of Experiment 2 represented 75% of the boundaries proposed by the majority of coders in Experiment 4. The kappa coefficient across the two experiments was found to be 0.79, confirming the strong correlation between the two experiments.

#### Comparison with Experiment 3

The boundaries proposed by the majority of coders in Experiment 4 were strongly correlated with the boundaries proposed in Experiment 3. We found that 81.6% of the boundaries proposed by the majority of coders in Experiment 3 were also proposed by the majority of coders in Experiment 4, and that 75.4% of the boundaries proposed by the majority of coders in Experiment 4 were also proposed

by the majority of coders in Experiment 3. The kappa coefficient across the two experiments was 0.785. In Experiment 4, segment boundaries corresponded to either top-level segments or embedded subsegments annotated in Experiment 3.

### 4.4.4 Agreements: Task Structure

Coders tended to annotate the same type of segment units as in Experiment 2. For example, the dialogue displayed in Figure 4-10 produced the same majority rule segmentation as in either Experiment 2 or Experiment 3. Segment units tended to be structured into *request-response* phases, with each phase elaborated over one or more *presentation-acceptance* contribution.

### 4.4.5 Disagreements: Different Levels of Detail

The major source of disagreement among coders was that some coders applied a more detailed analysis than others, annotating individual *presentation-acceptance* discourse contributions as separate segments. These contributions included responses, clarifications, confirmations, subtasks, digressions and repair sub-dialogues. For example, the following section displays four possible boundaries between discourse contributions that were annotated by some, but not all coders.

| |
|---|
| 6 C: Okay, in Snellville, Septum movies<br>7 A: Sure, just one moment please |
| 8 A: And that was the Septum Theater?<br>9 C: [Yeah]<br>10 A: Okay<br>11 A: Okay, I have a Cineplex Odion in Snellville |
| 12 A: Do you know that there is one called the Septum?<br>13 C: [Yeah], it's called Septum, but Cineplex Odion is probably the theater who owns it, but it's called the Septum Theater, but [yuh], [yeah] that's it<br>14 A: Okay [heavy breathing] playing there, I have |
| 15 A: this would be the one at Highway seventy eight at Walton Court?<br>16 C: Yes<br>17 A: Okay |
| 18 A: Playing there is Exit to Eden, Little Giant, Only You, The River Wild, The Specialist, Wes Craven's New Nightmare, and that'll do it<br>19 C: Okay, thanks |

## 4.5 Discussion

While the first experiment produced less reliable results than the other three experiments, the analysis of the segmentations proposed by a majority of coders allowed us to evaluate the empirical

basis of at least two theories of discourse segment structure. At the top level, segment boundaries were consistently placed at switches in task-related purposes, such as listing the movies playing at a particular theater (e.g., Figure 4-3). At a more detailed level, segments tended to correspond to one or more discourse contribution, containing a presentation phase followed by an acceptance phase.

In all of the experiments, coders tended to disagree in placing segment boundaries around speech and dialogue repairs and preliminary sequences. Repairs and preliminaries momentarily violate the strict *presentation-acceptance* and *request-response* organization of a dialogue. Coders tended to disagree also about how to segment multiple purposes that were pursued at the same time, and about how to assign a hierarchy to the segmentations. Often more than one hierarchical view of a dialogue might be plausible, given that the segment purpose labels that we have chosen did not have precise hierarchical relations among them. The results provided very useful data for future annotation experiments. We believe it is possible to obtain more reliable results in annotating hierarchical segmentations, provided the instructions give extensive examples of *subtasks* and *digressions* and provide precise guidelines about the differences between *top-level* segments and *subsegments*.

The last three experiments presented in this chapter are specific to the movie schedule application domain. To what extent can they be generalized to other corpora and genres? For example, in Chapter 2 we listed four other corpora of task-oriented dialogues, and we discussed how they differ in speaker's roles and task structure. Our belief is that the participant *roles* in the conversation are very important in determining the structure and size of discourse segments in a dialogue. As a consequence, we cannot draw any conclusions about the task structure of dialogues with other roles, such as *expert - novice* and *teacher - student*. On the other hand, we believe that our results can be extended to many other types of information-seeking dialogues, in which the agent role is one of reporting some specific information that is relevant for the customer, such as restaurant addresses and classified ads.

Finally, discourse segment structure is correlated with specific acoustic and prosodic cues such as changes in pause duration, speech signal amplitude and pitch contour [82, 36, 46, 52]. For example, Hirshberg and Nakatani report an increase of the kappa coefficient from 0.67 to 0.80 when trained coders were able to listen to the speech signal as well as read the transcription [45]. All of the annotations reported in this thesis are based on text alone. We leave the issue of segmenting dialogues from a combination of text and speech to future work.

The analysis of the annotated data demonstrates that although reporting movie schedules is a relatively simple information-seeking domain, it has a very rich conversational structure, including preliminary sequences, repairs and multiple active purposes. The next chapter analyses in detail the discourse structure of the movie schedule domain. The data analysis is based on the annotation of 190 dialogues using the linear segment structure and purpose labels used in Experiment 2.

# Chapter 5

# Case Study: Information-Seeking Dialogues

In this chapter we present a case study of conversation analysis that is based on a corpus annotated with *Nb* . This case study is an example of how *Nb* has been used to analyze the discourse structure of natural dialogues. The annotated data provide quantitative evidence about turn-taking transitions within segments and segment transitions within a dialogue. The analysis focuses on how human-to-human conversation differs from IVRs and question-answer systems.

The first question we would like to address is whether or not a structural model is appropriate for predicting the internal organization of discourse segments. Examples of structural models are finite state machines, transition networks and context-free grammars. They are rooted in syntactic and semantic analysis of sentences. In analogy with syntactic models for individual sentences, structural models have been proposed to model the observed sequence of communicative acts in natural spoken dialogue [97, 65, 108]. Using a grammar-based approach has been motivated by the observation that sequences of communicative acts tend to appear in adjacency pairs such as *Statement-Acknowledgment* and *Question-Answer* [91]. Finite state machines and context-free grammars have been used to model graphical user interfaces [76], typed natural language interfaces [108, 29] and spoken language interfaces [4, 8, 18, 70]. They have been used to model natural language interactions in Wizard-of-Oz studies, in which users talked to a system simulated by an engineer, before the system is fully developed [98, 29], and sequences of communicative acts in spontaneous telephone conversations of the Switchboard corpus [51]. They have also been used to provide a predictive dialogue model for human-to-human spoken interaction within a speech translation system [50]. One important feature of finite state machines and context free grammars is that state transitions can be weighted by probabilities. It is therefore possible to estimate numerically the most likely next dialogue state given the current state, using relatively simple computational techniques such

as dynamic programming. Critics of the structural model approach have argued that such models may be appropriate only for IVR and question-answer systems in which the sequence of the interactions is essentially defined a priori (e.g., [24, 87]). They argue that the number of states and state transitions would become too large if one wished to model co-operative dialogues with clarifications, confirmations, switches in intentions initiated by either speaker, and purposes that can be completed out of order. In addition, grammars may be ambiguous, and a sentence may be interpreted by more than one communicative act, depending on the intention of the speaker and the discourse context. As a result, there could be multiple plausible state transitions after each dialogue turn. In this chapter, we will examine a simple structural model and analyse whether or not it is appropriate for predicting the internal organization of a discourse segment, and whether or not natural dialogues lead to a state space explosion. We distinguish two related problems. The first one is to determine if a simple structural model is adequate to predict the customer communicative acts. The second one is to determine whether the model is adequate to predict the agent's actions. The annotated data provide different answers to these two problems.

The second question that we would like to address is to characterize how the agent reports information to the customer. When speech is the only medium available, it is impractical to report large amount of information in a single dialogue turn. Organizing and reporting the information in a natural sounding way is crucial to designing usable spoken language systems. The annotated data provide some quantitative evidence about how the information is reported and confirmed in each discourse segment.

The third question we would like to address in this chapter is to assess the extent to which a hierarchical data structure (e.g., a stack, or a tree) can process segment transitions during a dialogue. The choice of a hierarchical data structure is motivated by the assumption that segment purposes are related to each other by hierarchical relations such as *task-subtask* and *task-digression*. Critics to the stack model have argued that a stack is not able to model some spontaneous spoken dialogue phenomena. One alternative to the stack based approach is linear recency. In a linear recency model, the meaning and intention of a sentence can be interpreted by a backward linear search in the discourse history [106].

The rest of the chapter is organized as follows. Firstly, we describe how we annotated a corpus 190 transcriptions with communicative act labels and segment labels. Secondly, we analyze turn transitions within segments using a probabilistic finite state model, and we discuss strengths and weaknesses of the model. Thirdly, we analyze segment transitions using a stack model.

| **Intentional Structure: Segment Purposes** |
|---|
| List Theater Showing Movies |
| Specify Theater Location |
| List Movies At Theater |
| List Show Times |
| List Phone Number for Theater |
| **Contributions To Discourse:** |
| Request, Response |
| **Contributions to Discourse:** |
| Presentation, Acceptance |
| **Communicative Acts:** |
| Request, Inform, Confirm, Statement, Acknowledge |

Table 5.1: Outline of the coding scheme used for annotating the movie schedule dialogues.

## 5.1 The Annotation Coding Scheme

Using *Nb* , one expert coder (the author) has annotated 190 dialogue transcriptions, using the coding scheme outlined in Table 5.1. The annotation has a four-layer structure. The top layer is the intentional structure of the dialogue, modeled by a linear sequence of segments. To study how discourse segments are organized internally, we added three more annotation layers. The second and third layers are contributions to discourse and the fourth layer is the sequence of communicative acts. Each segment is typically structured as a request followed by a response, although either may be missing. Request and response are each realized by *presentation-acceptance* pairs. By default, the first communicative act of each request or response is the *presentation* and the following acts are the *acceptance*.

Figure 5-1 lists an example annotated dialogue. The example shows how a request contribution is typically initiated by a communicative act of type **Request**, and the response is started by the agent's **Inform** statement, which accomplishes the purpose of the segment. In the second segment, the request is absent, because the agent volunteers some information that was not explicitly requested by the customer.

This coding scheme combines the theory of intentional discourse structure (i.e., top-level segment purposes) [39] with the theory of discourse contributions [22] (i.e., requests, responses, presentations and acceptances). It is also related to the Map Task coding scheme [16], in which top-level segments correspond to *Transactions*, and nested contributions corresponding to *Games*. However, we have made one important simplification. While the three references mentioned above assert the existence of arbitrarily embedded discourse segments, we explicitly fix a priori the level of detail in the segmentation to include only four layers: segments, request - response, presentation - acceptance, and communicative acts. We justify this simplification on the basis of two observations. Firstly, our experience in inter-coder agreement indicates that it is hard to obtain reliable annotations when an arbitray level of detail is allowed. Secondly, this information seeking-domain allows an unambiguous

| | *Text Transcription* | *Communicative Acts* | *Contributions to Discourse* | | *Segment Purpose* |
|---|---|---|---|---|---|
| 1 | A: Movies Now this is Demita. | A Statement | | | |
| 2 | Can I help you? | A Request | | | |
| 3 | C: Yeah I was trying to locate a movie | C Request | Present. | Request | Segment *(List Theater Playing Movie)* |
| 4 | and it is at a movie theater or what theater it is. | | | | |
| 5 | A: okay | A Ack | Accept. | | |
| 6 | A: What film were you trying to | A Request | | | |
| 7 | C: Stargate . On Cobb Parkway ? | C Inform | | | |
| 8 | It is called Galleria. | | | | |
| 9 | A: okay I do show Stargate at the Galleria Eight . | A Inform | Present. | Response | |
| 10 | C: Is that is that Galleria Specialty Mall I wonder? | C Request | Accept. | | |
| 11 | A: I believe it is. | A Statement | | | |
| 12 | It is just across the street from the Cumberland Mall. | A Inform | | | |
| 13 | C: Yes that is it | C Statement | | | |
| 14 | A: okay | A Ack | | | |
| 15 | C: okay | C Ack | | | |
| 16 | A: The show times are five o'clock seven forty five | A Inform | Present. | Response | Segment *(List Show Times for Movie)* |
| 17 | and ten thirty . | | | | |
| 18 | C: Five o'clock sounds okay | C Confirm | Accept. | | |
| 19 | I guess five would not be considered twilight or? | C Request | | | |
| 20 | A: No it would not. I am I am looking now | A Statement | | | |
| 21 | and they probably do have some because | | | | |
| 22 | I do see it listed but not on that particular feature. | A Inform | | | |
| 23 | C: I got it okay, okay | C Ack | | | |
| 24 | A: okay | A Ack | | | |
| 25 | C: thank you | C Ack | | | |
| 26 | A: okay | A Ack | | | |
| 27 | C: goodbye | C Ack | | | |

Figure 5-1: Example annotated dialogue.

definition of the concepts of segment purposes, requests, responses, presentations and acceptances, as discussed below.

### 5.1.1 Segment Purposes

We divided each dialogue into a linear sequence of segments using five domain-dependent purpose labels listed in Table 5.1. The dialogue participants' actions are motivated by wanting to accomplish common purposes, such as **List Theater Showing Movie**. Unlike communicative act labels, purpose labels do not encode the individual intentions of the speakers. Instead, we consider the purposes as being *joint projects*, or *shared plans*, between the customer and the agent [39, 21, 22, 37, 57, 58]. Each segment is a sequence of agent and customer actions that co-operate to accomplish the purpose. This type of linear segmentation is the one that produced the highest inter-coder agreement in the experiments presented in Chapter 4. While segments are annotated as a linear sequence, their *purposes* may be related by hierarchical relations, which we did not annotate explicitly for this study.

### 5.1.2 Contributions to Discourse

Each segment has been divided into two optional contributions, a request and a response. The first contribution of a segment is the request. The response begins when the agent reports the information that is appropriate for accomplishing the purpose of the annotated segment. For example, the response of the segment **List Theater Playing Movie** begins when the agent says: *Okay I do show Stargate at the Galleria Eight.* Both request and response are optional. Some segments contain only a request contribution, because the purpose is switched before the agent reports the information. Some other segments contain only the response contribution, if the agent volunteers the information without a specific request, or if another segment separates the request from the corresponding response. Each request and response has been further divided into presentation and acceptance [22]. The presentation is the first communicative act of each contribution. In the presentation contribution one speaker presents some content to be evaluated (for example, a Request or an Inform act). In the acceptance contribution, both speakers are involved in setting a common ground by accepting, clarifying, confirming or rejecting the presented content.

Dividing a segment into request and response contributions is appropriate for an information retrieval domain such as the movie schedule, in which the roles of the speakers are clearly defined. The customer is the information seeker, and the agent responds by reporting some information. In general, this *request-response* division can be easily determined for tasks that involve database queries. This division might be difficult or even inappropriate for different tasks such as scheduling meetings or giving instructions.

| *Act* | *Description and examples* |
|---|---|
| **Request** | Direct or indirect request for information.<br>*A: Any particular movie?*<br>*C: Do you happen to know the number?* |
| **Inform** | Statement reporting some new information.<br>*A: The next show is at nine twenty five.*<br>*C: It is located in Midtown or Buckhead.* |
| **Confirm** | Explicit confirmation of some given information.<br>*C: nine twenty five?*<br>*A: okay-Midtown or Buckhead.* |
| **Statement** | Yes/No answer, explanation, or some other statement that is not an Inform statement.<br>*A: I don't know the name of it though.*<br>*C: That is the one I wanted but – yeah.* |
| **Acknowledge** | Brief response such as *hmm-hmm* and *okay* or a polite form.<br>*A: Okay, hold on a second.*<br>*C: Alright. Thanks.* |

Table 5.2: List of communicative acts used in the annotations.

### 5.1.3 Communicative Acts

Each clause in each dialogue turn has been classified into one of five communicative act types that we found representative for information retrieval dialogues. The list of communicative acts is displayed in Table 5.2. The list is a subset of acts which are frequently used in discourse analysis [92, 93, 1]. On average, a dialogue turn is composed of one to three clauses. Each clause has been annotated with a separate communicative act. In our annotated data, 74.5% of the dialogue turns are a single clause. The most common sequence of acts included in the same dialogue turn were (from the most frequent to the less frequent):

- **Acknowledge + Inform.** A: Okay. I do show Stargate at the Galleria Eight.
- **Acknowledge + Statement.** C: Okay. That's the one I wanted to see.
- **Inform + Statement.** A: That one is playing at half past three and five forty five. And that's all.
- **Confirm + Statement.** A: At the Galleria Eight? I am not sure.

## 5.2 Structure of the Dialogues

On average it takes 28.5 turns to complete a movie schedule dialogue. About half of the dialogues last between 8 and 20 dialogue turns and are composed of one or two discourse segments, while the other half involve 21 dialogue turns or more. The most common pair of segments is **List Theater Playing Movie** followed by **List Show Times At Theater**, as illustrated by the example in Figure 5-1.

In principle, if a request is fully specified and immediately answered, a segment should require

| *Direct Request-Response Pairs* | |
|---|---|
| Specify Theater Location | 0.47 |
| List Show Times | 0.45 |
| List Phone For Theater | 0.27 |
| List Movie Showing At Theater | 0.24 |
| List Theater Showing Movie | 0.20 |

Table 5.3: Fraction of customer's requests that are directly followed by an agent's Inform response. All other requests are separated from the response by at least two dialogue turns that clarify or confirm the request.

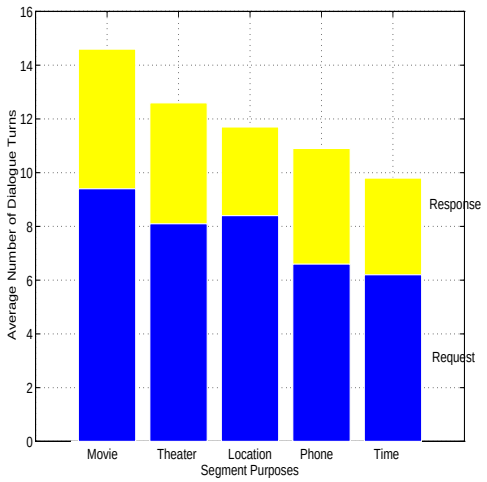



Figure 5-2: Average duration (number of turns) of the movie schedule segments, detailed by discourse segment purposes.

two dialogue turns. Table 5.3 shows that *Request* - *Response* pairs are rarely realized by a simple *Question-Answer* adjacency pair. Instead, segments tend to be organized into one or more discourse contributions. About half of the segments require 5 dialogue turns or fewer to complete, and half require 6 turns or more.

Figure 5-2 lists the average segment durations broken down into purpose and request and response contributions. The figure indicates that, on average, about $\frac{2}{3}$ of the segment duration is spent specifying the request, and $\frac{1}{3}$ reporting the information. The request contribution of a segment takes an average of 6 to 10 dialogue turns, and the response contribution takes 3 to 6 turns. Each contribution contains a presentation and an acceptance. For example, in Figure 5-1 the first segment is **List Theater Playing Movie**, from line 3 to line 15. The request contribution is from line 3 to line 8, and the response contribution is from line 9 to line 15. The second segment, **List Show Times For Movie** starts directly with the response contribution, in which the agent contributes some information that was not explicitly asked for with a request.

Table 5.4 lists the frequency of occurrence and the average word count for the annotated commu-

| *Agent* | | | | *Customer* | | | |
|---|---|---|---|---|---|---|---|
| *Act* | *Freq.* | *Words* | *% Elliptical* | *Act* | *Freq.* | *Words* | *% Elliptical* |
| Acknowledge | 30.8 | 3.1 | 87 | Acknowledge | 47.9 | 2.3 | 91 |
| Inform | 27.8 | 12.7 | 31 | Request | 29.5 | 9.0 | 11 |
| Statement | 15.0 | 6.7 | 17 | Confirm | 13.1 | 5.3 | 65 |
| Request | 15.0 | 12.3 | 15 | Inform | 5.9 | 7.9 | 42 |
| Confirm | 11.3 | 6.4 | 62 | Statement | 3.4 | 6.9 | 3 |

Table 5.4: Frequency of occurrence, average word count, and fraction of elliptical realizations of each annotated communicative act.

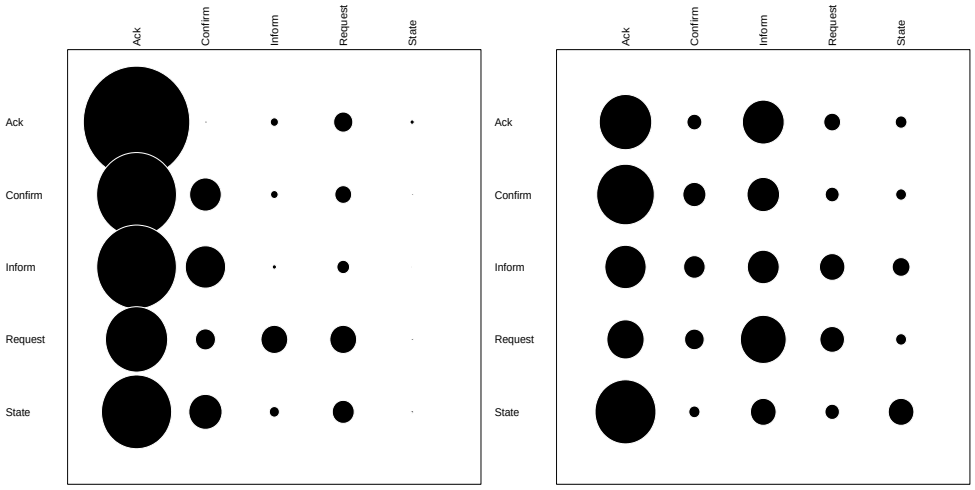



Figure 5-3: Most frequent communicative act transitions. Left: After the agent's speech. Right: After the customer speech. Rows: preceding communicative act. Columns: following communicative act. The size of the bubbles are proportional to the frequency of occurrence of each transition.

nicative acts. Acknowledgments and confirmations account for more than one half of the customer's speech and more than one third of the agent's speech. Acknowledgments tend to be short phrases of 2 or 3 discourse cue words such as *okay* and *alright*. The frequency distribution of the communicative acts is typical of a database query domain, in which the customer's most frequent acts are to either request some information, or acknowledge or confirm the information reported by the agent. Along with frequent acknowledgments and confirmations, one of the features of natural dialogues is that communicative acts are often realized as elliptical clauses - short noun phrases and prepositional phrases rather than full sentences. The column labeled *% Elliptical* shows the fraction of communicative acts which were realized by elliptical clauses. The data indicate that the customer tends to use more elliptical clauses than the agent, and that about $\frac{2}{3}$ of confirmations are elliptical. Interestingly, more than $\frac{1}{4}$ of the agent Inform statements are also elliptical sentences.

To simplify data analysis, in the following sections we assume that there is one communicative act per dialogue turn. In the annotated data, we mapped **Acknowledge** + **X** and **Confirm** +

**X** to **X** (**X** being another communicative act, such as **Inform**). For example, **Acknowledge** + **Inform** is mapped to **Inform**. All other sequences of type **Y** + **Statement** are mapped to the other communicative act **Y**. For example, **Inform** + **Statement** was mapped to **Inform**. Many clauses with the same communicative act, such as *Inform* + *Inform* were mapped to a single instance of that (e.g., **Inform**). These mappings allow one to predict the content of the next dialogue turn given the preceding dialogue turns and to distinguish between predicting the agent's actions as opposed to predicting the customer's actions, without being biased toward the most frequent communicative acts (confirmations and acknowledgments).

Figure 5-3 displays the most frequent communicative act transitions. The display indicates that the customer's communicative acts may be predicted more easily than the agent's communicative acts. For example, the most common responses to the agent's acknowledgments and confirmations are also acknowledgments and confirmations. This indicates that sequences of confirmations are joint activities that often require both speakers' participation for more than one dialogue turn [88]. We will discuss in more detail the issues in predicting communicative acts in the next section. Frequent confirmations and short, elliptical dialogue turns are typical of spontaneous dialogue. They appear frequently also in other corpus-based studies (e.g., the London-Lund corpora described in [78] and the task-oriented conversations analyzed in [105, 79]).

## 5.3 Modeling Turn Transitions within Segments

In this section we assess the extent to which a probabilistic finite state model is adequate for predicting the internal turn-taking organization of discourse segments. We use statistical language modeling concepts such as entropy and perplexity to provide an empirical estimate of the complexity of natural dialogue within each discourse segment. The term complexity is used here to mean how hard or easy it is to predict the customer's and agent's communicative acts within a discourse segment, given the preceding communicative acts. Each segment has been divided into one or two optional contributions. Rather than computing the complexity of a dialogue as a whole, we focus on request and response contributions separately, to provide more specific answers about the internal organization of segments.

### 5.3.1 Request Contribution: Co-operative Agent Behavior

Figure 5-4 lists the fraction of segments that are initiated by customer requests and the fraction of segments that are initiated by the agent. While a large majority of discourse segments are initiated by the customer, more than one out of four segments are initiated by the agent's prompts. The annotated data indicate that the conversations are truly mixed initiative. While the first segment is most frequently initiated by the customer, between 15.7% and 56.2% of the following segments

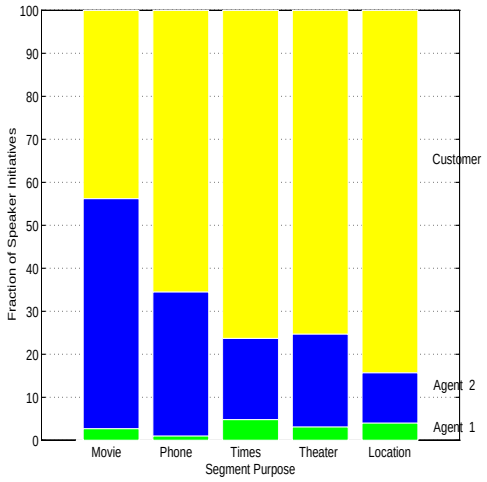



Figure 5-4: Segment initiative statistics by segment purpose. The bottom section is the fraction of agent's initiatives for the first segment (label: Agent 1). The middle section is the fraction of agent's initiatives for the subsequent segments (label: Agent 2).

are initiated by the agent with a request or inform statement, depending on the topic. We also found (not shown in the figure) that 27% of the time, the request contribution of a segment contains between one and four agent' requests for clarification, but it contains one or more customer' requests for clarification only 19% of the time. In general, customer initiatives are either *wh-questions*, direct questions, statements of need such as: *I was looking for...* or elliptical phrases such as: *the other one*. The following examples list some typical initiatives which start request contributions:

| *Customer Initiatives* | *Agent Initiatives* |
|---|---|
| C: Do you have the directions on how to get there? | A: Do you want to know what is playing there? |
| C: I was looking under the Northlake Festival. | A: Would you like help with any other movies ? |
| C: Could you just give me the phone number. | A: Which theater did you want? |
| C: What about Frankenstein? | A: Any particular area? |
| C: What time is that? | A: Do you want the phone number just in case? |
| C: And the other one. | A: Do you want those show times? |

Only a small fraction of segments contain direct request-response pairs of type: *C Request-A Inform* (e.g., Table 5.3). The majority of segments contains an acceptance contribution that follows the request for information. The purpose of such acceptance sub-segment is to set a common ground between the agent and the customer by clarifying the customer's request. In the annotated data, $\frac{2}{3}$ of the sub-dialogues that followed a request act were confirmations or acknowledgments, while $\frac{1}{3}$ were clarifications that involved specifying some additional related information, such as theater name or exact location.

**Language Model Perplexities**

To determine the complexity of the clarification section of each request, we computed the perplexity of the bigram and trigram statistical language models defined by sequences of alternate agent and customer communicative acts. This measure is used frequently to evaluate the predictive power of structural dialogue models for human-to-computer spoken language systems (e.g., [70, 51]). Perplexity is an empirical measure which estimates the average number of likely communicative acts at a particular point in time, given one or two preceding communicative acts. We distinguish between training set perplexity, test set perplexity and Markov model perplexity.

Training set perplexity and test set perplexity are derived from the entropy $K$ of the (either training or test) sequence of $N$ observed communicative acts. In the case of a unigram language model, the entropy $K$ is computed from the frequency distribution of the communicative acts:

$$K = -\frac{1}{N}\sum_{i=1}^{N}\log_2 P(a_i) \tag{5.1}$$

where $P(a_i)$ is the fraction of observed data samples tagged with the communicative act $a_i$. In the case of a bigram language model, the entropy $K$ is:

$$K = -\frac{1}{N-1}\sum_{i=2}^{N}\log_2 P(a_i|a_{i-1}) \tag{5.2}$$

where $P(a_i|a_{i-1})$ is the probability of observing communicative act $a_i$ given that the preceding communicative act is $a_{i-1}$. In the case of a trigram language model, the entropy $K$ is:

$$K = -\frac{1}{N-2}\sum_{i=3}^{N}\log_2 P(a_i|a_{i-1}a_{i-2}) \tag{5.3}$$

The bigram and trigram probabilities are estimated from frequency counts over the training set, and the sequence perplexity is then computed as:

$$P = 2^K \tag{5.4}$$

To assess the perplexities separately for each speaker, we computed the Markov language model perplexity. For example, for the agent's acts, we computed for each segment purpose the perplexity $P = 2^H$ from the Markov model's entropy:

$$H = -\frac{1}{25}\sum_{i=1}^{5}\sum_{j=1}^{5}\log_2 P(A_j|C_i) \tag{5.5}$$

where $A_j$ is an agent communicative act and $C_i$ is the customer's act that preceded it. A similar

| *Language Model* | *Perplexity* | | *% improvement over uniform model* |
|---|---|---|---|
| | *Train Set* | *Test Set* | |
| Unigram | 4.30 | 4.33 | 13.4 |
| Bigram | 3.97 | 4.08 | 18.4 |
| Trigram | 3.66 | 3.86 | 22.8 |

Table 5.5: Training and test set perplexity for predicting the sequence of communicative acts in the request contribution of a discourse segment.

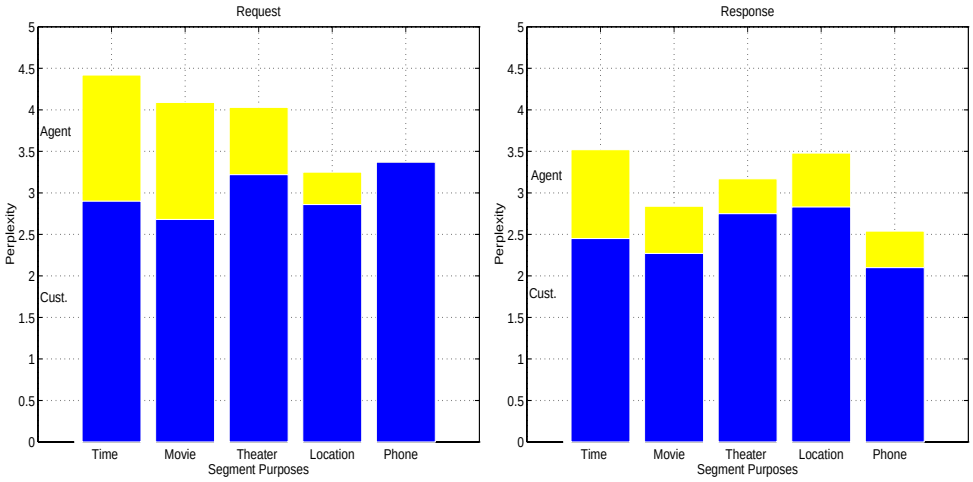



Figure 5-5: Communicative act language model perplexities for different speakers and for the request and response contributions.

equation, with $A$ and $C$ reversed, has been used for the customer's perplexity.

The annotated data provided 370 request contributions with at least two dialogue turns each, for a total of 2144 communicative acts, over an alphabet of 10 different acts. We distinguished five different acts for the agent and five for the customer, with the constraint that acts from one speaker had to be followed by one of five acts from the other speaker. We estimated the sequence perplexity by averaging twenty different runs in which we randomly split the data into 30 test contributions and 340 training contributions.

Table 5.5 lists the average perplexities for the training set and the test set. In a uniform (or maximum entropy) model all five acts would be equally likely with probability 0.2 and the model would produce a maximum perplexity of 5. A deterministic model would be able to predict exactly one act (with probability 1), and produce the lowest perplexity (equal to 1). The table indicates that with respect to a uniform model with no a priori knowledge, the trigram language model produces only a small improvement of at most 22.8% in predicting communicative acts. The small difference between training set and test set conditions indicates that the probability estimates are not poorly trained (the differences between training set and test set are within 4.8%).

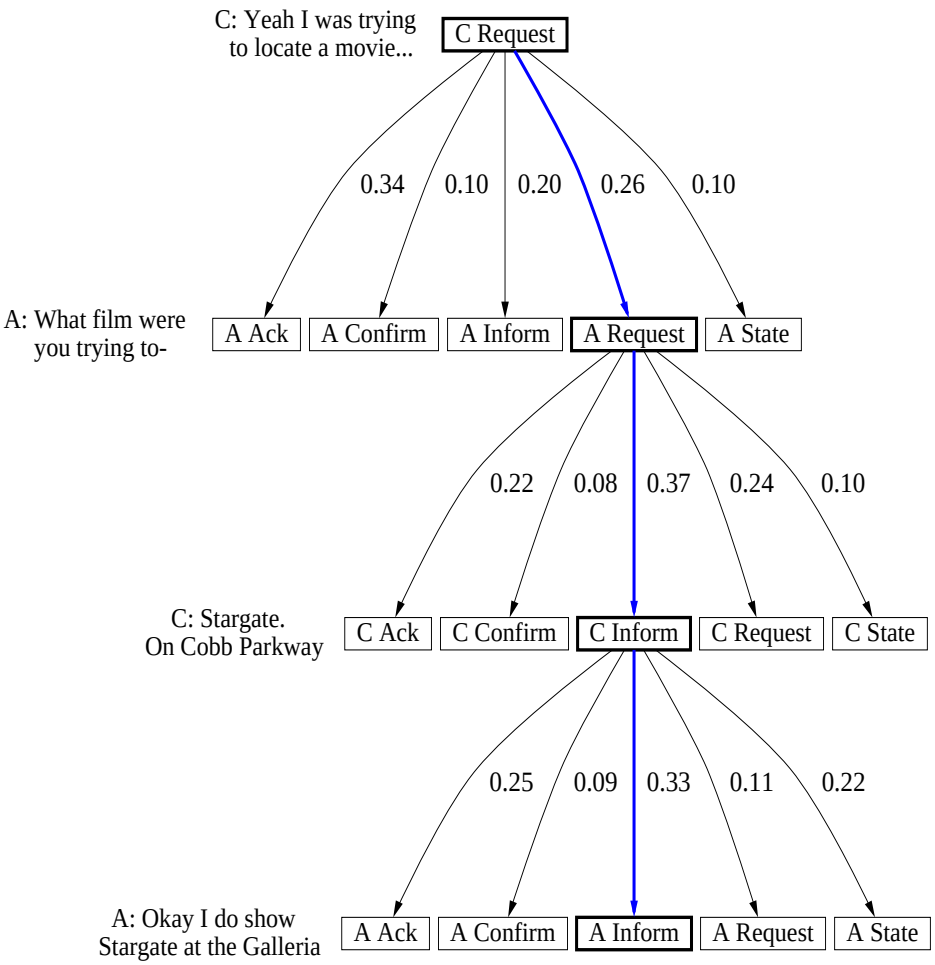



Figure 5-6: State transition diagram for a sequence of communicative acts in a request contribution. Transitions are labeled with the corresponding frequency of occurrence in the training data.

The left plot in Figure 5-5 displays how the Markov model perplexity varies with the segment purposes and the speaker. The plot indicates that the agent's communicative acts are less predictable than the customer's communicative acts, for all but one segment purpose for which the perplexities are equal (i.e., listing a phone number) . On average, using a bigram language model, the customer's acts can be predicted with a probability of 60%, a 40% improvement over the uniform model. In predicting the agent's acts, however, a bigram model improves on the uniform model by only 15%. In general, the speaker who has the initiative is the one with the less predictable behavior, and the other speaker tends to follow a more predictable pattern of responding to questions, statements and requests. While the customer tends to initiate discourse segments (see Figure 5-4), it is the agent who mostly takes the initiative of clarifying the request in the acceptance contribution of a segment.

Figure 5-6 shows an example illustrating the problem of predicting communicative acts in the request contribution for the purpose **List Theater playing Movie**. Transitions are labeled with

the probabilities estimated from a training set. Many probabilities are close to the uniform value of 0.2, confirming the fact that the model is not substantially more predictive than a uniform model. The thick edges correspond to a section of the dialogue in Figure 5-1. The last two transitions (i.e., A Request to C Inform and C Inform to A Inform) are the ones with the highest a priori probability scores. In contrast, the first transition (i.e., C Request to A Request) is less probable than a transition to an acknowledgment (i.e., C Request to A Acknowledge). This example illustrates that the a priori probabilities are not sufficient to predict the agent's actions. It is crucial to model the agent behavior because any conversational system must incorporate a computational model for generating the agent's prompts and answers. In the example, after the customer's request, the agent must decide whether to simply acknowledge it, confirm it explicitly by repeating it, clarify it or respond immediately. This decision process requires reasoning based on many different constraints such as the level of confidence of having understood the customer's words, meaning and intentions, whether or not the customer's request has been fully specified, and the intentional context in which the customer's sentence was spoken. This type of reasoning goes beyond a probabilistic finite state model.

Figure 5-7 illustrates - with four example segments drawn from our corpus - the reasoning steps involved in specifying the initial request *I'm looking for the Buford Cinema.* The four reasoning steps are the internal nodes of the tree and correspond to knowledge pre-conditions in the intentional theory of discourse [39, 37, 57, 58]. The steps are partially ordered. Recognizing the customer's words is a pre-condition to recognizing her intentions and topics. On the other hand, topics and intentions can be recognized independently of each other. If the agent needs to clarify both the intentions and the topic, she may select the order based on a particular dialogue strategy. In the theory of discourse contributions [21, 22, 23], the initial customer sentence is the presentation contribution, and the segments at the leaves of the tree correspond to different acceptance contributions. Failure at any one of the reasoning steps results in a different type of repair segment initiated by the agent. At the end of this chapter, we discuss two possible computational models which have been proposed to model the agent's reasoning steps illustrated in the figure.

### 5.3.2 Response Contribution: Reporting Information in Multiple Turns

Table 5.6 reports the training set and test set perplexities for the sequences of communicative acts in the response contribution. The measures have been estimated from a sample of 620 response segments and 2598 communicative acts. The reported measures are obtained by averaging 20 runs in which we randomly selected 30 response segments for testing and 590 segments for training. The trigram perplexity is 43% better than the uniform estimate. Compared to the request contribution (see Figure 5.5), the perplexity is reduced by 26.5%. The data indicates that the response contribution is less complex than the request contribution, and a simple structural language model

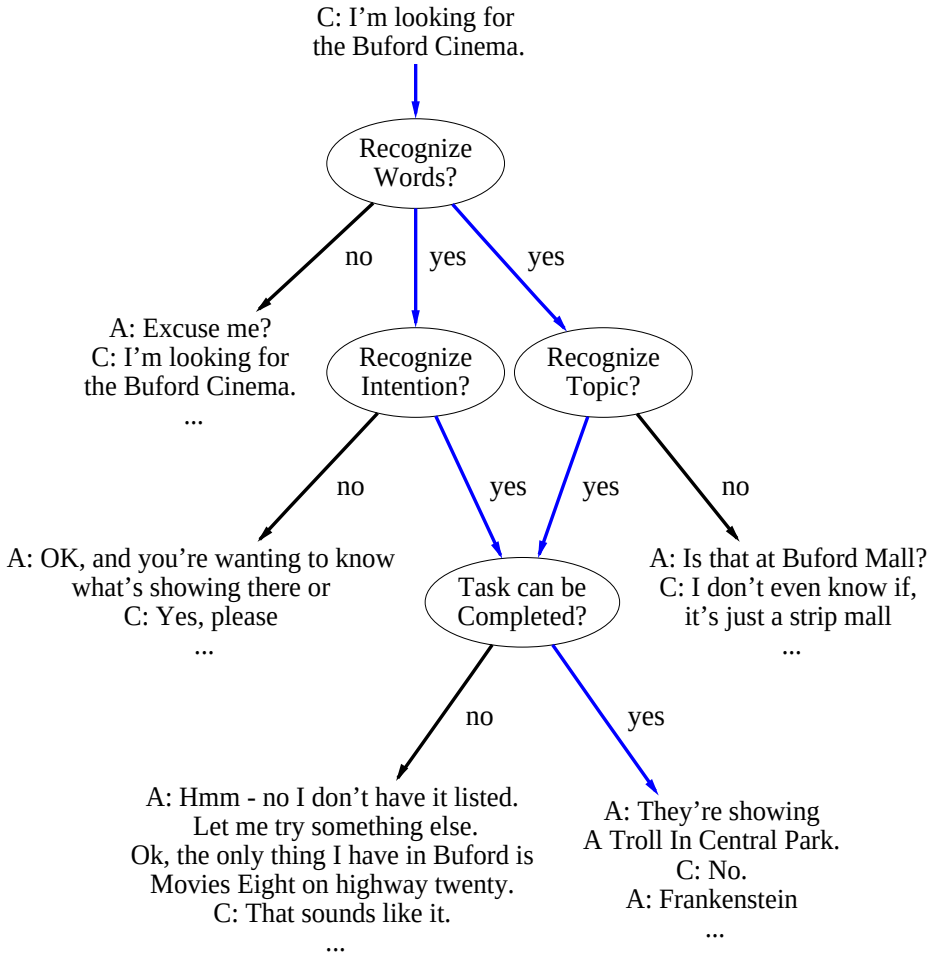



Figure 5-7: Reasoning steps involved in specifying a request for information in the movie schedule domain. The steps are indicated with circles. Each leaf in the tree is an example segment drawn from our corpus.

| *Language Model* | *Perplexity* | | *% improvement over uniform model* |
|---|---|---|---|
| | *Train Set* | *Test Set* | |
| Unigram | 3.80 | 3.85 | 23 |
| Bigram | 2.80 | 2.95 | 41 |
| Trigram | 2.66 | 2.86 | 43 |

Table 5.6: Training and test set perplexity for predicting the sequence of communicative acts in the response contribution of a discourse segment.

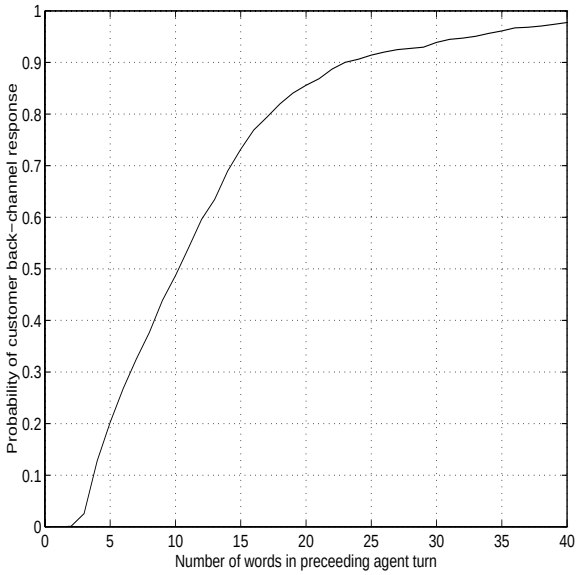



Figure 5-8: Observed frequency of customer's acknowledgments as a function of the preceding agent's A Inform dialogue turn duration.

is potentially useful in predicting the communicative acts that follow the agent's reporting of the information. In general, the response contribution consist of one or more dialogue turns in which the agent delivers the information, followed by confirmations or acknowledgments.

The plot on the right side of Figure 5-5 displays the differences in perplexities between the two speakers in the response contribution. As was the case for the request contribution, the agent's actions are less predictable than the customer's actions, indicating that the language model probabilities may be more useful for predicting the customer's actions than for predicting the agent actions. In general, the response language models have lower perplexity than the request models, indicating that responses have a more regular structure than requests.

How is the information delivered by the agent? We found that while the agent speaks in longer sentences compared to the customer, the agent delivers information using one or more relatively short turns, followed by frequent explicit confirmations and acknowledgments from the customer. Figure 5-8 is a cumulative frequency plot that displays the number of words spoken by the agent in a statement of type **A-Inform** before getting a confirmation or acknowledgment from the customer. 70% of the time, the agent does not speak more than 15 words before the customer responds with an acknowledgment. The data reported here are consistent with the analysis by Orestrom of a similar corpus of telephone conversations between British-English operators and customers (the London-Lund corpus) [78]. The average agent word count reported by that study is strikingly close to the one reported here (80% of the time the agent speaks 15 words or less, with an average of 12 words). The consistency in the results seems to indicate that there may be a practical limit of 15 words,

perhaps due to an underlying universal phenomenon of short term memory constraints for spoken words.

Figure 5-9 displays the histograms of the number of agent's **Inform** turns by segment purposes. The histograms show, for example, that 48% of the time, the agent reports multiple movie titles in two to five turns, and 52% of the time, the agent breaks down a phone number into two or three turns. Except for listing a show time, which requires one dialogue turn 78% of times, all the other information is reported in two or more dialogue turns nearly half of the time. For example, reporting multiple movie listing may require two or more dialogue turns, depending on the style chosen by the dialogue participants. In some cases, the agent lists one or two movies at a time. When the agent breaks down the presentation of the information in multiple dialogue turns, the customer has the opportunity to accept or reject the information, to acknowledge it, to confirm it, or to ask for specific details, such as show times.

Figure 5-10 compares three examples from our information-seeking corpus, one from the movie schedule domain, one from the job classified domain, and one involving giving directions to a theater. The examples have a similar discourse structure and illustrate that reporting lists of information in multiple dialogue turns is not specific to the movie schedule domain. The examples indicate that the agent presents the information using at least two levels of detail, from the generic to the specific, using mostly elliptical clauses (e.g., noun phrases). Generic information include movie titles and company names. Specific information include precise show times (e.g., eight-o-clock) and type of position available (e.g., retail managers). Breaking down the information into the generic and specific serves two purposes. First, it allows the agent to break the information into chunks that can be easily processed by the listener. Second, it provides an opportunity to the listener to explicitly accept or reject the information, therefore minimizing the likelihood of reporting information that is either given, redundant or not desired.

The examples also illustrate the differences between genres such as conversations, monologues, and written text. In a monologue or a text, the discourse would flow sequentially along the right edges of the trees. In a dialogue, at specific time instants indicated by the numbered nodes (e.g., after each movie title has been spoken by the agent), the customer has the opportunity to accept or reject the information, or to ask for specific details, such as show times, in which case the discourse enters a subsegment indicated by the edges to the left of each node. When a subsegment is concluded, the discourse flow returns to the right edge of the parent node. Psycho-linguists have argued that these opportunities for switches in speaker tend to occur at specific syntactic, semantic and intentional boundaries, (e.g., just after reporting an item in a list) and may be indicated by the agent with acoustic correlates such as pauses and raising intonation contours [85, 48]. The examples also illustrate the fact that such turn-switching opportunities exist even while the agent is speaking, whether or not she indicates them to the customer. For example, at node 6 in the movie schedule

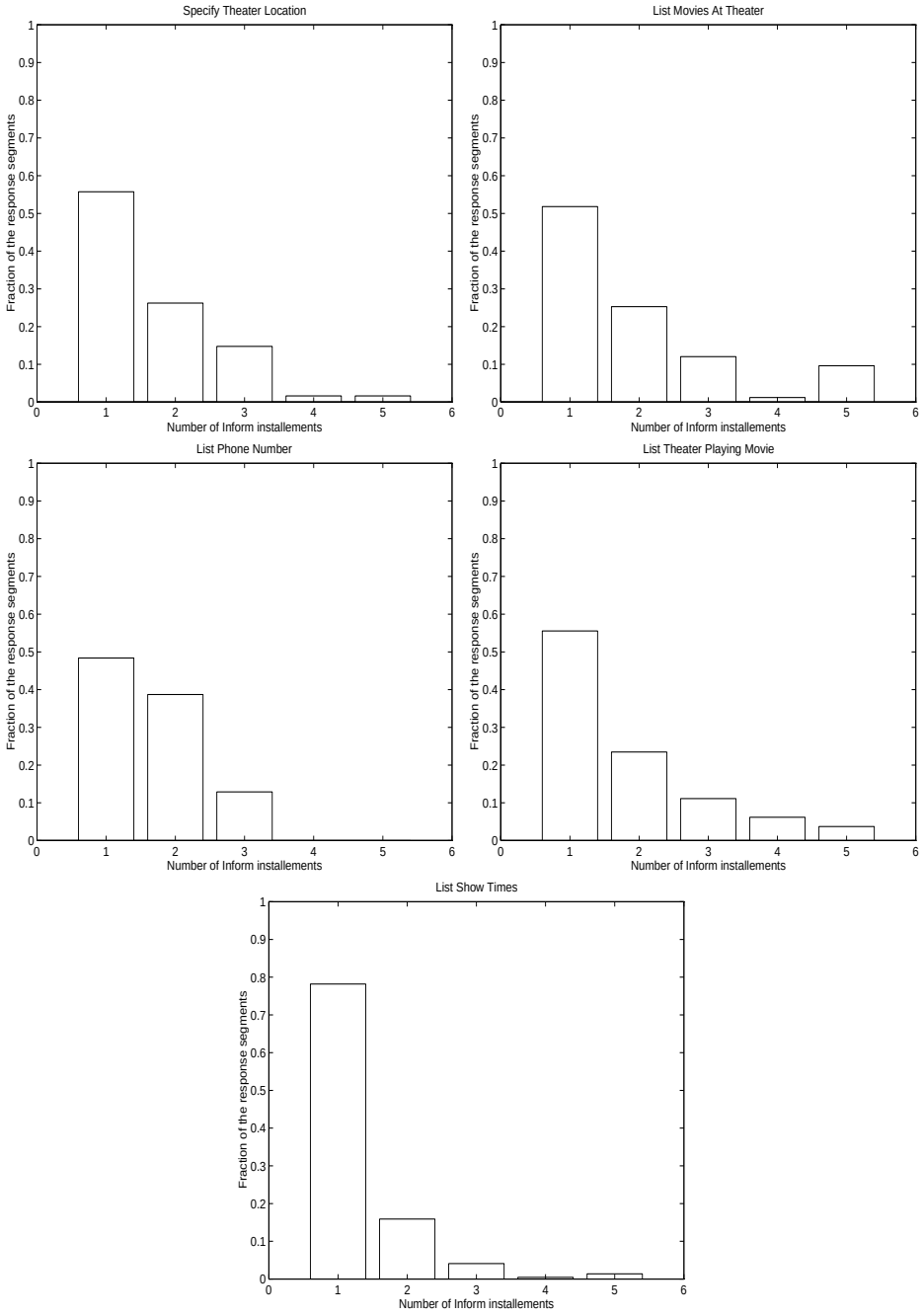



Figure 5-9: Histograms of the number of agent's Inform turns per discourse segment. For example, 48% of the time, the agent reports a phone number in a single dialogue turn, 38% of the time it takes two turns, and 12% of the time, it takes three turns.

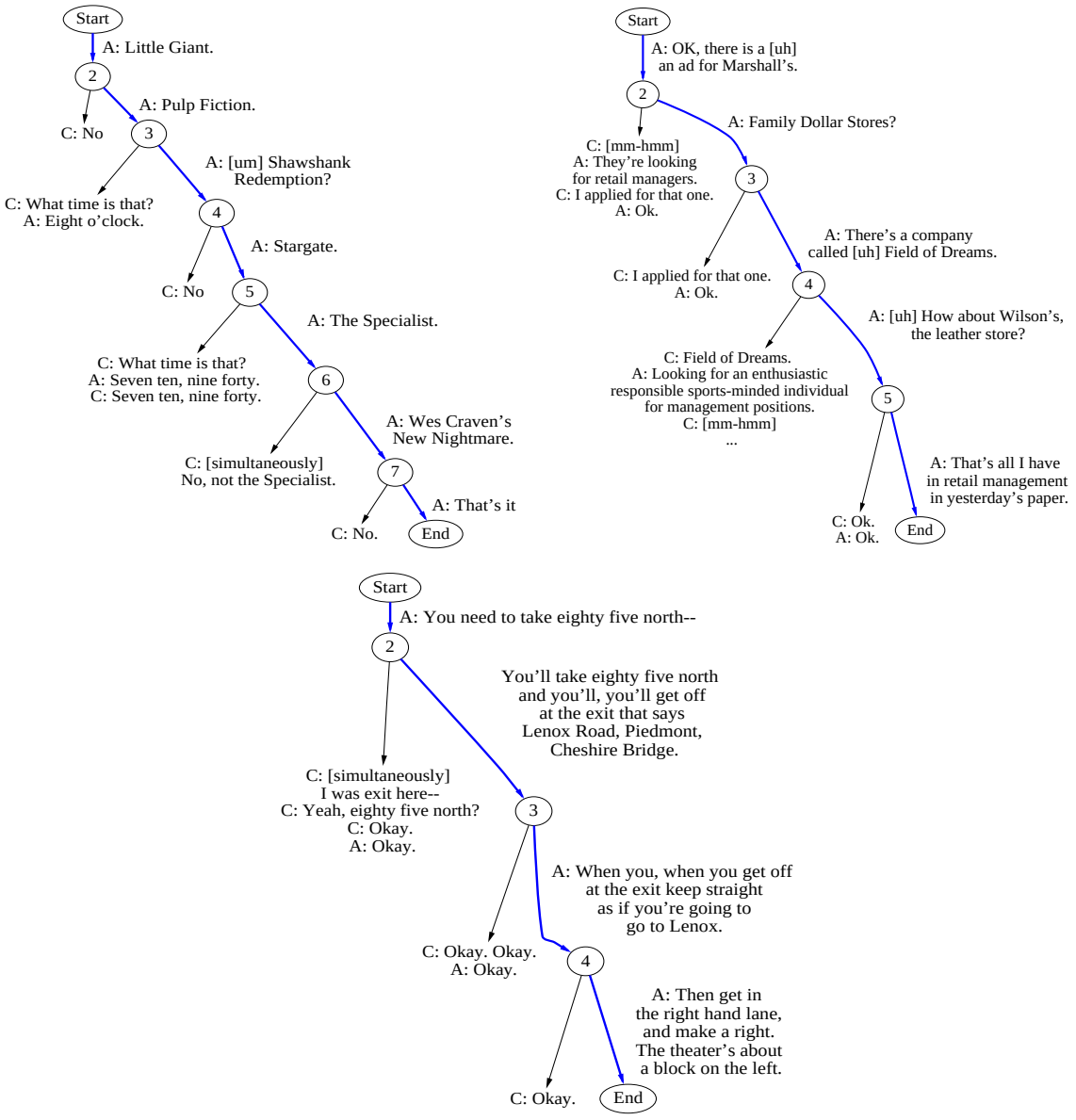



Figure 5-10: Three examples of response contributions drawn from our corpus.

listing and at node 2 in the direction listing, the customer speaks *simultaneously* with the agent, interrupting momentarily the agent's report in order to either reject some information (e.g., The Specialist) or confirm a crucial piece of information (e.g., the highway number).

The size of each item in the list is dependent on the topic. The three examples indicate that an appropriate size for each item may be a proper noun phrase for listing movie schedules and job classifieds, while in giving directions, the size of each item corresponds to one or two full sentences. In general, dividing the information into multiple short turns and breaking down the information from the generic to the specific is consistent with Grice's conversational maxims [34]. Grice enumerated generic guidelines of what should be said and how it should be said when reporting some information. Four of the maxims are:

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.
3. Be relevant.
4. Be brief (avoid unnecessary prolixity).

In summary, reporting information in natural dialogue has at least three distinguishing features: dividing the report into short phrases that can be easily memorized and referred to by the listener, adapting the level of detail of the report from the generic to the specific according to the listeners assumed knowledge and intentions, and continuously allowing the listener to confirm, clarify or interrupt the report.

## 5.4 Modeling Segment Transitions

In this section we discuss whether or not a stack data structure is adequate for modeling the type of segment transitions that we have observed by annotating the corpus of movie schedule dialogues. First, we describe the annotated segment transitions by dividing them into six different types. Second, we define the extended stack model. Third, we report a preliminary empirical evaluation of the model against the annotated data.

### 5.4.1 Data: Six Types of Segment Transitions

Figure 5-11 lists the six different types of segment transitions that we have observed by annotating the corpus of movie schedule dialogues. This taxonomy combines the types considered in the intentional theory of discourse [39, 58] with the ones considered in the theories of discourse contributions [21, 22] and of turn-taking in conversations [85, 48]. A *preliminary* segment (or pre-sequence) has a generic purpose which is the introduction to a more specific, task-oriented purpose. A preliminary purpose is a pre-condition to the purpose that follows it. In the movie schedule domain, a *new task* corresponds

**A. Preliminaries**

*1 C: is there a [ah] number that you dial*
*to just get all the different theaters?*
*2 A: I can give you that information*
*3 C: You can?*
4 C: Okay, in Snellville, Septum movies.

**B. New tasks**

1 A: And Frankenstein is playing at ten.
*2 C: No Pulp Fiction?*
*3 A: No.*
4 C: What about Marietta?
5 C: Check the Marietta Mall.

**C. Sub-tasks**

1 C: Yeah, I'm looking for the Buford Cinema.
*2 A: Is that at Buford Mall?*
*3 C: I don't even know, it's just a strip mall...*
4 A: They're showing A Troll In Central Park.

**D. Digressions**

1 C: I'm looking for Priscilla, Queen of the Desert.
2 A: Ok, one moment.
*3 C: How much does this phone call cost?*
*4 A: The first three calls are free...*
5 A: Ok, I have it showing in Marietta...

**E. Multiple active purposes**

1 C: I'm looking for the number to the Gwinnet...
*2 C: Do you know what movies are playing?*
3 A: Ok [uh] the number first is...
*4 A: They're playing The Puppet Master...*

**F. Fresh starts and repairs**

1 C: Okay, [ah] what's playing around nine forty?
2 A : [humming]
*3 C: Well what's playing period?*

Figure 5-11: A classification of segment transitions observed in the movie schedule dialogues.

| *Operations* | *Description and examples* |
|---|---|
| **Push** | Introduce a subtask, digression, confirmation or clarification. |
| | *C: And, do you know what's playing at the theaters today?*<br>**Push** *A: What part of town?* |
| **Pop** | Acknowledge completion of purpose. |
| | **Push** *A: What part of town?*<br>*C: Buckhead.*<br>**Pop** *A: Ok, In Buckhead, at Litchfield Ostelle...* |
| **Next** | Complete the current purpose<br>and propose a new one for evaluation. |
| | *A: It's at the Movies at Gwinnett*<br>**Next** *C: Do you have anything else like maybe Douglasville?* |
| **Swap** | Switch focus between the top two purposes. |
| | **Next** *C: I'm looking for the number to the Gwinnet Cinema...*<br>**Push** *C: Do you know what movies are playing?*<br>**Swap** *A: Ok [uh] the number first is...*<br>**Swap** *A: They're playing The Puppet Master...* |
| **Replace** | Replace the just introduced purpose with a fresh start or repair. |
| | **Next** *C: Okay, [ah] what's playing around nine forty?*<br>*A: [humming]*<br>**Replace** *C: Well what's playing period?* |

Table 5.7: List of five possible stack operations which are used to model discourse segment transitions in information-seeking dialogues.

either to a customer request for new information, or to an agent proposal to list new information, such as a show time. A *sub-task* is a clarification or confirmation sub-dialogue, i.e., a pre or post-condition to the top-level segment that contains it. A *digression* (or diversion) is a segment which is not related to the top-level segment purpose that contains it. *Multiple active segments* may alternate in a dialogue section. In the movie schedule corpus, we did not observe more than two active purposes at any time. Finally, a *repair* occurs whenever a new purpose is introduced and immediately replaced by another one, within a few (usually one or two) dialogue turns. Segment boundaries for new tasks, sub-tasks, digressions and multiple active purposes can be annotated reliably by trained coders using a linear annotation coding scheme. However, the inter-coder agreement experiments reported in Chapter 4 indicated that, because of their transitory nature, the exact boundary locations for preliminaries and repairs is annotated less reliably using linear annotations.

### 5.4.2 Model: A Stack with Extended Operations Is Still A Stack

The model we consider for segment transitions is an extension of the stack model proposed by [35] and reviewed in [21]. In this section we present the extended model and discuss how closely it is related to the original model. In the original model, a stack provides a priority list of all of the segment purposes. The customer and agent take turns applying sequences of elementary push and pop operations to a stack that represents the focus of attention in the dialogue. A segment initiative pushes a new segment purpose onto the stack. For example, the request *What time is it playing?*

would push the purpose **List Show Times For Movie** onto the top of the stack. After one or more dialogue turns, the corresponding agent Inform statement *It is playing at three and five* would match the purpose at the top of the stack. After a few other confirmation turns, the purpose would be considered completed and it would be popped out of the stack as soon as one of the speakers introduces a new purpose for consideration, or returns to a previous purpose which can be completed. The stack model considered here extends the basic push and pop operations with three additional operations: *Next*, *Swap* and *Replace*. The model is summarized in Table 5.7. In this model, a single dialogue turn or communicative act may correspond to a *sequence* of a few stack operations. From a computational point of view, only two basic operations are required: push and pop. The other three can be implemented by sequences of two or more of the basic operations, as explained below.

*Next* and *replace* correspond to the sequence: *pop, push*. If a push-down automaton $A$ is able to process the stack in a dialogue by applying push and pop operations, then another push-down automaton $B$ (with a bigger state space) is also able to implement the *swap* operation by the sequence: *pop, pop, push, push* provided that each state in the automaton $B$ encodes information about the two symbols on top of the stack. When the original automaton $A$ is in state $i$ and the top of the stack contains the symbols $a, x$ the extended automaton $B$ is in state $[i, a, x]$. When $A$ moves from state $i$ to state $j$ and pushes $b$ onto the stack (which becomes $b, a, x$) $B$ moves to the state labeled $[j, b, a]$ and pushes $b$ onto the stack. At that point, a swap operation can be computed by first popping the top of the stack twice (which leaves $x$), then pushing $b$ fowolled by $a$ onto the stack (which becomes $a, b, x$), after which $B$ moves to state $[j, a, b]$. In general, the swap operation can be extended to permute the order of the top $N$ elements of the stack (with $N \geq 2$ and finite) and a push-down automaton is still able to model it, provided that each state in the extended automaton encodes knowledge about the top $N$ elements on the stack.

### 5.4.3 Preliminary Evaluation: Stack Operations Are Adequate

We tested empirically the coverage of the stack model using the 190 text transcriptions annotated with linear segment boundaries, segment purpose labels, request and response contributions, and communicative acts. An automated script ran through the annotated corpus, simulating a stack by pushing the annotated segment purpose at every segment initiative, and then searching for it in the stack at the beginning of the corresponding response contribution. If the response follows the request within the same segment or separated by a subtask or digression segment, the purpose should be found at the top of the stack. Every time the response does not match the purpose on top of the stack, there is a violation of the stack-based hypothesis.

The stack depth measures the number of purposes that are active at any point in the dialogue, because they are not yet answered by the appropriate agent Inform statement. We found that 57% of the dialogues have a stack depth of 1 - indicating that a majority of dialogues are realized as

| | |
|---|---|
| **Request 1: List Movies At Theater** | |
| 1 C: Yeah, [um] I'm looking for the Buford Cinema. | **(push 1)** |
| 2 A: OK, and you're wanting to know what's showing there or | |
| 3 C: Yes, please. | |
| 4 A: Are you looking for a particular movie? | |
| 5 C: [um] What's showing. | |
| 6 A: OK, one moment. | |
| 7 C: OK. | |
| **Segment 2: Specify Theater Location** | |
| 8 A: Is that at Buford Mall? | **(push 2)** |
| 9 C: [um] I don't even know if, it's just a strip mall. | |
| 10 A: I mean is it | |
| 11 C: It's off of twenty. | |
| 12 A: Yeah, that's it. | |
| 13 A: It's called Movies Eight or Buford Mall Theater. | |
| 14 C: That's | |
| **Response 1: List Movies At Theater** | |
| 15 A: They're showing A Troll In Central Park. | **(pop 2)** |
| 16 C: No. | |
| 17 A: Frankenstein... | |

Figure 5-12: An example section of a conversation annotated with stack operations.

linear sequences of *request-response* contributions. We also found that 32% of the dialogues have a stack depth of 2, indicating a maximum of two concurrent active purposes (often they are a task and a diversion task such as **List Show Times** and **List Phone Number For Theater** or a main task and a related subtask, such as **List Show Times** and **List Theater Playing Movie**). We found that 10.4% of the dialogues have a stack depth of 3 or more. The example displayed in Figure 5-12 illustrates how the stack can reach a depth of 2 in case of clarification and digression segments. After line 14, when the agent and the customer have set a common ground by specifying the name and location of the theater, the second purpose is considered completed and pops out of the stack, and the agent can complete the first purpose, which is to list the movies at that theater.

We found that 90% of the time, the Inform statement either matched the purpose on top of the stack or was some over-informative response volunteered by the agent, and 10% of the time, the Inform statement did not match the top of the stack, indicating a violation of the simple stack based processing hypothesis. Closer examination of this data revealed that such violations corresponded to either multiple active purposes or transitory events such as preliminaries and repairs.

The two dialogues displayed in Figures 5-13 and 5-14 illustrate two typical cases in which segments apparently violate the strict push-pop process. Both examples can be adequately processed when the stack is augmented by the swap operation. In particular, at line 34 in Figure 5-14, the agent pushes new content onto the stack, without first clearing the stack because the customer has not acknowledged explicitly having received and understood the information. In line 35, the customer switches the top two elements on top of the stack. Note that a single communicative act or dialogue

| | |
|---|---|
| **Request 1: List Phone Number** | |
| 1. C: [um] I'm looking for the number to the Gwinnett Cinema and | |
| 2. A: [mm-hmm] | **(push 1)** |
| **Request 2: List Movies at Theater** | |
| 3. C: Do you know what movies are playing? | |
| 4. A: Yeah, I could tell you what's playing there. | |
| 5. C: OK. | **(push 2)** |
| **Response 1: List Phone Number** | |
| 6. A: OK, [uh] the number first is seven two si- I'm sorry. | **(swap)** |
| 7. A: Seven six two, nine six three six. | |
| 8. C: OK. | |
| **Response 2: List Movies At Theater** | |
| 9. A: OK, they're playing The Puppet Masters, Wes Craven's New Nightmare, | **(pop 2)** |
| 10. C: [mm-hmm] | |
| 11. A: Little Giants, Pulp Fiction, Time Cop and that's all. | |
| 12. C: Oh, OK. | |

Figure 5-13: An example of how multiple active purposes can be processed by the *swap* operation.

| *Transcription* | *Operations* | *Stack* |
|---|---|---|
| 31 A: Stargate. | | **listing1** |
| 32 C: What time is that? | **(push time1)** | **time1** **listing1** |
| 33 A: Seven ten and nine forty. | | **time1** **listing1** |
| 34 A: The Specialist. | **(push listing2)** | **listing2** **time1** **listing1** |
| 35 C: Seven ten, nine forty. | **(swap)** | **time1** **listing2** **listing1** |
| 36 C: No, not the Specialist. | **(pop time1)** | **listing2** **listing1** |
| | **(pop listing2)** | **listing1** |
| | **(pop listing1)** | |
| 37 A: Wes Craven's New Nightmare. | **(push listing3)** | **listing3** |
| 38 C: No. | | **listing3** |
| 39 A: That's it. | | **listing3** |
| 40 C: Thank you. | **(pop listing3)** | |

Figure 5-14: Another illustration of how the *swap* operation can be used to model non-sequential events.

| *Transcription* | *Operations* |
|---|---|
| 20 A: I have it at the Cobb Place Eight | |
| 21 C: Is that it? | **(push question1)** |
| 22 A: They're at Parkway near Highway Forty One | **(pop question1)** |
| 23 C: Is that the only one? | **(push question1)** |
| 24 C: Is it at Galleria? | **(replace question2)** |

Figure 5-15: An example of repair which can be handled by the *replace* operation.

turn may correspond to a *sequence* of stack operations.

Fresh starts and dialogue repairs can also be represented by another stack operation, *replace*, which simply replaces the (partially specified) content at the top of the stack with some new content which is introduced by the fresh start or repair. Figure 5-15 illustrates how the *replace* operation is applied to the top of the stack to substitute the just question just introduced with another one.

This preliminary evaluation informally illustrates the feasibility of a stack model for discourse structure. Preliminaries and new task segment transitions are processed by the *next* operation. Subtasks and digressions are processed by *push* and *pop*. Multiple active purposes are processed by *swap*, and fresh starts and repairs by *replace*. A more formal treatment of an underlying computational model migth based on the foundations set forth by [39] and [21]. For example, a detailed model is necessary to distinguish between a fresh start (handled by the *replace* operation) and multiple active purposes (handled by the *swap* operation).

## 5.5 Discussion

### 5.5.1 Modeling Co-operative Agents

In spite of the apparent regularity of the observed discourse segment structure in terms of request-response and presentation-acceptance adjacency pairs, we found that the internal organization of each discourse segment is highly interactive. Even the simplest *Request - Response* segment requires many communicative acts in which both speakers are involved in setting a common ground by clarifying the request and confirming the information received. When we computed the statistical language model perplexity of the request and response contributions of each segment, we found that communicative acts in the request contribution were less predictable than those in the response contribution, and that the agent communicative acts were less predictable than those of the customer. A finite language model such as a trigram may be useful in predicting the customer's next action from the few preceding actions, but it seems inappropriate in helping to predict the agent's behavior.

One alternative to finite state models is the co-operative plan-based approach (see [39, 37, 57, 58, 24, 86, 87], among others). In this approach, the dialogue state and sentence meaning are represented by a set of logical predicates. Communicative acts are operators which have predicate pre-conditions

and side effects - they may add or delete predicates to the dialogue state. Segment purposes and speaker intentions are represented by plans - sequences of operators that lead from an initial dialogue state to predicates that describe some goal, such as reporting specific values for a theater name and show times. Interpretation of the speaker's intentions from the sentence meaning and the dialogue state is performed by automated inference, or theorem proving. If the set of predicates is found to be inconsistent or ambiguous, the agent can initiate a clarification sub-dialogue expressed as a repair plan. The CNET Artimis system is an example of a deployed co-operative spoken language system that uses a set of rationality principles and a detailed inference model for recognizing the speaker's meaning and intentions and generating the system responses [86, 87]. In order to circumvent the state space explosion problem faced by finite state models, another alternative has been to represent the state space by equivalence classes that contains several individual states at once. Systems such as CSELT DIALOGOS [30], AT&T Amica [81] and MIT Bianca [94] apply the principles of propositional production systems [76, 77]. According to Olsen's definition [77], a propositional production system consists of a state space of equivalent classes and a set of rules. The equivalent classes are determined by a set of predicates over a finite set of conditions. At every dialogue turn, the rules are applied in order. They test the predicates to determine which communicative act is appropriate. Once a communicative act is completed, one or more condition has been changed as a side effect, resulting in a new dialogue state. The predicates may encode knowledge about expected and recognized topics and intentions. Propositional production systems are equivalent to simple propositional logic [76, 77], which does not infer logical consequences beyond the ones expressed by atomic logical conditions. In this model, intentions and purposes are encoded explicitly a priori by one or more of the atomic conditions. In contrast, inference models such as Artimis's system apply first-order predicate calculus, which is a more complex computational model [109, 74]. First-order logic inference allows one to deduct the truth value of predicates such as the speaker's beliefs, intentions and purposes from other related predicates that encode the meaning of a sentence, by way of automated reasoning. However, automated reasoning may be combinatorially intractable, unless specific heuristic search strategies are employed.

### 5.5.2 Modeling Segment Transitions with a Stack

The computational power of the three dialogue models discussed in this chapter varies from very simple to quite complex when we consider in turn finite state models, propositional production systems and automated reasoning. The data analysis reported in this chapter does not answer the question of whether first-order logic is more appropriate than propositional logic for modeling a natural task-oriented dialogue. However, the analysis of the annotated corpus indicates that a stack can be used to model segment transitions.

The task structure of the movie schedule domain is relatively simple. It involves database queries

such as selecting theaters by location or by movie title, and reporting show times. It does not involve many negotiating steps. Completing a task in this domain does not require the completion of more than two other related subtasks, such as selecting a movie by title and theater location. When we analyzed the dialogues using stack operations that matched requests for information with the corresponding inform statements, we found that a majority of segments were indeed realized as matched *Request-Response* pairs, that 87% of dialogues processed a maximum of two concurrent active purposes, and that simple *push-pop* stack processing was violated only 10% of the time. The instances where the stack data structure seems inappropriate may be handled if sequences of push and pop operations are combined to form the *swap* and *replace* operations. While these results are preliminary and specific to an information-seeking domain, we believe there is a general cognitive limit on the number of multiple purposes that can be managed in spoken dialogues, and on the ability of speakers to handle non-sequential reasoning and planning. The empirical results support the hypothesis that a stack data structure is potentially very useful as a computational device for handling the focus of attention in a natural task-oriented dialogue.

The inter-coder agreement results presented in Chapter 4 indicate that much work remains to be done when designing an annotation coding scheme that would represent accurately all types of segment transitions. In particular, coders tend to agree about where to place linear segment boundaries, while it is more difficult to reliably annotate embedded segments and events such as preliminaries and repairs. One possibility to explore would be to first annotate linear segments with their purposes as specified in Chapter 4, and then to annotate co-reference relations among segments. The co-reference relations should link pairs of segments and should be a small set (i.e., *is same task as*, *is subtask of*, and *is digression from*). In order to limit the cognitive load for the coders and to produce more reliable results, the annotation process and the evaluation can be explicitly divided into two separate sessions. Another possibility to explore would be to annotate the text transcriptions directly with stack operations.

### 5.5.3 Levels of Interaction

Natural spoken dialogue is highly interactive. We can associate interactivity empirically with five measures. The first measure is the number of dialogue turns required to complete a request or a response contribution. On average the number of dialogue turns required to get a response from the database is high, between 3 and 5 dialogue turns. The second measure is the level of mixed initiative. In the annotated data, the customer takes the initiative in the first segment, but the agent may take the initiative for the following segments between 15.7% and 56.2% of the time, depending on the segment purpose. We also found that 27% of the time, a request for information will be followed by at least one agent's request for clarification before obtaining a response from the database. The third measure is the number of acknowledgments and confirmation acts. In our corpus, nearly half of

the communicative acts are acknowledgments and confirmations. These frequent grounding acts are required for confirming the request as well as the reported information. The fourth measure is the size of each dialogue turn. We found that each dialogue turn is short in size, on average between 3 and 12 words. The fifth measure is the number and type of turns needed to report some information. We found that the agent very often reports movies, show times and theater names in multiple turns. The agent Inform statements are short (15 words or less on average), and at least 31% of them are elliptical sentences. The many confirmations are even shorter (9 words or less). Collectively, the findings presented in this chapter provide substantial empirical evidence for theories of dialogue as a joint activity in which discourse segments are initiated by either speaker with the purpose of either finding a solution to the task at hand [39] or repairing and preventing misunderstandings [22]. In addition, the statistics reported here are consistent with at least two other empirical studies conducted on a similar corpus of British-English telephone conversations, the London-Lund corpus [99] analyzed by Orestrom [78] and by Clark and Schaefer [22].

The movie schedule domain is within reach of state-of-the-art spoken dialogue systems. How can our findings be applied to spoken language systems? Based on the analysis of human-to-human conversations, we believe that a user friendly dialogue system should not assume that even a simple task such as a database query can be implemented by a simple question-answer pair. A natural dialogue may involve a large number of dialogue turns. As a consequence, measuring success by the number of dialogue turns required to complete a task may be misleading, since more dialogue turns may indicate a more natural conversation. Carrying out a natural conversation with a user requires at least two features. First, it should be possible to clarify the user's request using sub-dialogues that may require more than one or two turns. Clarification sub-dialogues may be motivated by the task structure (setting a parameter that is missing) or by a low level of confidence in the recognized word sequence. Second, the information should be reported in small, interruptible chunks of 15-20 words or less. We believe that controlling the level of the system's verbosity depending on the dialogue context is an important area of research for building systems that are more user friendly. For example, shorter rather than longer Inform and Confirm statements should be preferred. Should a system be able to both understand and produce acknowledgments and short confirmations, such as *hmm-hmm* and *okay*, as in the following dialogue exchange?

| | |
|---|---|
| C: I'm wondering if you can give me a fare<br>from Albuquerque to Detroit on the twenty seven of July. | Present. |
| A: Okay.<br>A: I will sure check for you.<br>C: Thank you | Accept. |

In natural telephone conversations, acknowledgments and confirmations are not just noise and fillers. They signal that the speakers are sharing a common ground, encourage them to complete the ongoing

task, or signal that a dialogue repair or additional confirmation is needed. From the point of view of speech and natural language understanding, it is yet to be determined whether users will talk to a machine the same way they would talk to a human, with many *hmm-hmms* and *okays*, and whether or not it is appropriate for a machine to utilize the same type of acknowledgments. While a machine may apply the same principles of co-operative behavior as a human agent, many dialogue system designers argue that it may more appropriate to employ more explicit prompts and feedback responses [7, 40, 110, 12, 30]. Explicit prompts and responses would take into account the fact that machines tend to make more errors than humans, and that they share with users only a very limited set of domain-specific knowledge bases, intentions, and goals.

# Chapter 6

# Conclusion

In this final chapter, we summarize our contributions in the area of text analysis of conversations and point to future research directions. The focus of this thesis has been on analyzing the text content of conversations (i.e., *what* people say). To limit the scope of our research, we left two important issues to future research: intonation contours and timing (i.e., *how* people speak).

## 6.1 Summary: What People Say In Conversations

A major part of this thesis has been an empirical exploration of the underlying data structures that can model what people say in information-seeking conversations. Our approach has been data-driven. Rather than postulating a theory first, and then seeking evidence to support it, we first let many coders annotate text transcriptions with limited instructions. We were then able to correlate the segmentations proposed by a majority of coders with theories that model conversation as a highly structured collaborative process [21, 39].

### 6.1.1 Discourse Segmentation from Text Can Be Performed Reliably

For a long time, the reliability of discourse segmentation of text has been a controversial issue. In this thesis, we have shown that coders with no prior knowledge of discourse analysis can reliably annotate discourse segment boundaries in text transcriptions of task-oriented conversations. We have achieved reliable segmentations by three means. First, the data to annotate was inherently structured. We deliberately chose a corpus of information-seeking dialogues because this is the genre for which we are interested in building better spoken dialogue systems. Secondly, we provided the coders with an easy-to-use graphical user interface that clearly displayed embedded discourse segments, and allowed them to edit the segmentation in a few steps. The tool, named *Nb* , is described in detail in Chapter 3. Thirdly, the design of the coding scheme has been iterative. Initially, we discovered

where people disagreed in assigning segment boundaries in an unconstrained experiment. Then, we included specific instructions precisely to overcome disagreements. The instructions which produced the most reliable results were the ones that minimized the decision process of the coders, allowing them to choose among a few independent alternatives.

The set of four annotation experiments described in Chapter 4 indicate that it is possible to reliably annotate the intentional discourse segment structure of information-seeking dialogues. However, while coders tended to agree in annotating segments with purposes that were assigned a priori, they tended to disagree in assigning a specific hierarchy between the segment purposes. In this respect, our results are not conclusive, and more annotation experiments are needed to settle this issue. We believe there are two possible explanations for the disagreement about hierarchical segmentations. Firstly, as mentioned by Grosz in [105], the structure of information-seeking dialogues is sequential rather than hierarchical. For example, in Chapter 5 we accurately modeled the movie schedule dialogues as sequences of request-response contributions, with no immediately clear dependence relationship between different purposes such as **List Movies At Theater** and **List Phone Number For Theater**. Secondly, the embedding that we observed in information-seeking dialogue has been mostly of clarifications and confirmation sub-dialogues. While we did not explore the reliability of annotating this type of embedded segments with specific labels such as **Clarify The Request** and **Confirm The Response**, we did show in Chapter 5 that they can be annotated consistently as the acceptance phase of discourse contributions.

### 6.1.2 Stack Operations Model Segment Transitions

In Chapter 5, we discussed what data structures are appropriate to model the spontaneous dialogue phenomena such as sub-dialogues (i.e., clarifications, confirmations and diversions), preliminaries, repairs and switches between two active purposes. We have argued that the stack data structure proposed in [35] is appropriate to model the focus of attention of the conversation participants during all of the examples which have been observed empirically. In particular, new tasks can be handled by the *next* operation, sub-dialogues can be managed by the *push* and *pop* operations, repairs can be managed by the *replace* operation, and purpose switches can be managed by *swap* operations (note that *next, replace* and *swap* are shorthands for sequences of *push* and *pop* operations). In addition, we argue that *replace* and *swap* are operations with limited scope. In particular, a *replace* can be performed only within one or two dialogue turns (i.e., it is not plausible to replace anything but the top of the stack, and only if it was introduced within the last one or two dialogue turns). This constraint is motivated by previous empirical studies in speech repairs which indicated that repairs occur sooner rather than later [90, 56, 89, 20]. In particular, this constraint is consistent with Clark's principle of repair [21]:

*When agents detect a problem serious enough to warrant a repair, they try to initiate and repair the problem at the first opportunity after detecting it.*

We argue that switching between segment purposes is also constrained by cognitive processes (e.g., limited memory for spoken words). In particular, the examples from our corpus demonstrate that speakers are able to refer to two previously mentioned independent active purposes, such as listing a phone number and listing show times. On the other hand, there is no empirical evidence in our corpus that speakers are able to seamlessly switch between three or more active purposes. A limited number of purpose switches can be adequately managed by a *swap* operation applied to the top elements of the focus of attention stack.

## 6.2 Future Directions: How People Say It

This thesis explored only a few aspects of the discourse structure of spoken dialogue, and many issues which are very relevant to developing effective spoken dialogue systems have been left out.

### 6.2.1 Reporting Information Efficiently

In Chapter 5, we have provided some examples of how the agent report lists of information in multiple turns, going from the general to the specific, and only providing detailed information on demand. One of the biggest challenges to developing telephone applications is precisely the fact that users only remember the last few words of what is spoken. As a consequence, spoken dialogue systems must incorporate computational models that specify how to break textual information into multiple short discourse contributions. While computational models of speech generation in dialogue systems can be inspired by text planning and generation algorithms [49, 62, 61], the analysis of human-to-human dialogues is instrumental in determining the size and type of discourse contributions that are appropriate in the context of interactive spoken communication.

### 6.2.2 Intonational Contours and Discourse Cues

Perhaps the biggest limitation of this thesis has been that discourse annotation has been performed from text alone. However, speakers convey changes in their intentional and attentional state using a combination of lexical, acoustic and prosodic cues such as discourse cue words, pause duration, speech signal amplitude and pitch contour [82, 46, 100, 36, 72, 52]. Hirshberg and Nakatani report very reliable results in annotating segments by listening to the speech signal as well as reading the text transcription [45]. The EVAR system, described in [52], illustrates how prosodic cues have been

successfully integrated into a spoken dialogue system precisely to interpret the speaker's intentions. For example, consider the following dialogue exchange, extracted from Chapter 5:

| | |
|---|---|
| A: Seven six two, nine six three six. | |
| C: OK, that's seven six two | *Declarative* |
| A: [mm-hmm] | *Acknowledgment* |
| C: Nine three? | *Interrogative* |
| A: Nine six three six. | *Explicit Confirmation* |
| C: Nine six. OK | *Declarative* |
| A: OK, they're playing The Puppet Masters, | *Switch in Purpose* |

In order to correctly respond to the customer, the agent must detect changes in intonational contour, as well as lexical discourse cues such as *OK*. Implicit requests for confirmation (e.g., *Nine three?*) are frequently spoken with a raising intonation contour, while acknowledgments (e.g., *Nine six. OK*) tend to be spoken with a falling intonation contour, and do not require that the agent further explain or repeat the information.

### 6.2.3 Collaborative Timing

Spoken dialogue is a *real-time* collaborative process in which speakers take the initiative at specific instant in time [20, 21, 85]. Consider the following example:

25 A: You need to take eighty five north–
*26 C: [simultaneously] I was exit here–*
*27 C: Yeah, eighty five north?*
*28 C: Okay.*
*29 A: Okay.*
30 A: You'll take eighty five north and you'll,
31 A: you'll get off at the exit that says Lenox Road,
32 A: Piedmont, Cheshire Bridge.

According to Clark's principle of repair which we stated at the beginning of this chapter, speakers attempt to repair a communication problem as soon as they detect it. In the example, as soon as the customer detects a misunderstanding of the highway number, she initiates a short confirmation sub-dialogue. The cost of misunderstanding is so high for the customer that she chooses to deliberately talk simultaneously with the agent. On the other hand, the agent must be able to listen for requests for confirmation *while* she is speaking. If a text corpus is time aligned to the corresponding speech signal, it is possible to conduct empirical studies that can provide precise answers about the timing, size and type of simultaneous contributions [20].

# Appendix A

# The Group-wise Kappa Coefficient

The coefficient $\kappa$ can be derived from the observed agreement $P_o$ and the chance agreement $P_c$:

$$\kappa = \frac{P_o - P_c}{1 - P_c} \tag{A.1}$$

In chapter 2, we demonstrate how to compute $\kappa$ for the simple case of two coders annotating two categories. This appendix demonstrates how to compute the group-wise coefficient in case of:

- more than two coders
- more than two categories
- missing annotated data

Because it is not easy to find the exact definition of the chance probability for this extended case, we report here how it has been computed for the evaluations discussed in Chapter 5. The computations are based on the article by Uebersax [104]. The $C$ source code of the computations can be downloaded by anonymous ftp to the site: `ftp.sls.lcs.mit.edu/pub/flammia/kappa.c`

Let $N$ be the number of coders, and $i, j$ be two different coders. Let $C$ be the number of categories, and $c, d$ be two different categories. $D$ is the number of data samples (e.g. text lines) which have been annotated by the coders. Different data samples are indicated with $m, n$. The chance and observed agreements are derived from the data matrix $R$. $R$ is a $N \times C \times D$ matrix. Each element $R(i, c, n)$ is set to 1 if coder $i$ assigns category $c$ to data sample $n$. If coder $i$ does not set $n$ to $c$ or does not set $n$ to any category (missing data) then $R(i, c, n)$ is set to zero.

## A.1 Chance Agreement

The chance agreement $P_o$ is the probability that any two coders $i$ and $j$ would agree a priori, given their observed marginal frequency distributions for each category $c$:

$$P_c = \sum_i \sum_{j \neq i} \sum_c PM(i,j,c) \times C(i,j) \tag{A.2}$$

$P_c$ is the sum of the products of marginal distributions $PM$ normalized by the total count $C$.

For each pair of coders $i$ and $j$ and each category $c$, the product of marginals is the product of the total counts of observing $i$ using category $c$ and $j$ using category $c$:

$$PM(i,j,c) = \sum_n [\frac{R(i,c,n)}{\sum_d R(i,d,n)} \times \frac{R(j,c,n)}{\sum_d R(j,d,n)}] \tag{A.3}$$

The normalizing count $C(i,j)$ ensures that the chance agreement's range is between 0 and 1:

$$C(i,j) = \frac{\sum_n [\sum_c R(i,c,n) \times \sum_d R(i,d,n)]}{\sum_n [\sum_i \sum_c R(i,c,n)] \times [(\sum_j \sum_d R(j,d,n)) - 1]} \tag{A.4}$$

If all of the coders have annotated all of data samples into one out of $C$ categories, the normalizing count reduces to:

$$C(i,j) = \frac{1}{N \times (N-1)} \tag{A.5}$$

## A.2 Observed Agreement

The observed agreement $P_o$ counts the fraction of data samples that have been annotated with the same category by pairs of coders:

$$P_o = \frac{\sum_c \sum_n [\sum_i R(i,c,n)] \times [(\sum_j R(j,c,n)) - 1]}{\sum_n [\sum_i \sum_c R(i,c,n)] \times [(\sum_j \sum_d R(j,d,n)) - 1]} \tag{A.6}$$

The numerator counts the number of pairs of different coders $(i,j)$ which have annotated data sample $n$ using the same category $c$. The denominator is a normalizing factor which counts all possible pair of coders. If all data samples have been annotated by all coders, then the denominator simplifies to:

$$D \times N \times (N-1) \tag{A.7}$$

# Appendix B

# Sample Annotation Instructions

The following sections report the full text of the on-line instructions used by *Nb* in the second annotation experiment described in Chapter 5. This experiment produced the most reliable annotations. The text was originally formatted as an hypertext document to be viewed from *Nb* , and it has been edited to fit the printed page. The instructions are complemented by four on-line exercises, which are not reported here.

## B.1 Introduction

You will be annotating some transcriptions of telephone conversations. The goal of this annotation exercise is to provide some data useful for the automatic analysis of these conversations.

The telephone conversations you will be annotating are about movies. Customers have been calling a service in Atlanta called Five One One Movies Now. It is a service similar to Boston's 333-Film, except that you talk to real people to get information. BellSouth Movies agents can list the movies and show times at different theaters, and the theaters' phone numbers.

Here is a sample transcription where the purpose is to list show times (C stands for the customer and A stands for the agent):

> C: [Ah] Barcelona, playing at The Screening Room, what time does that start?
>
> A: Okay, hold on please...
>
> C: You like Howdy.
>
> A: Okay, sir?
>
> C: Yes.
>
> A: It's playing at three, five fifteen, seven thirty, and nine forty.
>
> C: seven thirty and nine forty. Thanks.
>
> A: You're welcome.

Note how the conversation is casual, and includes background talk such as "Okay, hold on please... You like Howdy. Okay, sir? Yes." (Howdy does not have anything to do with movies).

## B.2 Example Annotated Dialogue

You will annotate these transcriptions by breaking them down into paragraph-like sections. Each section has to be labeled with a specific purpose. The purpose name must specify what is the information that the customer gets from the agent. Here is a sample annotated dialogue, in which each purpose is annotated by a different color:

| |
|---|
| 1. A: Movies Now, this is Marguerite |
| 2. A: Can I help you? |
| **List Movies At Theater** |
| 3. C: Yes. I'd like to check the movies at Akers Mill. |
| 4. A: It's A I K E R? |
| 5. C: No. A K. |
| 6. A: Akers Mill is showing Jason's Lyric, Love Affair, |
| the Puppet Masters, and Road to Wellville. |
| **List Theater Playing Movie** |
| 7. C: Oh, no Pulp Fiction? |
| 8. C: [huh] |
| 9. A: No, ma'am. |
| **List Show Times For Movie** |
| 10. C: [uh] Well, the Galleria's right next to that. |
| 11. C: Check the Galleria. |
| 12. A: Galleria Eight? |
| 13. C: Yes. |
| 14. A: OK. |
| 15. A: The next show times there are seven thirty and ten fifty. |
| 16. C: Thank you. |

Let's review the sequence of annotated segment purposes:

- Lines 3-6: List Movies At Theater. The agent lists the movies at Akers Mill.
- Lines 7-9: List Theater Playing Movie The customer wonders whether Pulp Fiction is showing at Akers Mill. It is not showing there.
- Lines 10-16: List Show Times At Theater. The customer wonders whether Pulp Fiction is showing at Galleria Eight. It is showing there, and the agent reports the next show times to the customer. Note that this last purpose is not tagged as List Theater Playing Movie, but rather as: List Show Times At Theater. The customer is indeed trying to know whether the movie is playing at this theater, and the agent answers by listing the show times, because he figured that's what the customer wants to know. By convention, we annotate a segment based on what the agent's reported information is about, independently of the customer's request.

## B.3 Segment Purposes

We're only interested in segmenting the dialogue according to the new information that the agent gives to the customer. The segmentation should provide a basic high-level outline of what information the customer gets from the agent.

For example, we can outline a typical conversation like this:

1. The customer asks to locate a theater where a specific movie is playing
2. The agent reports the next few show times.

Another typical segmentation could be like this:

1. The customer asks what's playing in a part of town, and the agent listing a few theater names.
2. The customer asking what's playing at a specific theater. the agent lists all the movies playing at that theater.
3. The customer selects one of the movies, and the agent finally reports the next few show times.

We have found that this list of five basic purposes is appropriate to segment a conversation into two or more sections:

- List Movies Playing At Theater
  This purpose usually starts with the customer asking: "what movies are playing at this theater?", "how about this theater?" or "what else is playing" When this purpose is started by the agent, the agent usually asks: "Would you also like to know what's playing there?"
- List Theater Playing Movie
  Starts with the customer asking "where is this movie playing", "how about this movie" or "is it playing anywhere else", or just mention a movie, as in: "I am looking for Pulp Fiction" or just mentioning the movie name. It can also be started by the customer just mentioning a location, such as "Marietta" (a part of Atlanta), meaning: "Is this movie playing anywhere in Marietta?" If started by the agent, this purpose usually starts with "would you like to know where is it playing?". Sometimes, the agent reports this information without being asked literally by the customer "where is it playing".
- List Show Times At Theater
  Starts by the customer asking to list one or more show times or the agent asking: "would you like to know the show times for this movie" Sometimes, this purpose starts with the customer asking whether a movie is playing at a certain theater, without explicitly asking for a show time, and the agent reports the show times, as seen in Step-1.
- Specify Theater Location
  The purpose of this segment is to identify the location or address of one or more theater. It usually starts with a request of type: What are the theaters located in this part of town? What part of town is the theater located in? What is the name of the theater? Where is the theater? What landmark building is next to this theater? What are the directions for this theater?

- List Phone Number For Theater

  Usually starts by the customer asking for the phone number for a theater, or with the agent asking: would you like to know the phone number?

When tagging segment purposes, you should focus on segmenting the information reported by the agent, and on including the request that triggered it. The purpose of each segment is for the agent to deliver some new information to the customer. This means that each segmented purpose should have one or more lines in which the agent tells something new to the customer, such as where a movie is playing or what are the show times. The segment is usually started by a request for information from the customer or the agent prompting the customer to request some information.

A segment can always be summarized in theory by a customer question followed by an agent's answer, while it can be realized in practice by many different conversation turns. For example, the following casual conversation segment:

| |
|---|
| C: How about the Brattle |
| C: Are there any othe show there? |
| A: At the Brattle? |
| C: Yes. |
| A: Hold on. |
| A: Ok. I have it listed at the Brattle |
| A: It's showing there at nine fifteen. |
| C: nine fifteen? |
| A: Yes. |

This whole segment can be thought of as a real conversation segment that is one possible realization of the following question-answer pair:

| |
|---|
| C: Is there another show at the Brattle for this movie? |
| A: Yes, it's showing there at nine fifteen. |

## B.4 Segment Initiatives

In the segment initial sentence, the customer request may not be in a form of a well-formed question, and sometimes it is the agent who starts the segment, even if it is the customer that wants the information.

A purpose often opens with a request for new information from the customer. This information is something that the customer would like to know from the agent. It can be a direct request, such as:

| |
|---|
| C: Where is it playing? |
| C: When does that start? |
| C: What is the phone number? |
| C: Is it playing at the Kendall Square Cinema? |

Or it could be an indirect request, or a partial sentence that can be understood as a request for information from its surrounding context:

| |
|---|
| C: how about the Kendall Square cinema. |
| C: In Cambridge, the Kendall Square cinema. |
| C: What about Stargate? |
| C: So, no Pulp Fiction? |
| C: nothing after that? |
| C: Check the Kendall Square cinema. |

A segment can also be opened by the agent asking a precise question to the customer, to get the customer tell the agent what information they are looking for:

| |
|---|
| A: Would you like to know the show times? |
| A: Any particular movie? |
| A: Any particular movie that you wanted to see? |
| A: Which one would you like the show times for? |

In deciding when a segment should start, look for the first line that is understandable and first introduces the segment purpose. However, Sometimes it is hard to decide which line exactly starts a new purpose. For example, let's take another look at this section, in particular lines 10, 11, 12 and 13:

| |
|---|
| 7. C: Oh, no Pulp Fiction? |
| 8. C: [huh] |
| 9. A: No, ma'am. |
| 10. C: Ok. |
| 11. C: What about [uh] — unclear |
| 12. C: [uh] Well, the Galleria's right next to that. —- new theater |
| 13. C: Check the Galleria. |
| 14. A: Galleria Eight? |
| 15. C: Yes. |
| 16. A: OK. |
| 17. A: Pulp Fiction is there. |

Line 11 is an incomplete, unclear sentence, and it is assigned to the first segment.

Line 12 introduces a new theater name: Galleria. At this point, the customer is having an idea, almost thinking aloud. The agent by now knows what the customer wants.

In Line 13 the customer formulates a more direct request to check the Galleria movie listing.

The convention that we ask you to follow in this case is that if the initial line (e.g. line 12) has a complete noun phrase that clearly introduces a new segment purpose that the agent can understand, then you should start a new segment at that sentence.

In this other example below, the question is "spread" over two lines (20 and 22), but can be understood by the first line (i.e. line 20.):

| |
|---|
| 19. A: It's not showing there. |
| 20. C: What about the Brattle? – new segment start here! |
| 21. A: What? |
| 22. C: Is it showing at the Brattle? |

Note that a segment can start also by the AGENT making a precise question to the CUSTOMER. By precise I mean the question has to be about a movie, a show time or a theater. For example, the following two segments are started by the agent asking a question to the customer:

| |
|---|
| A: Any particular movie? — segment can start here! |
| C: Yes, Stargate. |
| A: Ok. Stargate. I have it listed at the Brattle. |
| A: Would you like to know the show times? |
| C: Yes. Give me the show times after ten P.M. |
| A: Ok. Tonight the last show is at midnight. |

## B.5 Sentences that should not start a segment

The initial few sentences of the dialogue usually contain greetings, polite forms and other generic requests, where the agent does not give any specific information in one of our five core segment purposes, and the customer does not make any specific request. As a consequence, these sentences should not be tagged. In the example below, we start tagging at line 4, and we ignore lines 1-3:

| |
|---|
| 1. A: Hello, this is Movies Now. How can I help you? |
| 2. C: Is this the number where you get all the different theaters? |
| 3. A: Yes. |
| 4. C: Ok. I was looking for Stargate. |

In general, phrases that are demanded by etiquette or communication needs and are not strictly necessary to the completion of the purposes should not start a new segment. "Feedback" talk and non-speech events following a request or a response should always be included in the current segment and should NOT start a new segment. Background talk typically include confirmations:

| |
|---|
| A: The show is at seven thirty and nine thirty. |
| C: Seven thirty and nine thirty. |

Acknowledgments, thank you, and goodbyes also continue existing segment:

| |
|---|
| A: The show is at seven thirty and nine thirty. |
| C: Ok, Thanks. - thanks |
| A: You're welcome. - polite form |
| C: Bye. - goodbye |

Non-speech events continue the existing segment:

| |
|---|
| A: The show is at seven thirty and nine thirty. |
| C: [paper rustling] - non-speech event |
| C: [cough] Ok [laughter] - non-speech event |

Out-of-domain sentences that are not about the movies should also just continue the existing segment:

| |
|---|
| A: The show is at seven thirty and nine thirty. |
| C: Hey, turn that radio off, I'm on the phone. — side conversation |
| C: What? Seven thirty? |
| A: Yes, and nine thirty. |
| C: Hey, this guy has all the show times, shut up! — side conversation |

Stuttering, incomplete sentences, false starts, speech errors and the like always continue the current segment. We use the convention to start a new segment only when the sentence that starts the task can be clearly understood.

| |
|---|
| A: And the next show is at four thirty and nine thirty. |
| C: what, what else – incomplete sentence, stutter |
| C: I mean, sorry [cough] – unclear |
| C: what time – incomplete, unclear |
| C: I mean, what else is playing there? – the new segment starts HERE |

## B.6 Clarification Sub-dialogues

Because we are interested only in information given by the agent to the customer, confirmation sub-dialogues started by the agent always continue the current segment and should NOT open a separate segment. Typically, these sub-dialogues are about specifying what area of town the customer is in.

| |
|---|
| C: Can you list me what's playing? |
| A: What part of town? (clarification sub-dialogue) |
| C: Kendall Square. |
| A: OK. Kendall Square in Cambridge. |
| C: Yes. |

Note that refining a segment purpose is not the same as switching to another purpose. For example, consider the following section:

| |
|---|
| 1. C: What time is Stargate? |
| 2. A: It's playing at two and four thirty. |
| 3. C: Any other show times later in the evening? |
| 4. A: Yes. Nine thirty and midnight. |
| 5. C: Ok. |

In this case, line 3 (C: Any show times later in the evening?) starts a new purpose, for three main reasons:

1. Line 3 is not a refinement of the preceding purpose in line 1-2. From the point of view of the agent, the purpose in line 1-2 is effectively completed when the agent has given the information that the movie is playing at two and four thirty.
2. Line 3 is about some new and different information that the customer wants from the agent, and not vice versa.
3. The purpose in lines 1-2 is completed. The agent has provided the information requested, and now the customer wants something more and different.

The final section of a dialogue usually includes a few lines of thank you, welcome, and goodbye. Because it is hard to distinguish a feedback line from a closing line, we ask you not to put the last few lines in a separate segment. Instead, we ask you to just continue the last segment until the end of the dialogue. So, for example, a typical final segment would include all of the following lines:

| |
|---|
| A: It's only playing at the Movies at Gwinnett,<br>Peachtree Corners, and Roswell Mall.<br>C: That's it?<br>A: That's it.<br>C: OK, thank you. |
| C: OK, and then go ahead and give me the show times at Gwinnett.<br>A: Gwinnett would be next show times four fifty,<br>A: seven fifteen, nine forty-five.<br>C: Thank you.<br>A: Thank you for calling Movies Now.<br>C: Bye.<br>A: Bye-bye. |

## B.7 Alternating Segment Purposes

Sometimes, a dialogue does not follow a linear thread such as: *Question.1 - Answer.1, Question.2 - Answer.2.* Instead, a purpose can be interrupted by another purpose, and then restarted later or even abandoned. You should carefully tag these purpose switches. You should not switch to another segment unless the current purpose is either completed or clearly interrupted and the next sentence starts another purpose that can be clearly labeled with one of the labels from the list used in the Tag menu. Sometimes a customer request looks like it's opening two purposes at the same time. For example, the sentence: *C: Can you give the show times at the Brattle, or the phone number.* can open two purposes: Show Times At Theater and Phone Number For Theater. In such cases, take a look at what is the answer that follows from the agent, and name the initial purpose based on what is the first purpose completed by the agent after the request. For example:

1. C: Can you give the show times at the Brattle, or the phone number.
2. A: First, the phone number is: 498-5400.
3. C: 498-5400.
4. A: Yes.
5. A: And today it's playing Casablanca at five and nine o' clock.
6. C: Thank you.

Note that although lines 1 to 6 as a whole answer the double request of the customer, we use the convention of breaking the section into two segments: the first segment includes the initial question and the first answer, and the second segment includes the second answer.

Sometimes a sentence starts a segment that is never completed, perhaps because the customer switches to another segment without waiting for the answer. Such incomplete segments should still be tagged. Consider the following example:

1. C: What's playing today at the Brattle?
2. A: Ok.
3. A: At the Brattle, today I have listed

4. C: Actually, can you give me the phone number, because
5. C: I'm going with someone else, and I don't know what we want to see.

## B.8 Summary

If you are tempted to start a new segment, ask yourself these three key questions:

1. Is the segment purpose a refinement of the current on-going purpose, that does not change the content and the goals of the task, and appears to be a step towards completing the current purpose? if you answer yes, then you should not start a new segment there.
2. Is this new purpose about information that the agent wants from the customer in order to complete the current purpose? Again, if you answer Yes to this question, you should not start a new segment.
3. Is the current purpose completed? That is, has the agent provided the information that the customer wants? Is the current task abandoned or interrupted by this line? If you answer No, you should not start a new segment.

# Bibliography


[1] J. Allen. *Natural Language Understanding (Second Edition).* Benjamin Cummings, Redwood City, CA, 1994.

[2] J Allen, K. Lenhart, and K. Schubert. The TRAINS project. Technical Report 382, Dep. of Computer Science. University of Rochester, May 1991.

[3] A.H. Anderson et al. The HCRC map task corpus. *Language and Speech*, 34(4):351–366, 1992.

[4] F. Andry. Static and dynamic predictions: A method to improve speech understanding in cooperative dialogues. In *Proc. Int. Conf. on Spoken Language Processing*, pages 639–642, Banff, Canada, 1992.

[5] C. Aone and S. Bennett. Evaluating annotated and manual acquisition of anaphora resolution strategies. In *Proc. of the 33rd Annual Meeting of the Association for Computational Linguistics*, 122-129, 1995. ACL.

[6] R. Bakeman and J.M. Gottman. *Observing Interaction: an Introduction to Sequential Analysis.* Cambridge University Press, 1986.

[7] N. O. Bernsen, L. Dybkjaer, and H. Dybkjaer. Cooperativity in human-machine and human-human spoken dialogue. *Discourse Processes*, 21(2):213–236, 1996.

[8] E. Bilange. A task independent oral dialogue model. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics*, Berlin, Germany, 1991.

[9] E. Black. Parsing English by computer: the state of the art. In *Proc. ISSD-93 International symposium on spoken dialogue*, pages 77–81, Tokyo, 1993.

[10] E. Black et al. A procedure for quantitatively comparing the syntatic coverage of English grammars. In *Proc. Speech and Natural Language Workshop 1991*, pages 306–311, San Mateo, CA, 1991. Morgan Kauffmann.

[11] S.E. Brennan. The grounding problem in conversation with and through computers. In *Social and cognitive psychological approaches to interpersonal communication*, Mahwah, NJ, 1997. Lawrence Erlbaum.

[12] S.E. Brennan and E. Hulteen. Interaction and feedback in a spoken language system: A theoretical framework. *Knowledge-Based Systems*, 8:143–151, 1995.

[13] G. Brown and G. Yule. *Discourse Analysis.* Cambridge University Press, Cambridge, 1983.

[14] J. Carletta. Assessing agreement on classification tasks: the kappa statistics. *Computational Linguistics*, 22(2):249–254, 1996.

[15] J. Carletta, N. Dahlback, N. Reithinger, and M. Walker. Standards for dialogue coding in natural language processing. Report on the Dagstuhl-Seminar 167, DFKI, Dagstuhl Castle, Germany, 1997.

[16] J. Carletta et al. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23(1):13–31, 1997.

[17] L. Carlson. *Dialogue games: an approach to discourse analysis.* D. Reidel Pub. Co., Dordrecht, Holland, 1983.

[18] R. Carlson and S. Hunnicutt. Generic and domain-specific aspects of the waxholm NLP and dialog modules. In *Proceedings of ICSLP-96, 4th International Conference on Spoken Language Processing*, pages 677–680, Philadelphia, 1996.

[19] N.A. Chinchor and B. Sundheim. Message understanding conference MUC tests of discourse processing. In *AAAI 95 Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 21–26, Stanford University, 1995.

[20] H.H. Clark. Managing problems in speaking. *Speech Communication*, 15(3-4):231–242, 1994.

[21] H.H. Clark. *Using Language.* Cambridge University Press, Cambridge, England, 1996.

[22] H.H. Clark and E.F. Schaefer. Contributing to discourse. *Cognitive Science*, 13:259–294, 1989.

[23] H.H. Clark and D. Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22:1–39, 1986.

[24] P.R. Cohen and H.J. Levesque. Preliminaries to a collaborative model of dialogue. *Speech Communication*, 15(3-4):265–274, 1994.

[25] R. Cole et al. Speech as patterns on paper. In *Perception and Production of Fluent Speech*, pages 3–50. Erlbaum, 1980.

[26] S. Condon and C. Cech. Manual for coding decision-making interactions. Technical report, University of Southwestern Louisiana, Discourse Intervention Project, 1992 (revised 1995).

[27] S. Condon and C. Cech. Problems for reliable discourse coding systems. In *AAAI 95 Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 27–33, Stanford University, 1995.

[28] M. Core and J. Allen. Coding dialogs with the damsl annotation scheme. In *AAAI Fall Symposium on Communicative Action in Humans and Machines*, Boston, MA, 1997. American Association of Artificial Intelligence.

[29] N. Dahlback and A. Jonsson. An empirically based computationally tractable dialogue model. In *Proceedings of Fourteenth Annual Meeting of The Cognitive Science Society*, pages 785–790, Bloomington, Indiana, 1992.

[30] M. Danieli. On the use of expectations for detecting and repairing human-machine miscommunication. In *Detecting, Repairing and Preventing Human-Machine Miscommunication, Notes from the AAAI-96 Workshop*, Portland, Oregon, 1996. American Association of Artificial Intelligence.

[31] G. Flammia and V. Zue. Empirical evaluation of human performance and agreement in parsing discourse constituents in spoken dialogue. In *Proc. Euroospeech-95*, volume 3, pages 1965–1968, Madrid, Spain, September 1995.

[32] G. Flammia and V. Zue. Nb: a graphical user interface for annotating spoken dialogue. In *AAAI 95 Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 40–46, Stanford University, March 1995.

[33] G. Flammia and V. Zue. Learning the structure of mixed-initiative dialogues from a corpus of annotated conversations. In *Proc. Euroospeech-97*, volume 4, pages 1871–1874, Rhodes, Greece, September 1997.

[34] H. Grice. Logic and conversation. In *Syntax and Semantics: Speech Acts*, volume 3. Academic Press, 1975.

[35] B. Grosz. Focusing and description in natural language dialogue. In B. Webber, A. Joshi, and I. Sag, editors, *Elements Of Discourse Understanding*, Cambridge, England, 1981. Cambridge University Press.

[36] B. Grosz and J. Hirshberg. Some intonational characteristics of discourse structure. In *Proc. Int. Conf. on Spoken Language Processing*, pages 429–432, Banff, Canada, 1992.

[37] B. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[38] B. Grosz and C. Sidner. Attentions, intentions and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

[39] B. Grosz and C. Sidner. Plans for discourse. In *Intentions In Communication*, Cambridge, MA, 1990. MIT Press.

[40] B. Hansen, D. Novick, and S. Sutton. Systematic design of spoken prompts. In *Proceedings of CHI'96: Conference on Human Factors in Computing Systems*, pages 157–164, Vancouver, BC, 1996.

[41] M. Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[42] P.A. Heeman. Spoken dialogue understanding and local context. Technical report 523, University of Rochester, Computer Science department, July 1994.

[43] L. Hirschman et al. Multi-site data collection and evaluation in spoken language understanding. In Bates M., editor, *Proc. Human Language Technology Workshop*, pages 19–24, Princeton, March 1993.

[44] L. Hirschman et al. Automating coreference: The role of annotated training data. In *AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, Menlo Park, CA, 1998. American Association of Artificial Intelligence.

[45] J. Hirshberg and C.H. Nakatani. A prosodic analysis of discourse segments in direction-giving monologues. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1996.

[46] J. Hirshberg, C.H. Nakatani, and B. Grosz. Conveying discourse structure through intonation variation. In *Proc. ESCA Workshop on Spoken Dialogues Systems*, pages 189–192, Vigso, Denmark, June 1995.

[47] G. Hirst et al. Repairing conversational misunderstandings and non-understandings. *Speech Communication*, 15(3-4):213–229, 1994.

[48] R. Hopper. *Telephone Conversations.* Indiana University Press, Bloomington, IN, 1992.

[49] E.H. Hovy. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63:341–385, 1992.

[50] S. Jekat et al. Dialogue acts in VERBMOBIL. Technical Report 65, BMBF Verbmobil, April 1995.

[51] D. Jurafsky et al. Automatic detection of discourse structure for speech recognition and understanding. In *IEEE Workshop on Speech Recognition and Understanding*. IEEE, 1997.

[52] R. Kompe. *Prosody in Speech Understanding Systems*. Springer-Verlag: Lecture Notes in Artificial Intelligence, Berlin, 1997.

[53] K. Krippendorff. *Content Analysis: An introduction to its methodology*. Sage Publications, 1980.

[54] L. Lamel, R. Kassel, and S. Seneff. Speech database development: design and analysis of the acoustic-phonetic corpus. In *Proc. DARPA Speech Recognition Workshop*, pages 100–109. Report No. SAIC-86/1546, February 1986.

[55] K.F. Lee and H.W. Hon. Speaker-independent phone recognition using hidden Markov models. *IEEE Trans. ASSP*, 37(11):1641–1648, November 1989.

[56] W.J.M. Levelt. Monitoring and self-repairs in speech. *Cognition*, 14:41–104, 1983.

[57] K. Lochbaum. Using collaborative plans to model the intentional structure of discourse. Phd thesis, Harvard University, 1994.

[58] K. Lochbaum. The use of knowledge preconditions in language processing. In *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*, pages 1260–1266, Montreal, Canada, 1995.

[59] S. Luperfoy et al. Discourse resource initiative. URL, Georgetown University, http://www.georgetown.edu/luperfoy/Discourse-Treebank/dri-home.html, 1996.

[60] W. C. Mann and S. A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[61] D. Marcu. From discourse structures to text summaries. In *he Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 82–88, Madrid, Spain, 1997.

[62] D. Marcu. The rhetorical parsing of natural language texts. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, (ACL'97/EACL'97)*, pages 96–103, Madrid, Spain, 1997.

[63] D. Marcu. The rhetorical parsing, summarization, and generation of natural language texts. Phd Thesis CSRG–371, Computer Systems Research Group, Department of Computer Science, University of Toronto, December 1997.

[64] M. Marcus, S. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[65] J.D. McCawley. *The Syntactic Phenomena of English.* University of Chicago Press, Chicago, 1988.

[66] I.D. Melamed. Manual annotation of translational equivalence: The BLINKER project. Technical Report IRCS 98–07, Department of Computer and Information Sciences, University of Pennsylvania, Philadelphia, PA, Winter 1998.

[67] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48, 1991.

[68] M. G. Moser, J. D. Moore, and E. Glendening. Instructions for coding explanations: Identifying segments, relations and minimal units. Technical Report 96-17, University of Pittsburgh, Department of Computer Science, 1996.

[69] M.G. Moser and J.D. Moore. Towards a synthesis of two accounts of discourse structure. *Computational Linguistics*, 22(3):409–420, 1996.

[70] T. Nagata and T. Morimoto. First steps towards statistical modeling of dialogue to predict the speech act type of the next utterance. *Speech Communication*, 15(3-4):193–203, 1994.

[71] C.H. Nakatani, B. Grosz, D. Ahn, and J. Hirschberg. Instructions for annotating discourses. Technical Report 21, Center for Research in Computing Technology, Harvard University, October 1995.

[72] C.H. Nakatani, J. Hirschberg, and B. Grosz. Discourse structure in spoken language: Studies on speech corpora. In *Working Notes of the AAAI-95 Spring Symposium on Empirical Methods in Discourse Interpretation*, pages 106–112, Menlo Park, CA, 1995. American Association for Artificial Intelligence.

[73] C.H. Nakatani and J. Hirshberg. A speech-first model for repair detection and correction. In *Proc. of the 31st Annual Meeting of the Association for Computational Linguistics.* ACL, 1993.

[74] N.J. Nilsson. *Principles of Artificial Intelligence.* Tioga Publishing Company, Palo Alto, CA, 1980.

[75] D. Norman. *The Invisible Computer.* MIT Press, Cambridge, MA, 1998.

[76] D.R. Olsen. *User Interface Management Systems: Models and Algorithms.* Morgan Kauffman, San Mateo, CA, 1992.

[77] D.R. Olsen, A.F. Monk, and M.B. Curry. Algorithms for automatic dialogue analysis using propositional production systems. *Human-Computer Interaction*, 10:39–78, 1995.

[78] B. Orestrom. *Turn-taking in English Conversations.* Gleerup, Lund, 1983.

[79] S. Oviatt and P.R. Cohen. Discourse structure and performance efficiency in interactive and non-interactive spoken modalities. *Computer Speech and Language*, 5(4):297–326, 1991.

[80] R. Passonneau and D. Litman. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–139, 1997.

[81] R. Pieraccini, E. Levin, and W. Eckert. Amica: the AT&T mixed initiative conversational architecture. In *Proceedings Eurospeech-97 5th International Conference on Speech Communication and Technology*, pages 1875–1879, Rhodes, Greece, 1997. Univerity of Patras.

[82] J. Pierrehumbert and J. Hirshberg. The meaning of intonational contours in the interpretation of discourse. In P.R. Cohen, J. Morgan, and M. Pollack, editors, *Intentions in Communication*, pages 271–312, Cambridge, MA, 1990. MIT Press.

[83] L. Polanyi. The linguistic structure of discourse. Technical Report CSLI–96–200, CSLI, Center for the Study of Language Understanding at Stanford University, Palo Alto, CA, 1996.

[84] J.A. Rotondo. Clustering analysis of subject partitions of text. *Discourse Processes*, 7:69–88, 1984.

[85] H. Sacks, E.A. Schegloff, and G. Jefferson. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50:696–735, 1974.

[86] D. Sadek. Towards a theory of belief reconstruction: Application to communication. *Speech Communication*, 15(3-4):243–250, 1994.

[87] D. Sadek and R. De Mori. Spoken dialogue systems. In *Spoken Dialogue With Computers (Chapter 15)*, NY, 1997. Academic Press.

[88] E.A. Schegloff. Discourse as an interactional achievement: Some uses of uh-huh and other things that come between sentences. In *Text and Talk*, Washington, DC, 1981. Georgetown University Roundtable on Languages and Linguistics, Georgetown University Press.

[89] E.A. Schegloff. Repair after next turn: The last structurally provided defense of intersubjectivity in conversations. *American Journal of Sociology*, 97(5):1295–1345, 1992.

[90] E.A. Schegloff, G. Jefferson, and H. Sacks. The preference for self-correction in the organization of repair in conversation. *Language*, 53:361–382, 1977.

[91] E.A. Schegloff and H. Sacks. Opening up closings. *Semiotica*, 7:289–327, 1973.

[92] J.R. Searle. *Speech Acts: An essay in the philosophy of language*. Cambridge University Press, Cambridge, 1969.

[93] J.R. Searle. A taxonomy of illocutionary acts. In *Expression and Meaning*, pages 1–29, Cambridge University Press, 1979.

[94] S. Seneff, P. Schmid, J. Polifroni, J. Glass, T.J. Hazen, C. Pao, and V. Zue. Pegasus: A telephone-access flight status information system. In *ICSLP-98: International Conference on Spoken Language Processing*, Sidney, Australia, Fall 1998.

[95] B. Shneiderman and P. Maes. Direct manipulation vs. interface agents. *ACM Interactions*, 4(6):42–61, 1997.

[96] K. Silverman et al. ToBI: A standard for labeling english prosody. In *Proc. Int. Conf. on Spoken Language Processing*, pages 867–870, Banff, Canada, 1992.

[97] J. Sinclair and M. Coulthard. Towards an analysis of discourse. In M. Coulthard, editor, *Advances in Spoken Discourse Analysis*, pages 1–34, 1992.

[98] R. Smith and S. Gordon. Effects of variable initiative on linguistic behavior in human-computer spoken natural language dialogue. *Computational Linguistics*, 23(1):141–168, 1997.

[99] J. Svartvick and R. Quirk. *A Corpus of English Conversations*. Gleerup, Lund, 1980.

[100] M. Swerts and M. Ostendorf. Prosodic and lexical indications of discourse structure in human–machine interactions. *Speech Communication*, 22(1):25–41, 1997.

[101] H. Thompson and D. McKelvie. A software architecture for simple, efficient sgml applications: The lt nsl software library. In *Proceedings SGML-96 European Conference on the Standard Generalized Markup Language*, Munich, 1996.

[102] D. Traum. Coding schemes for spoken dialogue structure. Unpublished manuscript, Universite de Geneve, January 1996.

[103] D.R. Traum and A.B. Hinkelman. Conversation acts in task-oriented spoken dialogue. *Computational Intelligence*, 8(3), 1992.

[104] J.S. Uebersax. A generalized kappa coefficient. *Educational and Psychological Measurement*, 42:181–183, 1982.

[105] D.E. Walker and B. Grosz. *Understanding Spoken Language*. North-Holland, New York, 1978.

[106] M. Walker. Limited attention and discourse structure. *Computational Linguistics*, 22(2):255–264, 1996.

[107] M. Walker and J.D. Moore. Empirical studies in discourse. *Computational Linguistics*, 23(1):1–12, 1997.

[108] T. Winograd and F. Flores. *Understanding Computers and Cognition: A New Fundation For Design.* Ablex Publishing Co, Northwood, NJ, 1986.

[109] P.H. Winston. *Artificial Intelligence (Third Edition).* Addison Wesley, Reading, MA, 1992.

[110] N. Yankelovich. How do users know what to say? *ACM Interactions*, 3(6), 1996.

[111] N. Yankelovich. Using natural dialogs as the basis for speech interface design. In Susan Luperfoy, editor, *Automated Spoken Dialog Systems*, Cambridge, MA, 1998. MIT Press.

[112] V. Zue. Conversational interfaces: advances and challenges. In *Proc. Euroospeech-97*, pages KN–9–KN–18. Rhodes, Greece, September 1997.