Intentional Influences on Object Redescriptions in Dialogue: Evidence from an Empirical Study

by

Pamela W. Jordan

B.S., University of Virginia, 1981

M.S., George Mason University, 1991

M.S., Carnegie Mellon University, 1994

Submitted to the Graduate Faculty of
Arts and Sciences in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

University of Pittsburgh

2000

UNIVERSITY OF PITTSBURGH

FACULTY OF ARTS AND SCIENCES

This dissertation was presented

by

Pamela W. Jordan

It was defended on

Jan 28 2000

and approved by

Prof. Richmond H. Thomason (Chairman)

Prof. Johanna D. Moore

Prof. Martha E. Pollack

Dr. Marilyn A. Walker

Committee Chairperson

Intentional Influences on Object Redescriptions in Dialogue: Evidence from an Empirical Study

Pamela W. Jordan, Ph.D.
University of Pittsburgh, 2000

In an extended discourse, speakers often redescribe objects that were introduced earlier in order to say something more about them. As an object is described, the hearer and speaker create a discourse entity to relate the information about the object in the utterance to the appropriate mental representation of the object [Kar76, Web78, Hei83, Kam93, Pas96]. Because the main goal is to generate a discourse anaphoric expression that will reevoke the appropriate discourse entity, previous work has focused on producing minimally complex expressions that single out the target entity from a contextually determined set of alternative discourse entities [App85c, Kro86, Dal92, HH95, Rei90a, Pas96]. However, since a goal-directed view of sentence generation suggests that speakers can attempt to satisfy multiple goals with each utterance [App85b] and that a single linguistic form can opportunistically contribute to the satisfaction of multiple goals [SW98], this dissertation seeks to empirically identify a broad range of goals that can influence the selection of attributes when generating discourse anaphoric expressions within dialogue.

This dissertation describes a two-part analysis of computer-mediated design dialogues that tests six hypotheses about redescription influences. The hypotheses are based on functions of repetition at the propositional level and observations about task intentions and problem constraint changes that are implicitly communicated by dialogue participants. The first part of the analysis checks for correlations between utterance and dialogue features and the attributes expressed in redescriptions. The second part of the analysis compares redescription strategies that consider only the identification goal with a strategy that incorporates the correlationally supported hypotheses. Parameters direct how supporting calculations for each strategy are made (e.g. how the distractor set is determined). The analyses indicate that five of the six hypotheses are valid and demonstrate that overloading is feasible.

Preface

Many people have contributed to my growth as a computational linguist, as reflected here in my dissertation research, and I wish to acknowledge as many of them as I can.

First, my advisor Rich Thomason pushed me just the right amount and prevented me from running off in too many different directions as interesting issues arose. I especially thank him for being interested in my topic although it wasn't exactly his cup of coffee. Lyn Walker acted as a co-advisor and also helped me stay on target. She was always enthusiastic about this research and provided a great deal of advice and encouragement. Johanna Moore and Martha Pollack always found time to read and respond to my updates and gave me thoughtful and helpful comments and advice. I feel lucky to have had such an involved, supportive committee.

All the people that I've worked and studied with over the years have had a significant impact on how I identify and approach research problems. In addition to my committee members, I give special thanks to Barbara Di Eugenio, Steve Whittacker, Bob Carpenter, Kathy Baker, Reva Freedman, Alex Franz, Carolyn Rosé, Sue Holm, Bonnie Dorr, Jack Benoit and Kate Finn. I appreciated everyone's encouragement and willingness to exchange ideas, respond to my questions, offer advise and collaborate with me as appropriate.

Perhaps behind every completed dissertation there are supportive family and friends. At least in my case I know that is true. I am especially thankful for Brian. I admire his ability to put up with the side effects of my persistent nature, and thank him for listening and comprehending despite the fact I never managed to understand jitter. I appreciate Grady's infectious smile. Because of his special needs and the extra time he needs to do things, Grady is helping me develop new dimensions to my patience and my attention to detail. These evolving, new dimensions have transferred to my research as well. Finally, I thank my parents, my sister Carol, my in-laws, my friend Stefni Agin, and Grady's friends Mellissa Trinh and Megan Beichner for their encouragement and help.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 The Problem

When we express our thoughts in language, we must choose linguistic forms to describe the objects we want to talk about. In work on computer generated natural language, the problem of how to choose such forms is called the problem of *generating referring expressions*. The problem is challenging even when it is confined to the generation of referring expressions in discourses that consist of only one sentence (see [App85b, Rei90b, Dal92, Lup91, Hor97, SW98] *inter alia*). But the problem is compounded by the fact that often we need to produce multiple-utterance contributions, consisting of several sentences. In these cases the same objects may have to be described multiple times; and in natural discourse the later descriptions are usually abbreviated.

The study of how these redescriptions are produced and interpreted is called the theory of *discourse anaphoric expressions*. This theory postulates that when an object or event is successfully described for the first time in a discourse, the hearer and the speaker create a discourse entity to represent that object or event: i.e., they add this discourse entity to the *discourse model* that they maintain.

In this study, we use the account of discourse entities of [Pas96] which is based on [Kar76, Web78, Hei83, Kam93]. A discourse entity is a variable or placeholder that allows us to index the information about an object or event that we extract from utterances to the appropriate mental representation of the object or event. When an object is first described, a discourse entity such as e_i is added to the discourse model. As new utterances are produced, additional discourse entities may be added to the model when new objects are described and new information may get added about e_i whenever it is redescribed. Sometimes the mental model of the object may get modified as new information is added to the discourse model. For example, with "Mary is my friend." we have two discourse entities and two mental models for "Mary" (e_j) and "my friend" (e_k). However, the hearer

may update the mental model of "Mary" with some of the attributes of the model for "my friend" as a result of the utterance. This interpretation means that a discourse entity links a noun phrase to a mental object that can change as the discourse progresses.

Discourse anaphoric expressions are generally thought to be adequate and efficient at re-evoking the appropriate discourse entity. Adequacy and efficiency mean that there is just the right amount of information included in the expression to re-evoke the appropriate discourse entity and to serve the purpose of the reference [Dal89]. Because of this, computational work on generating referring expressions and discourse anaphoric expressions [App85a, Kro86, Dal92, HH95, Loc95, Rei90a, Pas96] has concentrated on how to produce a minimally complex expression that singles out the discourse entity from a contextually determined set of alternative entities. According to these approaches, a descriptor containing information that is not needed to single out the referent would not be minimally complex.

This general approach can be seen as an implementation of the two parts of Grice's *Maxim of Quantity*, according to which an utterance should both say as much as is required, and no more than is required [Gri75]. Although such models of Grice's Quantity Maxim seem plausible from a theoretical standpoint, recent work on naturally occurring speech has produced compelling evidence that they are overly simplistic [Wal93]. Applying the Quantity Maxim with an impoverished interpretation of what is required rules out repetition as a potential way of achieving particular communicative goals. Attributes that are logically unnecessary in a discourse anaphoric expression may be intentional repetitions with a communicative purpose. By ruling out repetition as a means of achieving goals there is a risk of increasing the costs of achieving these goals, or even making them impossible to achieve.

Furthermore, a goal-directed view of sentence generation suggests that speakers can attempt to satisfy multiple goals with each utterance [App85c]. It suggests that this strategy also applies to lower-level forms within the utterance [SW98]. That is, the same form can opportunistically contribute to the satisfaction of multiple goals. This many-one mapping of goals to linguistic forms is more generally referred to as *overloading intentions* and was first coined in [Pol91]. Subsequent work by computational linguists has shown that this overloading can involve tradeoff across linguistic levels. That is, an intention which is achieved by complicating a form at one level may allow the speaker to simplify another level by omitting important information. (E.g. a choice of clausal connectives at the pragmatic level can simplify the syntactic level [DW96], and there are tradeoffs in word choice at the syntax and semantics levels [SW98]).

Although we have learned that overloading is natural and perhaps even necessary, we have no well supported account of what degree of overloading is reasonable and what

forms can more readily address multiple goals in dialogue. Without such an account, we have no principled way to deploy overloading in the automatic generation of natural language. Without well supported constraints on overloading, we are liable to create overloads in unnatural ways which will actually impede effective communication. For instance, we may produce descriptions and utterances that are too densely packed to be readily comprehensible by the hearer.

In current computational approaches, the meaning of "serving the purpose of the reference" in the definition of adequacy is too vague and the problem of resolving this vagueness is largely unaddressed. This creates a further research challenge: although theoretical models of reference have shown the importance of task and discourse constraints in guiding the identification of the referent [AK87, Loc95], nobody has identified what constraints are appropriate. The computational work has only made use of constraints that consider perceptual properties[1], and which only react minimally to task and discourse goals and constraints [App85b, App85c, Dal92, Rei90a, Loc95, Edm93, Pas95]. For example, in these models the task goals lead to a need to describe an entity (e.g. KNOWREF) or some property of an entity (e.g. KNOW location of X), but these goals are not used to help decide which properties will more readily re-evoke the appropriate discourse entity. If other communicative goals can be partially satisfied by a discourse anaphoric expression, then perhaps this can make a particular description preferable over another. With the current approaches, if there is more than one way to minimally re-evoke the entity, the choice between them is arbitrary.

The narrow focus on perceptual constraints creates a further problem. It limits attention to describing objects in the perceptual fields of the participants at the time of the description. However reference is frequently made to spatially or temporally remote objects, or even imagined objects, that exist solely in the speaker and hearer's mental models of the world. Story problems of the kind found in textbooks provide good examples of this phenomenon as shown in Figure 1.1[2]. Here the reader builds up mental models of objects such as locations, temperatures and radiator fluids. It is not necessary that these objects be physically identified in order to talk and reason about them.

Design tasks have a cast of characters that is similar to that of the story problems in that the designers may work with combinations of real-world objects and purely hypothetical objects. Potential objects provide another example, such as the product (e.g., batter) that results from performing contemplated actions (e.g., mixing eggs and flour to create

[1]The perceptual capabilities of the discourse participants are simulated in the computational work on reference.

[2]Taken from http://www.hawaii.edu/suremath/k4_12dir/k4_12menu.html.

Jessica is planning to drive to Sun Valley, Idaho from her home in Portland, Oregon for a ski vacation. She is concerned that her car's cooling system may not have enough antifreeze for the colder temperatures in Sun Valley. Her car has a 20% solution of antifreeze that protects her car's 8 liter system down to 15°F. However, temperatures in Sun Valley can be as low as -17°F. How much of the 20% antifreeze should Jessica drain from her car's cooling system and replace with pure antifreeze to protect her auto's cooling system down to -17°F?

Figure 1.1: An example of a story problem

batter)[WB92]. Considering only perceptual properties provides no guidance on how to deal with these and with similar cases.

1.2 Contributions of the Dissertation

This dissertation seeks to identify a broad range of goals that can influence the selection of anaphoric discourse expressions in dialogue. By doing this, and accounting for the interaction of these goals, we can begin to develop a picture of what makes particular instances of overloading felicitous.

To focus the study and provide explicit empirical content, we study a corpus which exhibits a high degree of non-minimality; the COCONUT corpus [DJTM00]. We assume that the primary communicative goal of discourse anaphoric expressions is to identify the intended referent by re-evoking the intended discourse entity. Any information that is redundant for identification purposes could be present either because it is a cognitive processing artifact [DR95, Pas96, Lev89, Cre96] or because it helps fulfill some other communicative goal besides identification.

We focus first on these non-minimal expressions. By starting with non-minimal expressions, and selecting a corpus in which there is a high degree of non-minimality, we are more likely to have measurable effects from other sources of intentional influence. Since pronouns are by definition always minimal, we only examine discourse anaphoric expressions that are full noun phrases. We call this subset of expressions *object redescriptions*. In fact, we want to distance ourselves from noun phrases as much as possible in this work. This is because the mental objects that the discourse entities point to aren't always expressed in a single place or solely via noun phrases—and discourse entities are the input we use for deciding the content of object descriptions. The description may actually not be fully realized in a noun phrase during sentence planning, as indicated in [SW98]. They argue

that you would probably prefer "Remove the rabbit from the hat" to "Remove the rabbit in the hat from the hat" even when there are multiple rabbits in the scene. Likewise, during interpretation, it seems plausible that we don't just update the mental model on the basis of what was expressed in a noun phrase.

We expect that the degree to which non-minimal expressions occur is dependent upon the task and the communications setting. For example, in the Pear stories corpus [Cha80] only 6% of the redescriptions are non-minimal [Pas96] whereas in the COCONUT corpus we found that 46% are non-minimal and in Cremer's building domain [Cre96], she estimated that 73% of the redescriptions contained redundant information. We also expect the relationship between task goals and the influence they may have on object descriptions to be dependent upon the type of task being discussed. For example, it seems reasonable to expect task goals that depend upon attributes of domain objects (e.g. design) to be more likely to influence object redescriptions than goals that depend more upon action attributes (e.g. planning) unless the domain somehow induces multiple redescriptions of an action during the course of a dialogue. The COCONUT corpus has all of the above characteristics allowing us to successfully study what other types of influence may exist.

Because we have undertaken a corpus study, we can also investigate many hypotheses about identification. We motivate our choices through our empirical investigations. Although we focus initially on a corpus with a high degree of non-minimality, this doesn't prevent us from also being able to study preferences between ways of describing an object that are nearly minimal.

First we investigate whether the influence of other communicative goals guides the selection of one description over another. All the approaches to generating anaphoric expressions depend on property saliency to provide some guidance with respect to choosing between several nearly minimal and equally good descriptions. Most of this work has been concerned only with perceptual properties. Studies of the saliency of perceptual properties can't be applied directly to cases where the discourse participants are dealing with only mental models of objects. We have no guidance on how to rank the saliency of non-perceptual properties relative to perceptual ones. This study doesn't settle the issue, but we do contribute some experimental results, showing what happens to performance when we vary how saliency is determined.

The approaches to generating anaphoric expressions all relate the form and content of the expressions to discourse structure. It is widely believed that the discourse structure or some partitioning of a discourse defines the distractors from which the target object must be singled out ([Hei83, Rei85, PS84, GS86, Kam93] *inter alia*). As [Dal92] points out, there

are many unanswered questions about how the discourse should be partitioned and how anaphoric expressions relate to the partitioning. While we don't settle these questions here, we do contribute an exploration of what happens to descriptions as we vary the distractor set definition among several definitions suggested in the literature.

1.3 Dissertation Overview

In this chapter, we have described some of the problems that arise in current computational approaches for deciding the content of object redescriptions; if they discuss non-minimal descriptions at all, they treat them as rare abnormalities and attempt to explain them all as accidents of nature. We have suggested here that the frequency of non-minimality in some genres is too high for non-minimality to be accidental. In the remainder of this dissertation we explore potential intentional causes of the non-minimality.

Chapter 2 describes the task and the communications setting for the dialogues that make up the COCONUT corpus. We discuss how the type of task (i.e. design vs. planning or scheduling) and the communications setting (i.e. uninterruptible, computer-mediated dialogue vs. face-to-face, spoken dialogue) may affect the content of object redescriptions.

Chapter 3 describes previous work that is related to deciding the content of discourse anaphoric expressions. In particular, we review representative work on planning referring expressions, sentence planning, repetition and risk taking in communication, cognitive models for object identification, memory and language, the relationship between saliency and selecting object perspective, and intentional and informational relations in discourse.

Chapter 4 presents our research approach. To motivate our approach we explore some of the possible reasons and theoretical explanations for including extra information in a description. We review some standard explanations for sources of influence, including the one where identification is the only intentional aspect of the description and everything else is due to the cognitive architecture. Next we analyze the degree of non-minimality in the COCONUT corpus and show that depending upon the relationship one defines between the discourse entities and the discourse structure, the non-minimality ranges from 39%-46%. This degree of non-minimality calls into question the standard identification explanation. Can this much non-minimality truly be accidental? Finally we outline some alternative explanations for this non-minimality; this motivates six hypotheses. We then describe two levels of testing that we use to explore these hypotheses in the remainder of the dissertation; correlations between factors in the corpus and comparisons of parameterizable description selection algorithms.

In Chapter 5 we describe the preparation of the COCONUT corpus to support the correlational study and the comparison of the parameterized algorithms. We use the corpus as both input and output data for testing the description selection algorithms. At a minimum, we extract discourse entities and discourse structure as input data and use the descriptions for the target discourse entities to evaluate the descriptions produced by the algorithms. We end this chapter by reporting the results of the correlational study of the corpus. The study clearly indicates that four of our six hypotheses are valid.

Chapter 6 describes four algorithms we tested that represent some of the standard *identification-only* approaches for selecting the content of object redescriptions. The variation between them provides for the accidental part of the descriptions. This chapter also describes the description selection algorithm that incorporates the hypotheses we wish to test; **intentional influences**. We evaluated this algorithm by comparing it to the performance of the four *identification-only* algorithms. In addition to presenting the description selection algorithms, this chapter also describes a necessary supporting algorithm: determining the distractor set. Since the definition of the distractor set is an unknown, we describe several definitions that we experimented with during the performance evaluation that is described in the next chapter.

Chapter 7, describes our experimental design for the performance evaluation and the results of this evaluation. There are two categories of experiments; those which explore internal settings within each of the algorithms used in the comparison and those which explore which of the final versions of the algorithms performs the best. Some internal parameters are peculiar to one of the four *identification-only* algorithms and some are common to more than one of the algorithms. For example, most of the *identification-only* algorithms require a distractor set as input. What distractor set definition is used is a parameter in each of the appropriate algorithms. This allows us to try a number of definitions and determine which one produces the best performance. Once we find the best settings for all of the parameters of each of the description selection algorithms, we then compare the algorithms to one another. We find that the **intentional influences** algorithm has a trend towards better performance compared to three of the *identification-only* algorithms and performs significantly better in comparison to the fourth *identification-only algorithm*. This part of the study indicates that five of our six hypotheses are valid influences on attribute selection.

In Chapter 8, we analyze the results of our evaluation, discuss the generality of the results and propose some future work.

Chapter 2

The COCONUT Corpus

To better understand attribute selection for redescriptions, we examined a corpus of human-human computer-mediated dialogues. By examining corpora of human language use, we can test our theories and approaches for description selection and see how well they explain or handle what we encounter in practice. An approach is not necessarily bad if it does not completely cover a particular corpus, but poor coverage can be indicative of a non-general theory or approach (i.e. the approach may apply only to a subset of language uses) or of a theory that is too simplistic and only explains a small part of what contributes to description selection.

An approach that does not explain a particular corpus well is not necessarily a bad approach to follow in systems building. For systems building we would like to follow approaches that can adequately deal with a majority of the cases that we expect to encounter. Ideally, we would like to study multiple corpora in order to span a wide range of language uses and test the generality of our theories and approaches. But it is a large undertaking to study even one corpus. Given the limited timeframe and resources for this dissertation work, we studied one corpus and then made predictions on how our results might apply to other language uses.

Our choice of corpora reflects an interest in specific discourse topics and genres. In this study, we are interested in problem solving discussions; we do not expect our findings to generalize to anything outside of this realm (i.e. we exclude, for example, casual dialogues, novels and courtroom interviews). Given our focus on problem solving, we need to consider how this type of task can influence the content of object descriptions and redescriptions. Furthermore, we are interested in dialogues. We do not expect our findings to generalize to other genres, such as written text and formal speeches. Since there are many types of dialogue settings that are derivatives of face-to-face dialogue, we need to consider how the dialogue setting affects the nature of the redescriptions we are studying. By considering the

possibilities for variations within problem solving dialogues, we can make better predictions about how general our findings are for other types of problem solving dialogues.

2.1 Overview of the Task and Communications Setting

For our empirical investigations, we use the COCONUT corpus [DJTM00].[1] This corpus contains 24 computer-mediated design dialogues in which two people collaborate on a simple design task, buying furniture for the living and dining rooms of a house.[2] The information needed to complete the design task is divided between the two designers in such a way that a good design cannot be achieved without collaboration. With this task, the designers typically describe the furniture items that they believe are relevant to the current subtask and design constraints. It should be noted that, in general, design tasks often require the designers to adjust their problem solving constraints in order to arrive at an agreeable solution [LS97, Lyo95].

The COCONUT task is related to those described in [Wal93, WGR93] but differs in the communications setting and the emphasis and complexity of the task.[3] Each of the two designers is given a separate budget and inventory of furniture that lists the quantities, colors, and prices for each item in that inventory.[4] Neither designer knows what is in the other's inventory or the money that the other has. The designers have the same types of knowledge but different instantiations of it. By sharing information about their different instantiations during their conversation, the designers can combine their budgets and can select furniture from each other's inventories. Purchasing decisions are joint: they must be mutually known and approved. The designers are equals in that there is no master-slave or expert-client relationship. The designers have been given the same briefing on the interface tools and the task goals and incentives, and have had no prior contact.

The designers' main goal is to negotiate the purchases; the items of highest priority are a sofa for the living room and a table and four chairs for the dining room. The designers also have specific secondary goals which further complicate the problem solving task. Designers are instructed to try to meet as many of these goals as possible,[5] and are motivated to do so by associating points with satisfied goals. The secondary goals are: 1) Match colors within a room,[6] 2) Buy as much furniture as you can, 3) Spend all your money.

[1] See http://www.isp.pitt.edu/~intgen/coconut.html.

[2] We also collected 12 trial dialogues that are not included in the corpus.

[3] Walker's similar task is performed by two artificial agents whereas our task and that in Whittaker et. al. is performed by two humans. Whittaker et. al.'s dialogues are spoken whereas ours are written.

[4] In Walker's task this information is committed to memory but in our task the participants have this information in written form.

[5] In Whittaker et.al.'s task the incentives and goals are simpler.

[6] Matching colors means that all furniture items selected for a room have the same color attribute value.

The designers are told of the score values associated with achieving particular goals and the score for a solution is a joint score.

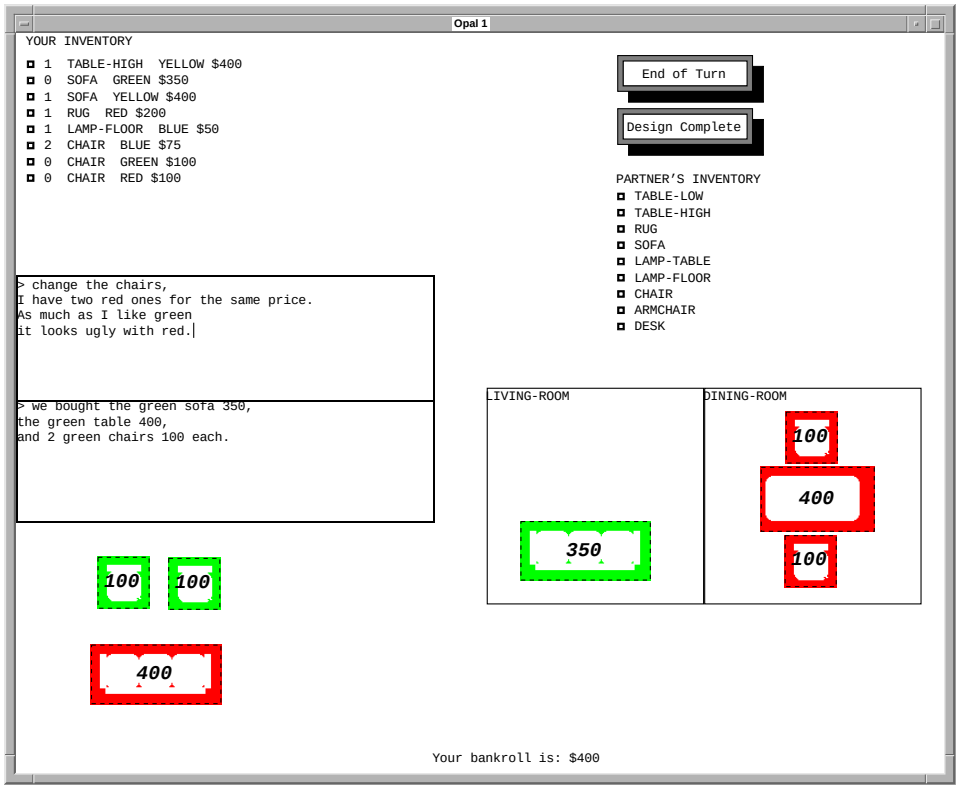

Figure 2.1: A view of the COCONUT interface

The designers are in separate rooms and can communicate via the computer interface only. They are asked to maintain private graphical representations of their discussions and incremental agreements. The designers share dialogue windows but the inventories, budgets and updated floor plans are private and appear only on the owner's color display. Figure 2.1 shows the interface as it looks in the middle of a design session.

The buttons in the upper right corner of Figure 2.1, "End of Turn" and "Design Complete", enforce turn-taking and initiate the incremental recording of the conversation and the graphics updates. No interruption of the partner's turn is allowed. Also note that only the designers' current turns are available, i.e., the turn being currently held in the top dialogue box and the partner's previous turn in the bottom one.

During an incremental recording, the most recently transmitted message is recorded as well as the state of the sender's graphics display. The graphics display record is a description of the furniture icons in the two rooms as well as those that have been created but not assigned to any room. The designers incrementally update the floor plan by placing the furniture icons in meaningful locations. Whenever possible we have used this private

information in our corpus analysis as partial evidence of what the speaker's utterance meant and what the hearer understood. However, the primary purpose of the graphics display is as a memory aid for the designers and is only intended secondarily to help clarify possible sources of misunderstanding during analysis.

Note that since a designer does not know what furniture his partner has available, there is a menu (see the mid-right section of the display in Figure 2.1) that allows a designer to define furniture icons that represent what he understands his partner to have as his partner shares this information with him. There is nothing to prevent the designer from creating an icon for a piece of furniture the partner does not actually have. An icon for a non-existent item could result from either a misunderstanding of his partner's item description or an error in selecting feature values for the item. At minimum the designer must know the type of the furniture item (e.g. chair, table). If the designer does not know or is uncertain about any of the other feature values of the furniture item, he can leave that feature unspecified (i.e. color and purchase price).

The designers first worked through a trial problem to familiarize themselves with the task and the communications setting. During this time they could ask for guidance on using the interface and clarification of the goals and incentives. We do not include the dialogues from the trial problem in the corpus. The designers then solved 1-3 scenarios with varying inventories and budgets. The problem scenarios ranged from ones where items are inexpensive and the budget is relatively large to ones where the items are expensive and the budget relatively small.

2.2 Possible Effects of the Setting on Dialogues and Object Descriptions

While the setting for most design tasks is face-to-face communication, it is important to recognize that communications settings are not static. New ones arise with each new communications technology advance and each new setting (e.g. telephone, email) changes some aspect of natural face-to-face communication [Cla96, OC91]. To understand how to generalize the results of a corpus study, one must consider how the aspect of dialogue being studied might be affected by a face-to-face derivative.[7]

Since we did not also collect COCONUT task dialogues within a face-to-face setting, we cannot directly study how a computer-mediated setting alters the dialogues and

[7]While there is ongoing research into computer-mediated design [LS97, SM98], we don't wish to claim that the COCONUT setting is one that would be useful for a design application. Our motivation was simply to make the job of analysis easier to do.

redescriptions. However, we can make some predictions for a range of expected differences by examining the results of studies that have compared similar settings for other types of tasks. In our case, we consider studies that compare a videoteleconferencing setting to a face-to-face setting. With the videoteleconferencing setting, the ability to interrupt is seriously disrupted and the participants tend to develop a formal method to relinquish the floor to the next speaker to compensate for this difficulty.[8] The COCONUT computer-mediated setting is similar in this regard because we prevented interruptions and provided a formal method for relinquishing the floor. The noted effect of fewer interruptions for the videoteleconferencing setting is that the turns are longer and mutual understanding is harder to achieve [Whi95, OWW93]. So we expect this to hold for the COCONUT dialogues as well.

```
S-1:  [1]: I have $550 to spend.
      [2]: How much do you have?

G-1:  [3]: I have 450 dollars

S-2:  [4]: So we have a total of $900 to spend.
      [5]: The cheapest sofa I have costs $300
      [6]: and it is yellow.
      [7]: How much are your sofas?

G-2:  [8]: My sofa's are more expensive
      [9]: so buy your $300 yellow sofa.
     [10]: Also.... Your $550 + my $450 is $1000

S-3: [11]: You are right,
     [12]: we have $1000 to spend.
     [13]: I will go ahead and buy the $300 yellow sofa.
     [14]: That leaves me with $250 to spend.

G-3: [15]: I have table-high for $200 green.
     [16]: Do you have any cheaper?
```

Figure 2.2: An interruption-free design dialogue with concise descriptions

[8]The difficulty is attributed partly to half-duplex voice transmission and partly to the loss of directionality of voice [Whi95, OWW93].

```
A-1: [1] I have-hello?

B-1: [2] Hello.

A-2: [3] one big piece of 40 that fits in the left window-in the left
room exactly.  It is-the bigger size is 12 x 10.

B-2: [4] 12 x 10.  It is a sort of triangle.

A-3: [5] Yes, a sort of triangle again.

B-3: [6] And er ok, could you describe it?

A-4: [7] And I have one-let me describe the pieces that I have first.

B-4: [8] Have you got a big pieces-piece for the other, for the right
rectangle.
```

Figure 2.3: An interruptible design dialogue with an extended description

By comparing a COCONUT dialogue in Figure 2.2[9] with a spoken dialogue from [WGR93] in Figure 2.3, one sees that the turns are indeed longer. The turns in the COCONUT dialogue are not as fine-grained or interactive[10] because there is no concurrent feedback (e.g. acknowledgements such as backchannels [KGBM77, KW66]). [OC91] shows that when giving instructions for the same task, the utterances are finer-grained in telephone-mediated dialogues than in recorded speech monologues. The telephone-mediated instructions frequently exhibited complete utterances whose purpose was merely to identify an object. Information about actions to be performed on the object would then be conveyed in separate utterances. In the case of the recorded monologues the object identification was combined with the other information.

[OC91] also notes that concurrent feedback is needed for reduced noun phrases to become established for redescriptions. This was supported in studies by [OC91, KW67, CWG86, GA87, IC87]. However there are some task differences between the COCONUT

[9] All the dialogue excerpts included in this dissertation appear as they have been recorded, i.e., typos have not been corrected. However, the turns for COCONUT are presented broken into utterances, according to an algorithm based on the one proposed by [Pas94]. Further details on our turn-breaking algorithm can be found in [DJP98].

[10] [OWW93] notes the relationship between turn length, feedback and interactiveness.

task and that in [OC91] that could also inhibit the use of reduced noun phrases in redescriptions. This will be discussed further in section 2.3.

We note that the full NPs that redescribe objects are not radically different in the spoken vs. computer-mediated settings for the similar tasks in [WGR93] and COCONUT. In the speech setting, we also see instances of non-minimal redescriptions where the attributes selected are not necessarily perceptual properties that would make identification easier (as suggested by the gestalt approach that we will describe in Chapter 3). For example, in Figure 2.4, a piece of value 40 is introduced in [3.A] and is redescribed minimally in [23.A], [24.B], [32.B]. In [41.A], the redescription is not reduced relative to [3.A] and appears to be non-minimal. Since the relative sizes of the two pieces are not mutually known, the information that the piece is big cannot help the hearer pick out the piece. Also, it seems that "the 40", is a stronger candidate for an established reduced NP. Similar redescriptions of the piece of value 35 seem minimal up until [60.A] where it could be redescribed without the value of 35 (i.e. the speaker could go directly to the shape description). However in this case one should question whether the non-minimal redescription is simply the reduced NP effect for redescriptions (i.e. lexical focus as we will describe in Chapter 3).

We may find a higher proportion of full NPs than pronouns being used for subsequent reference in computer-mediated settings than in spoken ones. Although we have not analyzed the spoken dialogue corpus for this, we can informally see from Figure 2.2, that fewer pronouns are used to describe the design task objects than in Figure 2.3 (1:3). This may be due to the need to communicate attributes of the design object that cannot be concisely described with a few words. That is the description must be broken across multiple utterances. In Figure 2.2 the initial description of *the yellow sofa* is covered in 2 utterance units [5] - [6] whereas in Figure 2.3 an initial description of the shape of the design object that is worth 40 points requires 4 turns (utterances [3] - [6]). But compare this spoken description to the one in Figure 2.5 where five simple shapes require just two turns, are much more concise and look more like what we see in the COCONUT dialogues. However, in this dissertation, we will not be examining pronoun usage, so possible differences in this area will not matter to our findings.

This ends our discussion of the longer turns induced by the inability to interrupt. We move now to the second effect noted in the videoteleconferencing setting: that mutual understanding is harder to achieve [Whi95, OWW93]. The effect noted refers to mutual understanding as the interaction progresses and not necessarily to the macro-level of the dialogue and whether the purpose of the dialogue is fulfilled. Studies have shown that lack of feedback about the hearer's understanding leads to longer descriptions [OC91, KW67,

CWG86, GA87, IC87] and longer times for successful completion of tasks even when the word lengths for instructions in the two settings were not significantly different [OC91].

At the macro-level we know that mutual understanding must have been achieved in the COCONUT dialogues since all the designers believed they had reached an agreement and their final floor plans were identical in all but one case. However, in keeping with the above findings when comparing settings, we expect that the designers may have compensated by using more explicit and repetitive descriptions of objects. We expect that having more explicitness and repetition in the descriptions and dialogue would make it harder to prove a correlation between object descriptions and information that is implicitly communicated.

The COCONUT dialogues appear to have more overlapping discourse segments than one might expect for spoken dialogues. This may be due to the longer, less interactive turns. For example, in Figure 2.6 utterance [11] presents an optional item for the living room (an inference supported by the designers' graphics displays) and then goes on to discuss items for the dining room in [12]. Here the current turn holder started the discussion of the new parameter before getting feedback from his collaborator. In a spoken dialogue with no imposed turn-taking mechanism, one would expect to immediately make use of the objects that were just brought into focus. However, the interdependencies of the task can cause overlaps as well. In Figure 2.7 (from the corpus described in [WGR93]), decisions about an item worth 40 points which is first described in [1-7] and an item worth 20 points that is first described in [9] are left pending until utterance [24] for the 40 point item and utterance [77] for the 20 point item. Since this dialogue setting is interruptible speech, one can argue that the task is more influential than the setting in creating overlapping discourse segments.

The designers can sometimes solve the problem in fewer turns by sharing everything that isn't mutually known at the beginning.[11] This means that all the objects are initially described at the beginning of the dialogue as in Figure 2.8. Having all of the objects salient initially should make it harder to uniquely identify objects subsequently without repeating many of the properties. However, since only 29% of the dialogues used this strategy, it shouldn't automatically cause our findings about redescriptions to be specific to design tasks.

[11] 7 of the 24 COCONUT dialogues used this strategy. The design corpus from [WGR93] also has some instances of this type of strategy.

2.3 Possible Effects of a Design Task on Dialogues and Object Descriptions

Most discourse studies in computational linguistics have concentrated on dialogues that involve planning or scheduling tasks. It is not unprecedented to select a simple task taken out of a larger context in order to control the situation and potentially allow for a more objective analysis. In the case of the COCONUT task, we too opted for a simple decontextualized task but we chose to analyze a design task instead. We view the design task as part of a larger product realization problem that also encompasses both planning and scheduling as subtasks (see, for example, [Lyo95]). For product realization, one typically needs to plan and schedule the design subtasks, and the resources and processes needed to manufacture the product. However, the design task itself primarily involves the negotiation of product features and the constraints between dependent design subtasks. The design task is most advantageously represented as a constraint satisfaction problem whereas, this is not usually the case for planning and scheduling tasks.[12]

To better understand what we mean by a design task, consider a group of electrical engineers who have the task of designing a circuit board.[13] They have the required functionality defined at a high-level but have to decide on which off-the-shelf integrated circuits (ICs) to use to help achieve this functionality. They have a variety of ICs to choose from where the choices offer a different but overlapping set of functions and have different costs and different impacts on the overall design of the new circuit board. Their goal is to make the board as cheaply as possible but to consider future enhancements and all the products that this particular board might be part of. Their choices may prove beneficial for some products and detrimental for others. Because of this, the team of electrical engineers need to communicate with one another to negotiate the options at the goals level. Since it might not be possible to meet all of the goals with a single design, they may have to give up on trying to meet some of the constraints. All of the designers involved may know of some ICs that are available but since there is a vast IC market, it is reasonable to assume that there will be some possibilities that are not mutually known among the team.

[12]Note that while it has been hypothesized that planning problems [Jos96] and scheduling problems [Qu97, WLKA97] can be translated into constraint satisfaction problems, we think that it is still useful to distinguish planning, scheduling and designing and that it may be more helpful to represent these tasks with specialized languages. However it seems intuitively clear that the existing planning languages offer few advantages for most design problems. Design problems are characterized by complex interdependencies and a sparse space of domain action types.

[13]For other design scenarios, see electromechanical [Lyo95] and architectural [Lot96, LS97] design task descriptions.

With design tasks, when the goals are mutually known but not necessarily all achievable, the focus may include negotiating what the goals should be. If the values for parameters are evenly distributed then negotiation will also emphasize assigning values to parameters.

When the primary goals are harder to achieve, it can lead to either backtracking to find a better solution, or goal changes or both.[14] We expect backtracking to result in more instances of redescriptions. The changes made during backtracking are valuable to us as well since we can study how these changes are communicated and negotiated.

Nothing intrinsic to this design task should result in unusual object redescriptions. It is reasonable to assume that design tasks, and the COCONUT task in particular, should not affect the number of redescriptions. While we will claim that the specific task does lead to the inclusion of identificationally unneeded properties in redescriptions, we expect that this should hold for a wide range of tasks in cases where many object properties are relevant to the problem solving task and where the definition of success is also negotiable. In the case of design tasks, we expect the attributes of domain objects to be more influential in constraint equation writing than with planning tasks. With planning tasks, we might expect the attributes of actions to be more influential so that we would expect to see more identificationally redundant properties in action redescriptions than in object redescriptions.

The size of our design task is smaller than what is typical of "real world" design problems,[15] but the size of the problem should not alter the nature of the object redescriptions. We simply expect to see fewer instances of object redescriptions than when solving a real design problem.

It is possible that the number of similar domain objects could alter the content of the object redescriptions since the larger the number of similar objects, the larger the number of possible distractors. The number of properties needed to identify an object should increase with an increase in the similarities between the distractors and the target object.

As we mentioned earlier in section 2.2, there are many studies that show that concurrent feedback is needed in order for reduced noun phrases to become established for use in redescriptions. One such study is reported in [OC91]. The implication is that lack of concurrent feedback will make reduced noun phrases less likely to appear as redescriptions. We suggest that features of the task might also limit the use of redescriptions since there are some task differences between the COCONUT task and that in [OC91] that could influence

[14] In this overview of the dialogues, our goal is to give the readers a more general impression of the corpus. None of the characterizations in this section have been empirically validated unless otherwise indicated.

[15] We have 6 basic constraints and 21 parameters whereas real design tasks are much larger. For example, the construction domain example [LS97] notes that it had 42 constraints and 54 parameters.

the nature of redescriptions. In [OC91], the water-pump parts for the assembly task are distinct from one another and the novice does not know which description goes with which part. In the case of the COCONUT task, there are no objects to physically identify and both participants know that each design object can be described by a limited set of attributes. Since all the design objects have the same set of attributes, this may make it more difficult to establish reduced noun phrases for the objects than when the domain objects are distinct from one another. Also, when we are making choices about objects and features of those objects, as with design tasks, we expect reduced noun phrases to be hard to establish as well.

3.A <one big piece of 40> that fits in the left window-in the left room exactly. It is-the bigger size is 12 x 10.

4-22 [utterances omitted that continue description of shape and other partial descriptions]

23.A yes, I have-let me describe what I have-all the pieces I have. I have two pieces of one of 20 and one of 25, that both fit in th right-hand er room; and I have <this other piece of 40> and <a piece of 35>, that I think fit together in room1, the left-hand room.

24.B Fit together, <40> -

25.A yes -

26.B and 35. -

27-31 [utterances omitted that discuss alternatives that B has]

32.B So if you have er <a 40> and <a 35> -

33-39 [utterances omitted that wrap up and start discussion of the other room]

40.B Ok, describe me <the 2 pieces for the part on the left>.

41.A Ok, so, <the big one of 40> er takes up the whole left side of 12.

xx.B Yes.

xx.A: And then at the top you go to the-<it> sticks out 2.

42-58 [utterances omitted that give directions for drawing and orienting the piece]

59.B The next one.

60.A The next one is <the 35 one>.

Figure 2.4: An interruptible design dialogue with identificational redundant attributes

```
B-1: [60] Have you got something linear?

A-1: [61] Only three chairs of 1, value 1.

B-2: [62] I have two chairs of 2, which exactly fit there.

A-2: [63] Ok, you can put them in there.
```

Figure 2.5: An interruptible design dialogue with a concise description

```
M-1: [7]-[9]
     [10] I do not have a sofa for a better price
     [11] but, i do have a lamp-floor, blue (250).
     [12] i have a green table (200) and four chairs for (75)
          a piece.
     [13]-[15]

D-2: [16]-[18]
     [19] the lamp and table sound good,
     [20] but the chairs seem expensive.
```

Figure 2.6: Overlapping discourse segments in interruption-free dialogue

1-7 [utterances omitted that describe an item worth 40 points for the left room followed by a request for permission to describe others]

8. B: Have you got a big pieces-piece for the other, for the right rectangle.

9. A: For the right rectangle, 6 x 12. I have the same-that's the size of the room, 6 x 12. I have one piece that is 20 points worth, and that is 6 x 6, the same as you had before in the previous thing.

10-22 [utterances omitted that describe an item worth 20 points]

23.A: yes, I have-let me describe what I have-all the pieces I have. I have two pieces of one of 20 and one of 25, that both fit in the right-hand er room; and I have this other piece of 40 and a piece of 35, that I think fit together in room 1, the left-hand room.

24-65 [utterances omitted that indicate a decision to use items worth 40 and 35, describe items worth 40 and 35 points to orient in room, and that describe fill-in items for the room]

xx.A: Now I have two pieces for this room: one is value 20, and one is value 25.

66-76 [utterances omitted that describe item worth 25]

xx.B: Wow! So I have fit both your pieces, 20 and 25, and another piece of mine which is worth 20.

xx.A: Hm.

xx.B: And, if I am not wrong, I've filled completely the rectangle.

77.A: Ok.

Figure 2.7: Overlapping discourse segments in interruptible dialogue

```
J-1: [1]: have $450.  [2]: 1 table high 400/1 table high green 200 1
          table high red 400/ 1sofa blue 400/1 sofa green 550/1 sofa
          yellow 350 1 rug yellow 150/1 lamp floor blue 250/6 chair
          green 100/ 2 chair red 50

D-1: [3]: i have $550.
     [4]: 1 table high green 400, 1 table high yellow 200/
          1 table hi red 300/1 sofa blue300/ 1 sofa yellow 400/
          1 sofa red 550/ 1 rug blue 250/1 lamp floor yellow 150/
          3 chair green 50/ 5 chair yellow 150/2 chair red 50.
     [5]: I suggest that we buy my blue sofa 300,
          your 1 table high green 200, your 2  chairs red 50,
          my 2 chairs red 50
     [6]: and you can decide the rest.
     [7]: What do you think

J-2: [8]: your 3 chair green my high table green 200 and my 1 chair
          green 100.  your sofa blue 300 rug blue 250.
     [9]: we get 700 point.
     [10]: 200 for sofa in livingroom plus rug 10
     [11]: 20 points for match.
     [12]: 50 points for match in dining room
     [13]: plus 20 for spending all.
     [14]: red chairs plus red table costs 600
     [15]: we only 650 points without rug and bluematch in living room.
     [16]: add it up
     [17]: and tell me what you think

D-2: [18]: Your perfectly right
     [19]: you are so much better than I am at this stuff.
```

Figure 2.8: An "initial dump" strategy design dialogue

Chapter 3

Related Work

In this chapter, we describe previous work that is related to deciding the content of discourse anaphoric expressions. In particular, we will review representative work on planning sentences and referring expressions, cognitive models for object identification, memory and language, the relationship between saliency and selecting object perspective and sources of goals and influences on generation: intentional and informational relations in discourse, and repetition and risk taking in communication.

Recall from Chapter 1, that we limited the problem to non-pronominal sources of discourse anaphoric expressions and this problem is usually addressed as part of the problem of generating referring expressions. A great deal of previous computational work has been devoted to the topic of generating referring expressions and the work generally uses a planning approach so that the intent for the hearer to identify an object naturally arises from higher-level domain and discourse intentions that the generator is planning to try to achieve. However, the planning approach does not extend down to the level of choosing descriptors. At this lower level, specialized routines which are supplied with identification motivated constraints, choose which and how many attributes are needed to uniquely identify the target object.

Next we will consider the cognitive models that have been proposed in the literature for identifying objects. These models appeal to notions of perceptual saliency but tend to overlook the influence of the task and discourse.

Identification motivated approaches for attribute selection incorporate work on the focus of attention and saliency. The expectation is that the discourse structure and the focus of attention should motivate the form (pronominal or not) and content of the discourse anaphoric expressions. The discourse structure provides guidance on what objects in the discourse are salient. These salient objects are the distractors from which the target object must be distinguished.

Once the salient objects are recognized there is also the problem of determining which of an object's attributes are salient. This is called the object perspective. While in the COCONUT domain there is only one object perspective, there can be more than one perspective for more complex object descriptions or more complex domains.

In addition to salient objects and attributes, there is the question of relative attribute saliency for an object. Given that we know which attributes are salient for an object, the theory on selecting attributes assumes that the degree of saliency between these attributes varies and that we would include the more highly salient attributes in favor of the less salient ones. In all of the above cases, it is reasonable to assume that similar mechanisms are at work for determining salient objects and salient attributes for objects.

In the computational work that has been done on modeling saliency in the context of referring expression generation, ease of implementation was the key issue whereas with cognitive models of saliency, explaining human behavior was the key issue. Although we prefer to implement simple models, we also want the models to be close enough to human performance so that they will more generally cover human language. This is because we want the language we generate to be acceptable and understandable by humans. Although the stack model that is typically used is computationally simple, it arguably does not adequately account for what we observe in human language.

The next area that relates to our work on object redescriptions is finding sources of influence besides unique identification. In general, there are two main categories of discourse goals that have been studied, informational goals that relate the content of propositions to one another (e.g. cause-effect) and intentional goals that indicate why the content of a sentence is being presented at this point in the dialogue (e.g. justify the premise, propose a solution, ask a question).

Finally, anything that is not new to a description and is not needed for unique identification can be considered informationally redundant. For this reason we will review the literature on the functions of repetition in dialogue. Although there may be reasons to repeat information, there are trade-offs in doing so. For this reason we will also consider some work on what is at risk by not being explicit.

3.1 Generating Referring Expressions

When interacting with others via language, we arguably appear to do a lot of planning about the form and content of our communication. We must decide what we hope to accomplish by the interaction (e.g. do we hope to persuade someone to do something), what specific content will best support our intentions, how to organize the selected content

so that our intentions are recognized, and finally the linguistic forms that will encode this content. We expect that as we get closer to the actual language we will utter, that our intentions at each stage will decompose into lower level supporting intentions.

There is a large and continually growing body of research in discourse planning for natural language generation ([App85a, MP93, HH95, Edm93, YM94, Loc95, Dal95] *inter alia*). A majority of this work is based on the theory that when a speaker produces an utterance, he has some intentions that the hearer is expected to recognize in order for the utterance to count as understood. The speaker generates a plan to have his intentions recognized and when this plan is executed it results in the production of an utterance. The hearer recognizes the speaker's intentions by constructing a plausible plan to explain the observed utterance. If the intentions are not recognized or cannot be fulfilled because some enabling condition is not satisfied then the speaker and hearer must collaborate on a repair until the intentions are successfully recognized. Cohen's [Coh81] and Appelt's [App85b] work were among the early computational approaches that treated the problem of generating referring expressions as a planning problem. While Cohen planned requests that the hearer identify a referent, Appelt focused on planning concept activations which are equivalent to Searle's propositional acts [App85a].

The typical natural language generation architecture is modularized so that once a decision is made on what content to include and how that content should be structured, there is the more local problem of how to create sentences that express that content. In particular, the content of each sentence must be organized. For example, decisions must be made about pronominalization, content aggregation to remove unnecessary redundancies, and ordering of prepositional phrases and adjectives. One of Appelt's goals in his work on planning referring expressions was to eliminate the distinction between deciding what to say and how to say it [App85a]. He believed that the modularization should be along other dimensions since with a planning view of language, the actions that can contribute to the satisfaction of communicative goals at any level all depend to some degree on the linguistic constraints in the current context [App85a].

Appelt's computational model is based on a theory of speech acts as described in [Sea69]. According to this theory, the speaker performs some utterance acts and the hearer tries to recognize the proposition that the speaker intended to communicate. The speaker communicates the components of the proposition via *propositional* acts of referring and predicating. From this the hearer must infer what the speaker intends and this is the basis of recognizing the *illocutionary* force of the utterance act. But Searle's theory equates

uttering a referring expression to performing one propositional act and uttering a sentence to performing one illocutionary act.

Another of Appelt's goals was to eliminate the one to one mapping between illocutionary acts and sentences and the one to one mapping between propositional acts and noun phrases as imposed by Searle's work. With this decoupling, multiple speech acts could be realized by a single sentence and multiple propositional acts by a single noun phrase. Appelt illustrated two types of action subsumption that allowed three illocutionary acts to be satisfied by one sentence and two propositional acts to be satisfied by one noun phrase.

Appelt's process model expands an intention by one level of abstraction and then critics examine the plan for ways to modify it that would opportunistically use the effects of an action to eliminate some other action. Appelt's levels of plan abstraction placed illocutionary acts at the top and these expanded into surface speech acts at the next level and finally into sentences. To illustrate how the critics operate and how linguistic constraints can interact with higher level intentions, consider Appelt's example of how his model would generate the sentence "Remove the pump with the wrench in the toolbox".

First the model's plan critics combine a request to remove the pump from the platform with an inform of what tool can be used to do so; "Remove the pump with the wrench" vs. "Remove the pump. Use the wrench to loosen the bolt". A plan critic recognizes the possibility of action subsumption at the speech act level with the choice of a particular verb. The critic recognizes this by looking for connections between actions. In this case there is a connection between the removal action and the wrench that is a tool for the removal action. The critic recognizes that it can opportunistically use the optional instrument argument slot for the verb "remove", allowing the request act to subsume the inform act. Later when another goal in the plan is expanded so that the illocutionary act of informing of the location of the wrench arises, a critic recognizes the opportunity to suppress the separate utterance, "The wrench is in the toolbox". It recognizes the connection between the modified request and the inform of the location (again that the wrench is a tool for the removal). In this case it can subsume the location inform by adding a new descriptor to the concept activation for the wrench. In this case, three illocutionary acts are now satisfied by the modified sentence and two concept activations are satisfied by the one noun phrase "the wrench in the toolbox".

What Appelt does not address are the influence multiple goals can have on what attributes of the concept will be used to activate it. Although the planner reasons about whether a description is adequate for activating a concept, the possible descriptions are generated external to the planning process so that goals related to the task and goals related

to communication are not used to generate or show that of all the adequate descriptions some would be preferable because they allow other goals to be subsumed. To generate the possible descriptions, the model first chooses a basic category description and then adds descriptors that are mutually known and that can be realized in a noun phrase. The description generator continues until the concept has been uniquely identified. Although the description generator was created under the assumed need to minimize the number of descriptors and the effort required to generate them, the description can be augmented to help realize additional informing acts as we saw above. However, we can see some of the drawbacks of following this approach by looking again at Appelt's example.

Consider what might happen if there were two wrenches in the speaker and hearer's perceptual space instead of one; a small allen wrench and a big crescent wrench, and both are in the toolbox. Suppose too that there are two types of bolts visible on the pump and it is not obvious on quick inspection which type of bolt attaches the pump to the platform. The system knowledge might encode that the tool to remove the pump is a crescent wrench so the first description for the concept activation might result in "Remove the pump with the crescent wrench". But since the hearer doesn't know the location of the wrench, the system must decide whether to remake the description as "the big wrench in the toolbox" since the size of the wrenches will probably be a more visibly salient attribute than the type of wrench, or whether to add descriptors only to get "the crescent wrench in the toolbox". Next, suppose that the bolts are rusty and as part of the pump removal there is the added condition of needing to put a solvent on the bolts first before trying to loosen them. The system might then prefer to say "To remove the pump you must first put liquid wrench on the hex nuts and then use the wrench in the toolbox" because the hearer can infer from the type of nut that the appropriate wrench is a crescent wrench and this will distinguish it from the allen wrench.

As each new goal is addressed, the description needs to be reconsidered. To react to these additional goals, the critics then seem to require a great deal of knowledge about how to remake descriptions as there is no clear connection between these multiple goals and the descriptions that get generated in order to activate concepts. It seems better to know all of the domain goals for the speaker's turn before making a decision about an adequate description. So it seems that at least the high level decisions about the maximum content of a turn should be made before trying to activate concepts. If not all of the content can be linguistically expressed in one sentence or turn then that unaddressed goal can carry over to the next sentence or turn. By knowing all the goals that might influence the choice of

description, the goals that can be addressed by a description of the concept can help select between all the possible descriptions.

While as of yet there are no complete accounts of how to combine sentence planning and realization so that multiple goals can influence the description selected, [SW98] is a start. Like [App85a], this work combines sentence planning and realization and allows multiple goals to guide the choices made. Instead of having critics modify the plan, each step of the SPUD algorithm adds a lexicalized entry to the representation. The representation consists of a tree that may contain multiple lexical entries and is paired with logical formulas. Given a discourse model, the formulas describe the semantic and pragmatic contributions of the tree. The lexicalized entries are elementary trees that are encoded with Lexicalized Tree-Adjoining Grammar (LTAG). The Tree-Adjoining Grammar provides guidance on sentence construction while allowing the order in which content is added to be flexible. Incorporating this into the SPUD algorithm eliminates the need for heuristic critics that modify the content and surface realization of a sentence as with [App85a]. When the semantic contributions of a tree are determined, they are added to the common ground of the speaker and hearer so that inferences about goals can be made. However, as we pointed out in Chapter 1, what goals one might want to consider (beyond the informational relations considered in [SW98, DW96]) and constraints on overloading goals is an open research topic.

Appelt's work was also limited to generating noun phrases that emanate from actions that have two main defining criteria: (1) identification intention - does the speaker intend that the hearer identify the denotation of the description and (2) shared concept activation - is there mutual knowledge about the description's denotation [App85c]. Later formalizations allow intentions behind a concept activation to influence the choice of description and acknowledge that unique identification is just one of many constraints that could influence what description is selected [Kro86, AK87, Loc93, Loc95]. Although these formalizations allow constraints on what properties are selected to be derived from the domain and discourse context, what constraints apply besides identification is still vague.

Kronfeld [Kro86] indicates the need for a new analysis of the speech act of referring that does not rely solely on identification. He considers the distinction between definite description cases in which identification is required (referential use) and those in which it is not (attributive use) that Donnellan pointed out [Don66]. Kronfeld shows that there are three aspects of Donnellan's distinction and that these correspond to three components of a plan-based model for reference:

1. The object representation: a knowledge base can have several representations that denote the same object in the world or it can have one representation that denotes

different objects in the world. This allows for the possibility that the speaker's world is not an accurate representation of the world.

2. Planning to satisfy intentions to refer: intentions to refer may include, for example, intentions to place constraints on the way the hearer thinks of the referent and intentions to identify the referent.

3. The linguistic realization of the referring expression; the denotation of the expression does not have to coincide with the intended referent. E.g., in the case of "the man in the corner drinking the ginger ale" where the man is actually drinking wine, "The man drinking ginger ale" does not denote the intended referent (the man drinking wine) but the intention to refer may still succeed.

With these three components – the object representation, planning intentions to refer and the realization of a referring expression – Kronfeld points out that the labels attributive or referential are no longer necessary. A further implication of the three components is that indefinite as well as definite expressions can have intentions to refer [AK87].

Lochbaum's formalization for identifying parameters as part of a collaborative planning approach for dialogue incorporates work from [Kro86, AK87] but does not attempt to account for constraints other than identification or the influence of multiple acts on a satisfactory description for a parameter. For an agent to be able to perform an act, the agent must be able to suitably identify the parameters of the act (e.g. in the act "Remove the pump", a parameter of the remove act is the pump). Then as part of identifying the parameters of the act, Lochbaum specifies that there must be a satisfactory description for the parameter under constraints imposed by the type of act and the parameter in question. The constraints are derived via an oracle function that is intended to model the context dependent nature of parameter identification. Lochbaum points out this context dependent nature with the example of describing John's residence for the purposes of sending a letter vs. giving directions so that the hearer can physically identify it (i.e. postal address vs. the visual appearance of the house).

The work of Heeman, Hirst and Edmonds [HH95, Edm93] extends that of [App85c] and is an example of plan based generation and interpretation that specifically addresses the problem of collaborating on the content of referring expressions in dialogue. Although this work allowed intentions to further motivate the content of descriptions, it only examined identification intentions and those that have to do with collaboration as suggested by the work of [CWG86, CS89]. Also there is no general mechanism in this work for allowing multiple goals to influence the selection of attributes.

By including intentions about collaboration, they were able to achieve a collaborative model of referring for tasks where the hearer may not know the object that is to be described so the speaker gives the hearer a plan for identifying the object. For example, this occurs with tasks such as direction giving [Edm93].

Like [App85a], the other, later computational models for generating referring expressions tended to use specialized procedures external to planning to decide what attributes to include in the expressions [Rei90a, Dal92, DR95]. [App85a] pointed out that, based on Grice's maxim of quantity, the planned description should be as simple or efficient as possible. [DR95, Dal92, Rei90a] explored this efficiency question in detail but just as it relates to additions that are not the result of a different goal. The discourse and task goals and the task constraints that could influence the description that gets generated are not reflected in the referring expression algorithm.

[Dal92] focused on describing a tractable algorithm for finding a minimal distinguishing description in a given context [Dal92]. Part of the problem [Dal92] addresses is the mechanism used to generate both initial and subsequent references to individuals, masses and sets. The generation of subsequent references addresses whether to use zero anaphora, pronominal anaphora, one anaphora or full NPs. When the decision is made to use full NPs for subsequent reference, the goal is to find minimal distinguishing descriptions.

Dale models action operators via knowledge base (KB) structures that correspond to underspecified events where the participants in the event and the begin and end states of the event must be identified or instantiated to more fully specify the event. To transform the KB structure, which is appropriate for the planning task, to a semantic structure which is appropriate for the clause generation task, Dale's model replaces the indices of objects with structures that describe the objects. Although information that does not need to be described is omitted, this does not mean that all the semantic information in the semantic structure has to be realized in a clause. Although the mapping from the KB structure to the semantic structure for a clause is one-to-many, Dale's model simply uses the first mapping that works.

To generate a referring expression, first the semantic structure for a clause must be built. For each intended referent, the generator specifies the index of the entity and some optional information on its status (e.g. whether it is obligatory for the case frame of the clause that is currently being constructed). Next the type of description that will be generated is partly determined. If the target object is the current center then the type of the object does not need to be expressed, otherwise if the target object is already present in

the discourse model then a subsequent reference will be built and if it is not then an initial reference will be built.

To build the semantic structure for a subsequent reference, first the structure is marked as a given object (following the terminology of [Pri81]). Next the distractor set for the target object in the current context is isolated. The algorithm then attempts to find a minimal distinguishing description by determining the set of properties L to be used to identify the target. This set of properties is added to the semantic structure for the subsequent reference and finally if the selected description succeeds in distinguishing the target from the distractors, the structure is marked as uniquely identifying the object. Stated more formally:

```
U = distractor set
N = number of distractors in U
<a,v> = property-value pair true of x
n = number of entities in U of which <a,v> is true where m <= n <= N
S = subset of entities in U that are target objects

Properties selected based on discriminatory power F:
F(<a,v>,U) = (N-n)/(N-m)

To find minimal distinguishing description:

Compute F for all <a,v>
Select <a,v> with highest F and add to description in RS
If F=1 return
If F>0 Update U by selecting those entities for which <a,v> is true
Repeat until F=1 or F=0
```

Once the semantic structure for a clause is completed, it is mapped to a surface semantic structure that is motivated by syntactic concerns. The surface semantic structure is much closer to the syntax of the linguistic form that will be produced. To syntactically realize the referring expression, if the target is the center and is syntactically non-obligatory then it will be omitted from the clause. If it is obligatory then it will be realized as a pronoun. Otherwise, the target will be realized as a one-anaphor if possible and if not it will be realized as a full noun phrase.

Reiter was also concerned with finding short distinguishing descriptions but looked in detail at issues of lexical preference and how the choices made can communicate multiple

goals [Rei90a]. He was concerned with finding the appropriate level of specificity for the category of the head noun so that unwanted implicatures do not arise (e.g. "Look at the dog" vs "Look at the pitt bull" where *pitt bull* might give rise to an unintended warning). Although Reiter does consider the mechanism for realizing attributes, he made the simplifying assumption that the goals for realizing the attributes are givens.

Reiter embellishes Dale's greedy approach for finding a distinguishing description. He uses a simple iterative algorithm that evaluates a proposed initial distinguishing description (which can be generated using Dale's greedy approach). This description is then checked for whether an attribute can be removed, or multiple attributes can be replaced by a single attribute or whether a value can be replaced by a lexically-preferred one. If modifications are made, the algorithm iterates, otherwise the current expression is the one that is selected as the description of the target object.

[DR95] combines aspects of [Dal92] and [Rei90a] by proposing an incremental algorithm that they call **IDAS**. This algorithm iterates through a list of ordered attributes and adds the attribute if it rules out any distractors. It terminates either when there are no more distractors to rule out or there are no more attributes to consider. The ordering of the attributes are domain dependent and fixed. **IDAS** does not attempt to find the optimal description but favors descriptions that are nearly minimal. [DR95] argues that this is closer to what people must do since it is computational tractable and since psychological studies show that people often produce non-minimal or overspecified descriptions for the purpose of identifying objects.

3.2 Cognitive Models of Object Identification

Several hypotheses based on cognitive processing limitations have been suggested by psychological and computational linguistics research to explain non-minimality or redundancy in object descriptions. Studies by [Deu76, Man86, Son82, Son84] (as cited by [Lev89]) show that it is easier for listeners to identify an overspecified referent than a minimally specified one. Levelt explains these findings by supposing that listeners tend to create a *gestalt* search template for the target object and that an overspecified template makes it easier to search for the referent [Lev89]. For example, searching a group of visible items for "big black bird" is assumed to be easier than "black one" or "black bird" even if the shorter expression uniquely identifies the referent. This could also explain why there is a preference to include a noun related to the type of the target object in the description even when it is redundant. Levelt also claims that the distractor objects should be irrelevant for whether the gestalt template is used.

Lexical focus [Pas96] provides another possible explanation for some cases of observed redundancy. This explanation is motivated by the observation that speakers have a tendency to repeat the last description for the target object in a redescription. While Passonneau doesn't cite any experimental studies to support this observation, Clark and Wilkes-Gibbs' findings in experiments with a Tangram identification task [CWG86] may provide some evidence. They found that between trials, the length of the descriptions get shorter as the participants establish common ground and come to settle on a particular description for each figure.

Finally, unnecessary or redundant information could merely be a side-effect of processing limitations associated with the communicative goal of identification. In making this suggestion, [DR95] point out that it is computationally intractable and therefore psychologically implausible to attempt to find minimal descriptions. This implies that people may use some combination of heuristic approaches that are along the lines of what we just discussed above. Dale & Reiter [DR95] proposed the computationally tractable algorithm, **IDAS**, that we described in the previous section. It attempts to capture some of the cognitive hypotheses suggested above.

3.3 Saliency

Regardless of whether intentions other than identification influence the selection of attributes in the approaches we described above for generating referring expressions, one thing that is common to all is the influence of object and attribute saliency. In discourse research, the attentional state of a discourse participant is essential for explaining many of the discourse phenomena encountered [GS86] (e.g., the reintroduction of an item that was previously mentioned in the discourse). The attentional state organizes what the agent has been exposed to in the previous discourse so that some elements are more salient than others as a function of discourse structure and recency. First we will review the work on object saliency.

3.3.1 Object Saliency

Anaphoric phenomena fall into 2 broad categories; those that are locally restricted and those that are not. [Dal92] operationalizes this distinction in terms of local and global focus, following [Gro77]. Dale models local focus as a cache memory and assumes it contains the lexical, syntactic and semantic details of current major clauses that are being generated as well as the previous major clause generated. He implements global focus using a number of focus spaces that record the semantic content but not the syntactic or lexical details

of the preceding discourse. The focus spaces are arranged as a stack to model the focus of attention. As focus spaces are removed from the stack they are added to the discourse history.

The content of the focus spaces are determined by the task structure as indicated in Grosz and Sidner's theory of the attentional and intentional structure of discourse [GS86]. A change in task or intention indicates the start of a new focus space. All of the discourse entities or objects described between the start of two topics are considered salient when the focus space is salient. However, as [Dal92] points out, there are many unanswered questions about what the relationship between distractors and salient objects should be. Objects that are in focus may not all be equally distracting in that there may be degrees of saliency between the objects that are in focus.

We are more concerned with global focus as it indicates what objects are currently in focus. Local focus deals with questions such as whether to use a pronoun or a full noun phrase. While one can overspecify a description by using a full noun phrase when a pronoun is allowed, we are more concerned with determining whether a full noun phrase is overspecified with respect to another full noun phrase. [DR95] acknowledge that humans often overspecify noun phrases but they do not consider whether this is intentional or not. Passonneau [Pas95] uses local attentional factors (centering) as a partial explanation for overspecification in the sense that a full noun phrase is used when a pronoun would have been allowed according to centering principles. In order to explain both under- and overspecified NPs in verbal narratives[1], she interleaves centering [GJW83, Kam85] with a search for a distinguishing description [DR95].

3.3.2 Object Representations and Perspective

Once object saliency is determined, we should also consider what perspective or what subset of what we know about the object is most salient. This is called object perspective.

McCoy's dissertation uses object perspective in order to respond to object misconceptions [McC86]. The notion of object perspective is applied to the user model so that it highlights certain aspects of the model based on the discourse context. It acts as a filter on the knowledge representation. Object perspective is likened to point of view where certain characteristics are more important than others. For example the point of view of a building as a home vs. as an architectural work would highlight who lives there vs. who designed it. McCoy's goal was to explain how the highlighting of a set of attributes might occur. Clearly

[1] the narratives are Chafe's Pear stories

a focusing mechanism which highlights attributes mentioned in the preceding discourse is not an adequate mechanism because not all of the attributes associated with a perspective are explicitly mentioned. Earlier, intuitively appealing notions of object perspective were presented whereby the object is viewed as a member of one superordinate out of the many that might define the object ([Gro77] *inter alia*). However this notion of object perspective fails to explain cases where more than one object is discussed during a conversation and there is a shared set of highlighted attributes but no superordinate that is in common, and cases where the object level described varies according to the perspective. McCoy instead makes perspective orthogonal to the object generalization hierarchy and defines a number of perspectives for a domain. These orthogonal domain specific perspectives are small, finite sets of attributes with associated saliency values that dictate which attributes are highlighted. Only one perspective is active at any point in the discourse. In addition, the *importance* of an object is determined by the saliency values projected on its attributes by the object perspective. The whole object becomes highlighted by virtue of the attributes highlighted. Likewise the object level is defined as the one that contributes the most highlighted characteristics. While perspective predicts the attributes highlighted and the level of the object selected, the object and its attributes determine how likely an object will be viewed from a certain perspective.

While McCoy offers a mechanism for realizing a given perspective, how an active perspective is chosen is still on open research issue. However, some of the factors that might influence the choice of perspective include: the user's current goals (e.g. in [MWM85] perspective is indexed by potential goals), and the important attributes and objects that were already mentioned that have a high saliency in the active perspective. Clearly, perspective can be used to help communicate some of these cues just as the cues can be used to communicate perspective. Issues that remain open have to do with the structure of the space of perspectives, transitions between perspectives and its relationship to discourse structure.

The issues in this dissertation are related to perspective in that the idea of salient properties or features is applicable. However what we may be seeing in the cases we are examining are finer grained choices within a perspective (e.g. degrees of saliency where all the salient features are part of the perspective but some of these features are more salient than others as the discourse unfolds). We claim that these finer-grain choices are motivated by domain and discourse goals. This could, for instance, allow one to infer new or modified task constraints based on the fine-grain choices. For example with "I have a $200 red sofa and a $100 blue sofa." and "I have a red rug for $50 and a blue one for $75. Let's get the red sofa and rug.", we can infer that the color match constraint is in effect. With "We only

have $200 left, let's get my $100 blue sofa and your $75 blue rug." we can infer that budget limit and color match constraints are in effect. Compare these with "let's get my red sofa and your blue rug", where we can infer that the color match constraint is not in effect, and with "We only have $200 left, let's get my $100 sofa and your $75 rug." where we can only infer that the budget limit is in effect.

As another example, we will consider the the basic electronics and electricity domain of the BE&E corpus [RDM99]. A student could infer that a polarity constraint is needed on an action when the tutor includes the features red/black for leads or positive/negative for a part of a circuit. Likewise, the student can infer a constraint of a difference in charge when the tutor includes the location feature for the leads. The tutor cues the student about constraints on the actions via the pre-selection of relevant properties or features of the objects discussed.

Note that with COCONUT, we have one perspective that is used throughout the corpus; a neutral physical perspective of the furniture objects. For comparison purposes, some other possible perspectives for furniture items that do not arise within the COCONUT domain are the fashion perspective (e.g. the blue chintz chair or the flowered Victorian chair) and the function perspective (e.g. the low cushy chair or the sturdy wooden chair). To compare to another domain, in the BE&E corpus [RDM99] there is the neutral physical object perspective as well as the function perspective which seems to influence the choice of the head noun more so than the properties expressed (e.g. the source vs the battery).

3.3.3 Attribute Saliency

Once we know which subset of object attributes are salient, which helps define the object perspective, there is the issue of relative saliency between these attributes. Memory architecture research generally models degrees of activation or saliency so that some objects and attributes are more salient than others [And93, JC92, Byr96]. Although [DR95] claim that the planning of referring expressions have to be sensitive to the hearer, they make the simplifying assumption that the more easily perceived properties are more salient and most implementations assume that this ordering remains static for the entire dialogue [Dal92, DR95, Edm94]. While [DR95] point out that the more general solution is to consult a hearer model for guidance on the relative saliency of properties, there has been little research on modeling the hearer's attentional state other than assuming the hearer and speaker have the same attentional capacity. However, research into memory architectures supports the idea that people's attentional capacity can differ because of variations in their memory architecture [Byr96]. However, there is some evidence that in some dialogue situations,

speakers do tend to assume that the hearer has the same attentional state when there is no feedback to the contrary [JW96].

3.3.4 Cognitive Models of Memory and Saliency

Walker extends the notion of attentional state that was described in [GS86] by relating it to psychological work on attention and giving it a cognitively plausible architecture. She puts limits on the attentional state so it is a subset of what the agent has been exposed to in the previous discourse [Wal93]. Walker carries the idea of limited attention further by using it to limit what is available for reasoning as well [Wal96a]. This use of limited attention is also related to work on contextually sensitive reasoning [JWW86, GSGF93, MB95]. However in this latter work the issue of determining the relevant subset of knowledge or doing so in a cognitively plausible way is not the focus.

Memory is generally hypothesized to be made up of long term and short term memory. Working memory is meant to capture the idea that short term memory is a temporary processing area as well as a temporary storage area. Many computational models today treat working memory as an activated portion of long term memory instead of as a separate architectural component. But this alone does not define working memory, since to capture the effects of a processing load on temporary memory, processing rates must play a role in determining what part of long term memory is activated.

One type of cognitive architecture that is supported by empirical evidence uses the chunk production style representation. This includes models such as ACT-R [And93], CAPS [JC92] and SPAN [Byr96].[2,3]

Procedural knowledge or rules act as links between chunks (such as propositions) that are stored in memory. Each chunk has an associated activation value and any active chunk will experience decay over time. Decay contributes to the limits on the capacity of working memory by providing a mechanism for losing or displacing chunks from working memory. Losing information from working memory corresponds to the recency mechanism in Walker's discourse model of attentional/working memory [Wal96c]. Each chunk also has a strength or bias associated with it which amplifies any activation propagated to it. The strength is a function of the frequency with which the chunk is used.

[2]SOAR uses a chunk production representation but does not make any claims about a complete architecture for working memory since the focus is on learning new productions and chunks.

[3]A production rule architecture was also used in earlier discourse research [Jos78]. In this early architecture, active elements are determined by conflict resolution instead of spreading activation. Also this early work viewed production rules as inference rules, whereas the current memory models view productions as associative links.

Each production is a pairing between input conditions and output activations. If all the matching chunks in the left-hand side of a production are above the activation threshold then an activation will be propagated to the chunks in the right-hand side. These target chunks will each receive an activation input at the level indicated by the link. The chunk production style models run in cycles, where a cycle consists of a phase of identifying all the matched productions and a phase of propagating activations.

Memory is not the only place where information is stored; there is also information provided by the external world. Much less is known about the interaction between the external world and memory but it seems safe to assume that this information doesn't decay as rapidly as the information in working memory. SPAN treats external memory as if it doesn't decay. However, since not all of the information in the external world is always available, the model can focus attention on parts of the external world. This is implemented in SPAN as a window on the external memory that can be moved by productions. The information inside of the window is all that is available from the external world for matching.

Finally, the treatment of goals is handled differently by the various chunk production models. SPAN allow goals to decay and be lost from memory whereas ACT-R and CAPS do not. Since there is some evidence for at least a stack-like hierarchical goal management capability, SPAN links goals hierarchically so that a parent goal continues to receive activation from its children and vice versa.

While Walker was able to use Landauer's trash can model of memory [Lan75] in her simulation environment to model attention/working memory [Wal93], this model is not an adequate representation of memory for more complex tasks. Landauer's model captures the recency and frequency phenomena associated with list recall experiments and while Walker's domain task is nearly isomorphic to list recall, the Coconut task is not. The COCONUT task and our hypotheses about redundancy in redescriptions requires a memory model that accounts for the associations between chunks as well as recency and frequency phenomena. Otherwise the associations between domain goals and known items will not be captured.

Although we will not use a cognitively plausible associative memory model such as SPAN or CAPS in this dissertation to relate objects and object attributes to task and discourse goals, it is important to consider more complex cognitively plausible memory models in place of the typical Grosz & Sidner stack model [GS86] in future experiments. However, we may find it more productive to wait for more guidance on the difficult questions

of how objects are represented in memory before substituting associative memory for the simpler stack model.

3.4 Sources of Influence

To arrive at a set of potential influences on attribute selection that could complement the influence of the identification intention, we will consider work on what influences discourse structure and the functions of repetition. We know that these factors influence content selection at the propositional level and hypothesize in Chapter 4 that they could also influence selection at the object description level as well.

3.4.1 Informational and Intentional Relations

Work on the interpretation and generation of language usually divides the relations that influence how individual utterances combine to form a larger unit of meaning into informational and intentional relations. With informational relations, the focus is on the meaning that comes from the semantic relationships between the utterance-level propositions (e.g. effect, cause, condition) whereas with intentional relations, the focus is on the possible intentions of the speaker in presenting two discourse elements consecutively (e.g. motivate, contrast, elaborate).

A significant number of natural language generation systems use the rhetorical relations defined by Rhetorical Structure Theory (RST) [MT87b] to help determine the structure of the discourse they produce ([Hov91, MP89] *inter alia*). [MP92] argue that the subject matter and presentational relations can be viewed as informational and intentional relations. The problem with RST is that while it allows that multiple relations can hold between two discourse elements, it advocates selecting one as primary and ignoring the others. Furthermore, [MP92] argues that given the guidance on resolving ambiguity that is presented in [MT87b], only the intentional relations will be selected for the analysis.

[MP92] reiterate the need for both types of relations for processing discourse. Although both relations may not be recoverable for every pairing of discourse elements, recognition and tracking of both are important for interpretation and generation. In the case of interpretation, recognizing one type of relation may allow the hearer to infer the other type of relation. In the case of generation, when participant B responds to participant A's contribution, A's follow-up may depend upon the intentions behind A's original contribution.

3.4.2 Repetition and Risk-Taking in Dialogue

One of the functions of a redundant utterance in dialogue is to manipulate the hearer's attentional focus so that a proposition is salient [Wal93]. In an interpretive (vs. empirically supported) analysis, [BPC94] compared the types and amounts of repetition for novice-expert interactions. They were primarily concerned with how repetition strategies contribute to the formulation and articulation (i.e. joint authorship) of knowledge. They found that the novice uses repetition to focus or seize one aspect of what the expert is saying. It is a way of diverting the expert's attention to the novice's specific concern and therefore acts has an attention getting function.

Another function of a redundant utterance that is discussed and supported in [Wal93] is to provide evidence that a proposition has been either understood or accepted (in the sense described in [CS87]). For example, a repetition can indicate that the hearer heard what was said. In the case of novice-expert interactions, the expert can repeat to both acknowledge the novice's responses to questions and to reinforce the knowledge. The claim in this case is that cognition is accomplished via repetition [BPC94]. The repetition evokes associations [Joh94].

Further, a redundant utterance can support the inference that the hearer not only understood but accepts or believes the proposition. This function serves to keep the participants coordinated or to realize what is mutually understood or believed. This type of utterance can also simultaneously demonstrate that an inference was made and ensure that the conversational partner makes the same inference [Wal93].

In deciding whether to repeat something there are risks involved. If a proposition isn't repeated then there are the risks that the desired function isn't achieved; for example, the desired cognition associations will not be made, mutuality may not be achieved or the hearer's attentional focus will be wrong and the desired effects will not be achieved. However, there are undesirable effects from the overuse of repetition. It may actually interfere with problem solving [Wal93] in that attention is diverted from possible alternative solutions.

Although [Car92] mentions the need to consider the effects of risk mitigation on the specification of referring expressions, she does not explore this since there there were no referring expression specification problems in her domain and corpus.

One can view risk mitigation in terms of confidence measures and confidence thresholds. In Edmonds' model of the interpretation and generation of referring expressions [Edm94], the confidence one has that a referring expression will be correctly resolved is expressed in terms of numeric values that represent the saliency of object attributes

and a threshold for how confident one should be. In this case the confidence measure is equivalent to saliency. Although Edmonds notes that attribute saliency should depend on context, he took the middle ground by considering only the static object type and attribute saliency. While object type captures some information about context it does not capture the influences of the cognitive architecture or the influences of problem solving and discourse intentions. Since the approach is static, there is no basis for changing the referring expression as the context changes. So variations in referring expressions are due simply to changes in risk-taking levels.

Chapter 4

Approach

To understand why redescriptions might be non-minimal with respect to a goal for the hearer to identify what is described, we need to consider some of the possible reasons and theoretical explanations for including extra information in a description. Although a nominal expression is generally the main way in which a description is linguistically realized, we also consider how the rest of the sentence and other nearby sentences can contribute to the description that the hearer may be mentally constructing. Recall from Chapter 1 that we are focusing on redescriptions since we assume and it is generally believed that their main goal is identification. With initial descriptions the goal to present new information is also frequently applicable (e.g. when an object attribute is not already mutually known). In the domain we are studying (see Chapter 2) none of the objects are mutually known initially. With redescriptions we can better distinguish between what is needed for identification purposes and what is potentially extra.

In Chapter 2 we presented several arguments for why studying a corpus would be beneficial in understanding content selection for redescriptions. We described the COCONUT corpus but did not address whether it was an appropriate one to study. We anticipate that different corpora, depending upon the domain of discourse and the communications medium, might exhibit different degrees and types of non-minimality (as considered in Chapter 2) and so we want to be certain that the corpus we have selected exhibits an interesting degree of non-minimality. We will begin this chapter by showing that the COCONUT corpus does exhibit a degree of non-minimality that we believe is adequate for our purposes.

Recall the explanations that have been considered in the cognitive processing literature to account for non-minimality:

- gestalt: properties are selected because they make the physical search for the target object more efficient.

- lexical focus: previous descriptions are repeated because they have been established as appropriate for identifying the target object.
- incremental minimality: true minimality is computationally expensive so we prefer an incremental approach to near-minimality.

Next, we present some additional, intentional reasons for non-minimality and we formulate hypotheses that are motivated by these reasons. We then tested these hypotheses by looking for correlations between the corpus conditions that match our hypothesized reasons for a selection decision and the actual selection decisions exhibited in the corpus. Next we implemented a parameterizable algorithm that has identification as a main goal. Various parameter settings allow us to experiment with internals that are not yet well defined. We then used the corpus for testing the performance of the various settings of the algorithm. Part of the information in the corpus about each utterance and the dialogue it appears in become the input for the algorithm. We compared the description selection decisions made by the various settings of the algorithm to what was expressed in the corpus for each domain object redescription. We expected to find that the intentional settings for making description selection decisions would perform significantly better than the cognitive processing limitation approaches.

4.1 Non-minimality in the COCONUT Corpus

During our preliminary investigations of the COCONUT corpus, we noticed that there seemed to be a large number of subsequent references to furniture items using highly redundant descriptions. This observation includes any explicit information in an utterance that relates to the mental model pointed to by the discourse entity.[1] For example, if a discourse entity[2] is mutually known to have the color *red*, then including *red* in the utterance, as with "My chair is red", makes the description of the discourse entity redundant.

To confirm our initial impressions, we must first determine for each description, what other mutually known items may be salient for the dialogue participants. Following the terminology of [Dal92], we call these salient, mutually known items the distractors. Different definitions for a distractor set are suggested in the literature [Dal92, Pas96, GS86, Lev89]. Since it is not clear what definition of the distractor set is correct, we take it as a matter of empirical investigation to try several plausible definitions taken from current theories in computational linguistics and psycholinguistics. Using several definitions of the

[1] Recall that a discourse entity links an NP to a mental object that can change over time.
[2] Our use of discourse entity in this way means the mental model pointed to by the index.

distractor set, we can see how much redundancy exists with each definition. If there is a high degree of redundancy no matter what the distractor set definition is, then it is plausible given our current theories that the coreference descriptions do indeed have a high degree of identificational redundancy.

Grosz and Sidner's theory of the attentional and intentional structure of discourse [GS86] provides an account of saliency that is widely accepted within computational linguistics for task-related dialogs. In their theory, the content of a focus space and a stack of focus spaces is defined by the task structure. A change in task or topic indicates the start of a new discourse segment. All of the discourse entities described in a discourse segment are classified as salient for the dialogue participants while that focus space is on the focus stack. The relative saliencies between focus spaces are left unspecified and are still to be determined.

Passonneau [Pas96] uses Grosz and Sidner's theory to define a distractor set. Her distractor set is the union of all the discourse entities in the current discourse segment and all the entities in the last segment that contain the entity to be described. To be conservative, she assumes that if the most recent segment containing the target entity is not the same as the current segment then it is a resumption and the intervening focus spaces should not be included in the distractor set. The descriptive content that is needed to avoid ambiguity and the size of the distractor set are positively correlated. So Passonneau's model, which minimizes the distractors, also provides a conservative measure of the number of redundant redescriptions in a corpus.

Using this definition for the distractor set, Passonneau found that only 6% of the non-pronominal noun phrases (NPs) in the Pear stories corpus [Cha80] contained redundant identificational information. We call this distractor set definition **SEG** and use it to confirm our initial impressions of a high degree of identificational redundancy in the COCONUT corpus[3]. Following Passonneau, we first identified all the NPs that were potentially overspecified by selecting subsequent descriptions that are longer than their previous descriptions. We then filtered this set of NPs using the **SEG** distractor set definition. With the first step, we found that 51% (84 of 166) of the full NPs in the COCONUT corpus were potentially overspecified. And after filtering these with **SEG**, we found that 46% (76 of 166) of all the non-pronominal NPs were identificationally redundant. This seems to confirm our initial impressions.

Because it is possible that **SEG** is not the best distractor set definition for all genres, we also tested for identificational redundancy using some other cognitively motivated

[3]We describe how we segmented the COCONUT corpus in Chapter 5 in Sections 5.1 and 5.5.1.

distractor set definitions. We followed the same methodology as above but we substituted the following distractor set definitions for **SEG**:

- **ALL**: all entities previously mentioned in the discourse
- **1UTT**: all entities mentioned in the previous utterance
- **SEG+SOLN**: all entities in the current discourse segment and all the entities currently in the solution set for the action being addressed
- **5UTT**: all the entities mentioned in the previous 5 utterances

ALL and **1UTT** are two simple and rather implausible definitions for the distractor set and have been included to determine what happens with identificational redundancy at the extremes. Actually, there is some theoretical merit to **ALL**. Poesio [Poe93] indicates that the distractor set should be based on a combination of the perceptual focus space and the discourse focus space. In the COCONUT setting, the designers often created graphics icons to help them remember what items their partner had described to them and which items they had presented to their partner. These graphical representations could behave as a perceptual focus space for the speaker. However, in view of the evidence discussed in [CM81], the participants would have to assume they are both keeping such a record (i.e. the representations would need to be part of their common ground) in order for the dialogue to proceed successfully.

1UTT, while extreme, also represents a focus space similar to that used in computational research on centering [WJP97] to determine acceptability conditions for anaphoric reference. We expected that this immediate focus set would be too limiting as a distractor set.

SEG+SOLN assumes that items that have been selected as part of the solution for the action being addressed are in focus during the rest of problem solving. The reasoning task provides a rationale for this definition since these items serve to limit the money that is left to spend and may be applicable for determining color match constraints. Finally, **5UTT** is a simple approximation of recency in the discourse.

Table 4.1 shows the percentage of identificationally redundant descriptions under each distractor set definition. The degree of redundancy or non-minimality runs as low as 39% with the **ALL** distractor set definition and as high as 46% with the **SEG**, **1UTT** and **SEG+SOLN** definitions. No matter which of these distractor set definitions we use, there is still a high degree of redundancy in the COCONUT corpus. The redundancy is therefore definitely a theory-independent phenomenon in this corpus.

	degree of redundancy	average distractor set size
SEG	46% (76)	5
ALL	39% (64)	19
1UTT	46% (76)	2
SEG+SOLN	46% (76)	4
5UTT	44% (73)	4

Table 4.1: Degrees of non-minimality under different distractor set definitions

4.2 Explaining Non-minimality

4.2.1 Factors that Influence Description Selection

There are numerous potential influences on description selection for both initial and subsequent object descriptions. Among the many possible influences that have been considered thus far in the computational research are: identification goals [App85c, Kro86, Dal92, HH95, Loc95, Pas96], goals to communicate new information about an object [Rei90a], syntactic constraints (e.g. it may not be syntactically feasible to express the entire description in one utterance) [App85a], semantic constraints (e.g. subcategorization constraints imposed by the verb selected for the utterance) [App85a, SW98], perspective (e.g. which properties of an object are most salient) [McC86], and attitude (e.g. which properties and lexical choices convey the intended attitude) [Hov88].

We also know that the intentional and informational relations that help bind together utterances to form a coherent discourse [MP92] can influence content selection and linguistic form at the utterance level [McK85, MM95]. If we assume that this influence extends down to the level of nominal expressions as well, we can arrive at some other factors that could influence content selection for descriptions. We know that utterances can, among other things, show contrast, elaborate or motivate other utterances [MT87a]. For example, in a context where all the object information is mutually known, there is a color match constraint, a table needs to be chosen and red chairs have already been selected, the second utterance in

(1) Let's use my table.
It is red.

can be viewed as a motivation for the proposal made by the first utterance,"Let's use my table". We can replace both of the utterances in this same context with "Let's use my red table". Now, the discourse appears to presuppose the same information is present, yet an intention to motivate the proposal might still be recognized by the hearer. That is, we

wonder whether the information must be in a separate utterance to increase the likelihood of the hearer making the motivation inference. Perhaps the goal to provide motivation for a decision can be satisfied in part by the property information in the nominal expression *my red table.*

In slightly modified contexts, the same utterances could satisfy elaboration and contrast intentions. For example, the second utterance could be supplying information that isn't mutually known but that is needed for problem solving (a case of elaboration) or if the context allows the first utterance to be interpreted as counterproposing an alternative solution, then the second utterance could be pointing out an important difference between the two alternatives (a case of contrast).

Finally, if we consider studies about redundancy in discourse, we find additional factors that could influence content selection for object descriptions. Recall that in Section 3.4.2 we listed some of the functions of repetition in dialogue. Walker's dissertation work [Wal93] summarizes the communicative functions of redundancy at the utterance level as:

- Attitude: to provide evidence that supports beliefs about mutual understanding and acceptance.
- Attention: to make a proposition salient.
- Consequence: to strengthen evidence that certain inferences are licensed.

These functions cover cases where a discourse partner might repeat an utterance to show that the utterance was understood or accepted [CS89, Bre90, Wal92, Wal93, Wal96b], to make a proposition more readily available for the discourse partner's inferential processing [Wal93, Wal94, MP89], or by providing no new information to the interaction show that the current stage of the interaction is completed (e.g. summarization) [WS88, JD97].

As with the intentional and informational relations, we question whether repetition might not have the same types of communicative functions regardless of the level at which it appears. Looking again at the same example, the second utterance in

(2) Let's use my table.
It is red.

is informationally redundant according to [Wal93] given the context we presented. We question whether replacing these utterances with "Let's use my red table." cancels out the redundancy function. We do not directly explore whether such utterance replacements have

equivalent intentional and informational relations and redundancy functions. We merely present them as motivation for the additional sources of influence for description selection that we will suggest in Section 4.2.2.

In concluding our discussion of the possible influences on content selection for object descriptions, we wish to point out that it would be impossible to dissect a particular description and definitively list which factors caused each property to be expressed. The property might have been selected because it partially or completely satisfies one or more goals. Since there is a possibility of goal overloading[4] at many levels of language generation (e.g. syntactic and semantic levels [SW98] and the pragmatics level [DW96]), it seems reasonable to expect that goal overloading might also be applicable when generating object descriptions. Since we do not expect to be able to precisely identify the reasons why each property was selected for an object description, we instead look for correlations between possible causes and actual content decisions (see Section 4.3 and Chapter 5).

4.2.2 Possible Sources of Non-minimality in Redescriptions

Recall our high level hypothesis: that because of multiple influences on description selection and the possibility of goal overloading [Pol92], extra properties might be included in cases of subsequent reference or alternative properties might be preferred. These properties may support communication goals other than unique identification of the target object. In this section we will formulate a set of testable hypotheses that refine this high-level hypothesis.

It is well established in the research community that goals for the hearer to identify an object and goals to communicate new information about an object influence which properties of an object are included in a description or redescription: (1) New information might be communicated about an object because of domain goals; i.e., if the hearer is to use an object in an action then he needs to know certain object properties as indicated by the action.[5]; (2) New information might also be communicated in a description in order to answer a hearer's question, e.g., How fast is the Pentium II? You're looking at the 750MHz Pentimum II right now, but we also have 800s and 850s. In case (1), where domain goals are operative, the action determines the selection of the relevant properties. In case (2),

[4]The general term, overloading, was defined by Pollack [Pol91] to be the use of one action to satisfy multiple goals and the phenomenon of one utterance satisfying multiple goals was first noted by Appelt [App85a]. Stone and Webber [SW98] point out the connection between overloading and textual economy at the syntactic and semantic levels. It seems plausible given the idea of overloading and textual economy that a property in an object description could satisfy multiple goals and result in textual economy as well.

[5]Since natural language semantics is a naive semantics [DMEPS89] there is not necessarily a one to one correspondence between semantic constraints and action knowledge [Di 98]. So while semantic constraints may explain some property requirements it is not guaranteed to capture all that are relevant to an action.

where the communication of new information is the purpose, problem or question constraints select the relevant properties. If an action or constraint can influence the properties selected for expression, then it seems plausible that those properties can also influence inferences about actions and constraints [McC86, MWM85]. In simulations involving two artificial agents, [Wal93] found that the agents were better able to make required inferences when mutually known premises were made salient. Property information can be a premise for some inferences and in Walker's simulations she treats the color property as a premise for inferences about matched colors. This indicates that properties could be particularly valuable in communicating changes in domain goals and constraints and might help to explain why a mutually known property might get repeated in a redescription.

To illustrate this possibility for the COCONUT domain, assume that there is an initial constraint setting to match colors. Also assume that the person about to speak has just discussed using a red table and prior to that had introduced tables of a variety of colors and four $100 red chairs, and four $75 green chairs. Finally, assume that she has just decided that she wants to drop the color match constraint so that they can use the cheaper $75 chairs. When she communicates her suggestion that they use these chairs, we hypothesize that she is less likely to say *the $75 chairs* although it adequately and economically identifies the target chairs and the price of the chairs are highly salient for her. Instead we expect her to say *the green chairs* or *the $75 green chairs*. By choosing "green," she still manages to adequately identify the target chairs while also enabling the hearer to more easily infer that she intends to drop the color match constraint. She has saved the expense of having to explicitly say *Let's forget about matching colors* [Wal93] and has reduced the risk of the hearer not making the inference [Car92] and thinking there is a mistake in his understanding of the color of the $75 chairs. There is an alternative explanation here: one might argue that the color property is by default highly salient and would generally be included in the description. However, we found that color is not always included in a full NP redescription even when it could rule out some distractors under the **SEG** definition of distractors (43 of 197).

During our initial investigations of the COCONUT corpus, we observed that as the human designers adjusted problem solving constraints, such as the color match constraint in the above example, they often did not talk directly about these adjustments. In fact, 38% of constraint changes are made implicitly. These constraint changes must be recognized, otherwise an agreement could not be reached on what a satisficing solution is. In all of the interactions in the COCONUT corpus, the designers believe they have agreed on what the solution should be. In only one case, there is a disagreement in the solutions recorded by

two of the designers. Given that the constraint changes must be conveyed somehow, we expect that the hearer must infer the change for the speaker's contribution to make sense. We might expect to find then that some properties might be included in the subsequent references both to help uniquely identify the target object and to enable the hearer to infer constraint changes. Given the results of [Wal93], it seems plausible that the redundant property information may be making it easier to infer changes to the problem definition so that it is not necessary to directly communicate the changes. Therefore, out first hypothesis is:

> DOMAIN CONSTRAINT CHANGES hypothesis: Properties related to constraint changes are expressed in a context where the change must be inferred by the hearer.

Considering the possible influence of informational relations in discourse, we know that these relations cause additional information to become salient for the hearer. If the information is already salient then it will not be necessary to explicitly state it [JW96]. For example, in the COCONUT corpus, if a speaker says

(3) I have the chairs and table.
That will cost us $500.

there is an *effect* informational relation between the two utterances: the action of buying the items causes a $500 diminution of the designers' joint budget. The hearer can use this information about the total cost to help pick out the appropriate entities. It might be less of a cognitive load for the hearer to actually say the price of each item but then it seems less likely that the speaker would give the total. A possible advantage of giving the cost effect for the action is that it helps convey the speaker's commitment to this potential solution [DJTM00]. Thus our next hypothesis is:

> INFORMATIONAL RELATION hypothesis: When the effect of, goal of, or the constraint imposed on an action is communicated then the properties made salient by such a relation are **not** expressed.

Next we consider the possible interaction between intentional relations and redundancy. Walker [Wal93, Wal94] experimented with a discourse strategy called explicit warrant. A warrant is equivalent to the *motivation* relation in [MT87a, MP89] in that it is intended to motivate or convince the hearer to do or agree to a proposed action. Walker found that it is beneficial under certain cognitive resource limitations to make warrants

explicit even when the warrant is mutually known. Recall that we presented an example of this for the COCONUT domain in Section 4.2.1. What remains to be seen is whether the warrant can be included in a redescription and still perform the same persuasion function. Our hypothesis in this regard is:

> PERSUASION hypothesis: Property values that are pivotal for deliberation are expressed in the context of goals to communicate a proposed action.

(4) S> ...I have a $300 yellow sofa...

G> My sofa's are more expensive
so buy *your $300 yellow sofa.*
Also.... Your $550 + my $450 is $1000

S> You are right,
we have $1000 to spend.
I will go ahead and buy *the $300 yellow sofa.*

Next, we consider the influence of the redundancy function on property usage when no new information is provided in a contribution to the dialogue [WS88, JD97]. Repeating properties for a recently evoked item could show that the current stage has been completed while doing so for an older item could indicate that a higher level subproblem has been completed. In (4), S's second utterance appears to end a stage in the interaction, in this case it seems to be the end of the agreement process for a *select sofa* action as described in [DJTM00]. For this case our hypothesis is:

> COMMITMENT hypothesis: In the context of a commitment to a proposal, all the properties expressed in the proposal will be repeated.

However there are cases in which no new information is presented and there is no show of acceptance of an object that was just proposed. That is, the current entity wasn't evoked in the previous turn. If a decision had already been made about the action associated with the current entity and the participants had moved on to a new part of the task following the decision, then we call the description part of a summary. Summaries differ from commitments in that they are delayed redescriptions. The COCONUT dialogue excerpt in (5), illustrates a summary. Note that D summarizes both living room (as requested) and dining room items.

(5) G> I got the rug.
What do you have in the living room and what are the prices of the items

D> the the green sofa in the living room 350
dining room—>
3 yellow chairs 75 each, 1 high-table yellow, 1 yellow rug

For this case our hypothesis is:

SUMMARIZATION hypothesis: In the context of a previously completed problem or subproblem, all the mutually known properties for an item will be repeated.

The last redundancy function influence we will consider is showing that an utterance was understood and accepted. In the COCONUT corpus, the hearer sometimes repeats back the entire content of the description in the utterance or turn immediately following the initial description of the domain object. For example, in (4), G repeats S's description of the sofa, although the sofa was introduced by S.

This is very similar to the lexical focus explanation (see Section 3.2) but instead of being an agreed upon way of referring to the object, we claim that it has an intentional function associated with it. It helps to verify that the property information was correctly understood. For this case our hypothesis is:

VERIFICATION hypothesis: In the context of a newly introduced entity, all the properties expressed will be repeated by the hearer in his/her next turn.

We used corpus analysis to look for correlations that lend support to these hypotheses. If these correlations hold in the data, then we can use this information in designing an algorithm that generates redescriptions of objects. Likewise, in future work, we can design interpretation algorithms that use these sometimes overspecified redescriptions to see if the inferences we discussed are more likely to be correctly made.

4.3 Methodology for Hypothesis Testing

To test our hypotheses, we annotated features of the dialogues in the COCONUT corpus that we expected would help define the contextual situations expressed in our hypotheses. In general, the features represent the entity properties that are expressed for redescriptions, inferences about constraint changes, informational relations between clauses and the agreement structure. We will describe these features in more detail in Chapter 5 and link them to the hypotheses that they help test.

We did a chi-square analysis [SC88, Kep91] using these corpus annotations. We checked whether there are correlations between the contextual situations that combinations of the features represent and the properties that were selected for expression in the corpus.

A chi-square analysis is a statistical technique that compares actual frequencies to expected frequencies for pairs of factors. To perform a chi-square analysis we need to set up an $n \times n$ contingency table. In our case, the rows of the table represent the contextual situations and the columns represent whether the property or properties associated with the situation were expressed in the redescription of the target object. For example, with the DOMAIN CONSTRAINT CHANGE hypothesis, the rows of the contingency table indicate whether the redescription occurred in a situation where the constraint change was implicit vs. otherwise (i.e. either there was no change from the previous utterance or the change was explicit).

Typically the contingency table is a 2×2 table and if the chi-square measure for the table is significant ($p < .05$ that we are wrong about a correlation existing between the factors), then one can tell which cell made the difference. However, when the table is larger than 2×2, we have to do chi-square partitioning to tell which cell made the difference [SC88, Kep91]. The contingency tables are sometimes larger than 2×2 when we combined multiple annotation features to define the contextual situation. For example, if we use features A and B to define a situation, we sometimes want to check $\neg A \wedge B$ and $A \wedge \neg B$ as well as $A \wedge B$ and $\neg A \wedge \neg B$. The partitioning creates a series of 2×2 tables and a separate chi-square measure of each of these partitions (using a different chi-square equation than for the original contingency table). Partitions with significant chi-square values are the sources of the difference for the overall $n \times n$ table.

If the chi-square measure for a hypothesis is significant and the source of the difference is as we expected, then we included a check for that contextual situation in our description selection algorithm. We also tried a version of context checking that includes all of the contextual situations represented by our hypotheses regardless of their chi-square analysis results. There were some cases were the expected frequency was too low. This indicated that we did not have enough instances in our data and the standard chi-square distribution was not applicable. In those cases we cannot assess the significance of the χ^2 measure so we instead use a Fisher Exact test.

Next we implemented a description selection algorithm that had identification as its main goal and that was parameterized for how it made selections to satisfy the identification goal, whether it did the context check indicated by our hypotheses and which context checks it performed. The parameter settings for how the identification goal was handled allowed us to test the performance of all the cognitive processing explanations (e.g. gestalt, incremental minimality). We then used the annotated corpus to test the performance of the algorithm under the various parameter settings. We used the same

features that defined the contextual situations for the chi-square analysis for the description selection algorithm as well.

We also needed to build representations of the discourse entities expressed in each dialogue as the dialogue progresses since the algorithm needed information about the entities already expressed in the discourse. The target object to be described was also represented as a discourse entity. We needed to annotate additional features of the corpus in order to support this. These annotation features will also be described in Chapter 5.

We compared the description selection decisions made under the various parameter settings of the algorithm to what was expressed in the corpus for each domain object redescription. We expected to find that the intentional settings for making description selection decisions performed significantly better than the cognitive processing limitation approaches where the only goal was identification. To compare the performance of each setting of the description selection algorithm we used analysis of variance (ANOVA) to first see if there were differences among the groups of selection settings. If there were significant differences we then used multiple comparison techniques to do pair-wise comparisons [SC88, Kep91]. The pair-wise comparisons told us which settings produced the best results relative to the other settings.

Chapter 5

Analysis of the Corpus

It is not always best to use as few words as possible in redescribing a previously mentioned object. In particular we showed in the previous chapter, based on a preliminary analysis of the COCONUT corpus, that a relatively large number of non-minimal redescriptions can occur in collaborative design dialogues. Following Passonneau's method for estimating the number of non-minimal noun phrases, we found that 46% of the non-pronominal redescriptions in COCONUT are non-minimal. This is large compared to the 6% that Passonneau found in the Pear narratives [Pas96][1].

To identify good alternative description selection strategies for redescriptions in collaborative dialogues, we said in the previous chapter that we would use corpus analysis to inform our strategies and then test the usefulness of those strategies by comparing the content they select against what was included in the redescriptions we find in the corpus. To conduct this corpus analysis, we must define an appropriate set of features for some testable hypotheses. There are some trade-offs in trying to find features that we believe we can reliably annotate in the corpus and the hypotheses that we can test. Each places constraints on the other. We start with an initial set of hypotheses and then attempt to define the features we need to test them. If we do not believe we can realistically annotate the necessary features for a particular hypothesis, then we must either restate and refine the hypothesis or give up the hypothesis.

Our approach to using the corpus for hypothesis testing consists of two parts. First, we look at property usage in redescriptions and sets of features that our hypotheses predict should be related to description selection. In particular, we wish to check for possible correlations between domain and discourse goals and the properties used to describe the objects in the utterances that address these goals. For example, one of our hypotheses

[1] Although Cremers [Cre96] estimated 73% of the descriptions in her building task corpus were non-minimal, we do not have the details of how she arrived at this number and cannot use it as well to compare the degree of redundancy in the COCONUT corpus. When we used a less conservative estimation approach, we found that the number of non-minimal descriptions in the COCONUT corpus was closer to 70%.

predicts that goals to communicate changes in task constraints can be overloaded with identification goals when the changes are related to object properties. This means that we need to look for correlations between property usage in redescriptions and the purpose of the utterance.

Second, we check the performance of algorithms that are based on the explanations for redundancy that we described earlier, as well as the performance when we use selection strategies based on our testable hypotheses. The selection strategies are informed by the correlational part of the hypothesis testing. We can eventually use the strategy performance results to further refine the hypotheses and selection strategies.

5.1 The Annotation Scheme

In defining an annotation scheme for a corpus, we must first consider the hypotheses we wish to test and the correlations we would expect to find that would at least partially support these hypotheses. Our high-level hypothesis, from Chapter 4, is that goals and constraints from the domain and discourse influence the selection of properties in redescriptions. Our final set of testable hypotheses are:

- Domain Constraint Changes: Properties related to constraint changes are expressed in a context where the change must be inferred by the hearer.
- Informational Relations: When the effect of, goal of, or the constraint imposed on an action is communicated then the properties made salient by such a relation are **not** expressed.
- Persuasion: Property values that are pivotal for deliberation are expressed in the context of goals to communicate a proposed action.
- Commitment: In the context of a commitment to a proposal, all the properties expressed in the proposal will be repeated.
- Summarization: In the context of a previously completed problem or subproblem, all the mutually known properties for an item will be repeated.
- Verification: In the context of a newly introduced entity, all the properties expressed will be repeated by the hearer in his/her next turn.

Keeping in mind our high-level and testable hypotheses and our testing methodology, we need two sorts of corpus annotation features: (1) utterance level annotations that

capture problem solving and discourse features, and (2) discourse entity level annotations that capture (a) the definitions and updates for discourse entities as a dialogue progresses and (b) the properties selected to describe the discourse entities in subsequent references.

Features of type (1) are needed to capture information about the role an utterance plays in the discourse and what actions and constraints it communicates for the problem solving task. Although the specific features for annotating actions and constraints are domain specific, how we identify the actions and constraints is general to problem-solving dialogues.

To annotate for the domain constraint changes hypothesis, we assumed the initial constraint settings were ones that would maximize the number of points earned.[2] In general, these initial settings held true for all of our participants since the task instructions that explained the scoring for solutions was the only common ground that the participants had at the start of the problem solving trials. Annotators were instructed to pick an appropriate constraint description from a given list whenever there was a change to that constraint from its previous setting and to indicate whether the change was implicitly or explicitly made.

In annotating the task, we also needed to identify the task structure and the discourse segments. Our approach to discourse segmentation is motivated by [Ter85]. Terken (*inter alia*), proposed that a change to a different domain action is a cue for the non-linguistic task structure. Each domain action provides a discourse segment purpose so that each utterance that relates to a different domain action or set of domain actions defines a new segment.

There is one general action type of interest for us in COCONUT: *select furniture for a room.* But there is a hierarchy of *select* action types for annotation purposes that ranges from this most general action type to the most specific action types. We assume that there are at least three and at most five specific action types that get addressed in each problem solving interaction (distinguished by furniture type and room): select four chairs for the dining room, select a table for the dining room, select a sofa for the living room, and select a set of optional items for the living room and dining room. Note that these action types are one level up from fully defined executable actions, in that parameter assignments have not yet been made.

Contiguous utterances that discuss a specific action type are taken to define a discourse segment. When one or more domain action types is being addressed, we consider that set of action types to reflect the current problem solving goals and to represent the discourse segment purpose. An utterance that relates to a different domain action type than

[2]See Chapter 2 for a description of how solutions are scored.

that of the previous utterance defines a segment boundary. For example, in the constructed dialogue (6) in the situation where only the living room contents remain to be decided, Agent-A's utterance addresses the action type "get a sofa for the living room" whereas Agent-B's utterance addresses the action type "get an optional item for the living room". This means that Agent-B's utterance is at a segment boundary since it addresses a different domain action. Utterances that introduce or restart action type discussions, while also continuing active discussions of other action types, are interpreted as starts of embedded discourse segments.

(6) Agent-A: Let's get the red sofa.
Agent-B: I have a red rug for $25 and red lamp for $25.

Not all utterances in the dialogue directly address one of the domain action types we are focusing on. We consider such an utterance to be part of the current segment or a nearby segment that it has some other defined relationship to. For example, "I have $50 left" is an effect of some action or actions that use up part of the budget. It does not directly address a domain action by itself but should be included in the same segment as the action or actions that cause it.

We instructed the annotators to indicate what action type was being addressed in each utterance by considering whether any furniture items or furniture templates being discussed in the utterance could unambiguously be related to one of these actions. Annotators were also asked to distinguish between when an action type was first addressed and when the utterance continued the discussion. If the relation of the furniture item or template to action types was ambiguous, the annotators were instructed to indicate the highest level action type that was unambiguous (e.g. select items for the dining room). Also, we needed segmentation information in order to build distractor sets for the selection algorithms we tested.

Informational relations between utterances and intentional relations between sections of the discourse can also affect the nature of the discourse. An informational relation between two utterances describes how the contents of the two utterances are related in the domain. An intentional relation indicates how the intentions associated with a particular part of the discourse are related to the intentions associated with another part of the discourse. We annotate informational relations between utterances but do not directly annotate any intentional relations. We annotate informational relations such as action:effect, action:condition and goal:action in order to test the informational relation hypothesis.

We do not annotate intentional relations directly because in the case of a collaborative problem solving corpus such as with COCONUT, most of the intentions would be

classified as trying to convince the hearer. That is, the speaker wants to supply enough information about the action to enable the hearer to reason about whether it is an appropriate action to take. We need a finer classification for the *persuasion* intentional relation in order for an annotation of these type of relations to be of any use to us. Instead of trying to come up with a finer-grained set of *persuasion* intentions we instead make use of the structure of the agreement process in collaborative problem solving since this can be considered as a way of further detailing the components of a *persuasion*. In [DJTM00], empirical evidence from the COCONUT corpus was used to support a division of the agreement process into the following three phases:

a. balance information and partially deliberate

b. propose options

c. dispose of proposal

In particular, we need to recognize [b] to test the persuasion hypothesis, [c] to test the commitment hypothesis and we need to recognize the end of the agreement process for an action type to test the summarization hypothesis. We used the annotations from the COCONUT project and the definitions for the following states that are associated with utterances and the actions they express [DJTM00]:

- partner decidable option: The speaker describes or offers an item but the current solution state is indeterminate because the speaker believes the hearer may have better options that he has not yet presented.
- propose: The speaker offers an item in a context in which he has already heard what the hearer has to contribute as an alternative.
- unendorsed option: The speaker offers an item in a context in which he has already heard what the hearer has to contribute as an alternative and he believes the alternative is better.
- unconditional commit: The speaker indicates a commitment to using an item.

Instead of directly annotating for proposed options, etc., the definitions are based on several lower level annotated features. One reason for this is that it was more difficult to obtain intercoder reliability for higher level features. Another reason is that the authors wanted to validate the definitions for the new constructs they proposed (e.g. unendorsed options).

In addition to these discourse and problem solving features, information is also annotated about the furniture entities introduced in the dialogue. The main objective is to identify the entity that the speaker has in mind, and how the information about that entity is communicated. Both initial and subsequent references were annotated so that we could capture how the description of a single discourse entity is developed during the course of the dialogue. By tracking the discourse entities in the dialogue we could tell when a subsequent reference to an entity might also add new information about the entity or correct erroneous information. For example in, "I have a $200 table. It is green.", entity_1 from the first utterance is ((type table)(owner A)(price 200)). The pronoun "it" in the next utterance corefers to entity_1 but also adds to it new information about the color of the object being referred to. The entity description then gets updated to ((type table)(color green)(owner A)(price 200)). These entity descriptions serve as input to the description selection algorithms. The algorithms cannot choose to use properties that are new to an entity (i.e. not mutually known) to corefer. In matching an algorithm's selections with those made by the human, choices about whether to describe a new property are not counted.

For the furniture entities, the annotators were asked to indicate the attribute-value pair information for each discourse entity in an utterance, and the sources for this information (e.g. from the utterance, the NP or locally inferred). Annotators were also asked to indicate whether the discourse entity was new or a coreference to a previous discourse entity and what other discourse entities the current entity might be related to. Here, some of the relevant relations include set, part-of, and class relations. Finally, we also asked the annotators to indicate which action type the discourse entity was related to, since multiple action types are sometimes addressed in a single utterance. We need to know which action type most directly influences that description.

5.2 Preprocessing of the Corpus

Before we annotated the corpus and validated our annotation scheme, we preprocessed the corpus by defining the minimal units for annotation; entity units and utterance units. The entity units are all the NPs that denote entities of type furniture, or pronouns that denote entities of type event that are in the `select furniture for room` hierarchy. Although a complex NP is often composed of simpler NPs, we only consider the NPs that describe the most distinct entities. For example in the NP *2 of the green chairs*, the entire NP is marked as an entity unit since it is more specific or limiting than the subcomponent *the green chairs* (i.e. this entity would be defined as having more than 2 green chairs). On the other hand, the NP *the green table and chairs* is marked as two entity units; *table* and

chairs. The highest level NP should not be delimited as an entity unit in this latter case because it is less specific.

Regardless of whether the NP is an initial or subsequent reference to a discourse entity or a definite or indefinite expression it is marked as an entity unit if the above definition applies. This is because both initial and subsequent references can describe discourse entities that represent a particular domain object or set of domain objects for the speaker. If a discourse entity was not intended to be related to a particular domain object or set of objects for the speaker (e.g. generic NPs as in "*Chairs* are a high priority."), it was not marked as an entity unit.

Implicit arguments of verbs were inserted as [0] whenever they were syntactically required arguments of the verbs and they denoted an entity of type furniture or an entity of type event that is in the `select furniture for room` hierarchy. Note that in (7), although there is a semantically implied argument (another sofa, whose price is understood), it is not syntactically required. If the implied argument were syntactically realized, it would be governed by the preposition "than".

(7) I do not have a sofa for a better price.

Wh-pronouns that have intended fillers of type furniture were also marked as entity units. Missing arguments for entity units of type event were not inserted. For example, *My rug matches*, is bracketed as *[My rug] matches [0]* since the verb *matches* requires two objects as arguments. However, the deverbal *the match* in *The match is good*, is not bracketed as an entity unit or as providing an implicit argument since this NP refers to a matching event.

Also, NPs such as *my inventory* were not delimited as entity units. In this case, the NP refers to an entity of type list and not an entity of type furniture. Although this entity has furniture items as part of it, it is not representationally a furniture item.

We used the same utterance segmentation that was carried out for the COCONUT project [DJP98]. The utterance segmentation follows Passonneau's guidelines [Pas94], which use the following criteria for identifying a functionally independent clause:

- It must contribute a proposition to the discourse that is semantically complete and fully specified, and if it is a full syntactic clause it must be syntactically independent and maximal.
- It must not be a formulaic clause serving the function of an interjection.

As a consequence, functionally independent clauses include, besides main tensed clauses, coordinate clauses, subordinate adjuncts, nonrestrictive relative clauses, clauses containing verb phrase ellipsis and response fragments.

5.3 Developing a Reliable Annotation Scheme

To develop and validate the annotation scheme, we conducted intercoder reliability studies using a balanced subset of the corpus. 30% of the corpus was annotated by two annotators.[3] We used the Kappa coefficient of agreement [SC88, Car96] to assess intercoder reliability; this measure factors out chance agreement between coders. The discourse processing community uses Krippendorf's scale [Kri80, Car96] to interpret and apply the Kappa coefficient, which varies between 0 and 1. Krippendorf's scale discounts any variable with $K < .67$, allows tentative conclusions when $.67 < K < .8$, and definite conclusions when $K \geq .8$.

In addition to using Krippendorf's scale to interpret the agreement measure, we must also test the significance of the measure to determine if the observed agreement was greater than what would be expected by chance.[4] If the significance level is good then we can assume it would be valid to annotate the rest of the corpus using just one annotator and that it would be appropriate to use the judgements of just one annotator when testing our hypotheses.

After an initial annotation and reliability study, we identified the deficiencies in the scheme by looking for consistent disagreements.[5] We then corrected problems in the annotation scheme and the instructions and re-annotated the dialogues by reviewing them relative to the changes in the annotation instructions. After re-annotating the dialogues, the reliability increased but was still generally lower than the .8 value that according to Krippendorf's scale indicates a reliably annotated feature. Next, we carefully reviewed the disagreements and found additional clarifications that were needed in the annotation scheme and the instructions. This time most of the disagreements had to do with conventions for recording the annotation features rather than with identifying feature assignments. After the convention problems were corrected, the reliability measures were all above .8 except for the three feature categories that involve inference (implicit changes in constraints, discourse inference relations, and informational relations).

[3] One annotator's area of expertise is linguistics; the other is the author of this paper.

[4] Note that this is different from the expected agreement due to chance that is accounted for by the kappa coefficient [SC88]. We computed the significance, z, using equations 9.30 and 9.31 in [SC88].

[5] Recent work by [WBO99], describes statistical tools for detecting patterns of disagreement and for producing Bias-Corrected tags. These tools are used to automate some of the process of developing an annotation scheme.

Our next step was to perform a partial reconciliation. The reconciliation was partial because we focused on just the three inference categories and did not try to resolve any remaining disagreements for the other feature categories. We found that the disagreements were primarily cases of omission by one or the other of the annotators. This is unsurprising for inferences. It is easy to miss information that isn't explicit while annotating since an annotator isn't actually involved in the problem solving. Also an annotator might easily miss the fact that she has made these inferences while trying to understand the dialogue.

After adjusting the annotations following the reconciliation, the reliability for discourse inference relations was above .8. For the other two categories, we identified the subfeatures that were causing a majority of the disagreements and decided to eliminate these subfeatures and any testable hypotheses that depended on them from our study. The decision was motivated by time constraints; the second annotator was not able to commit more time to the project so that we could do another iteration. We expect that further clarification of the instructions would have eventually resulted in adequate reliability measures for the problematic subfeatures.

It would be advisable to use multiple coders when annotating inference categories of features since there is a tendency for omission errors. Since we used just one coder for the rest of our annotations following the development and validation of the annotation scheme, we expect that we will only find a subset of the inferences of interest. This may make some of our hypotheses harder to prove.

Table 5.1 shows the final intercoder reliability measures and their significance after the two development iterations, the partial reconciliation pass and the removal of problematic subfeatures as described above. The results in Table 5.1 indicate (based on Krippendorf's scale) that all of the remaining feature categories are defined clearly enough so that they can be reliably annotated by one annotator and used in studies.

Although the *requests for properties* feature only allows tentative conclusions about the agreement, it is reliable enough for our purposes. We are not making claims directly related to this feature but instead are using it to filter out those cases where a speaker repeats a property because the hearer explicitly requested a repeat of it (e.g. What color did you say your table was?). Filtering out false positives means that we have fewer valid data points to consider and leaving in false negatives means that our data will have some noise in it. The final annotation scheme and instructions are included in Appendix A.

Actions & Constraints	Introduce Actions	Continue Actions	Change Constraints	
	.897 (z=8, p<.01)	.857 (z=27, p<.01)	.881 (z=11, p<.01)	
Discourse Entities	Reference Coreference	Discourse Relations	Properties	Entities to Actions
	.863 (z=19, p<.01)	.819 (z=14, p<.01)	.861 (z=53, p<.01)	.857 (z=16, p<.01)
Other	Informational Relations	Requests for Properties		
	.816 (z=10, p<.01)	.772 (z=5, p<.01)		

Table 5.1: Kappa values for the annotation scheme

5.4 Correlation Results

Using the annotation features we described in the previous section, we are now in a position to test our 6 hypotheses; (1) Domain Constraint Changes (2) Informational Relations (3) Persuasion (4) Commitment (5) Summarization and (6) Verification. We use the χ^2 or Fisher Exact Test to check for correlations between factors.

For each hypothesis we need to use the annotation features to build up or define the higher-level factors in the chi-square tests. To establish which annotation features would best define the higher-level factors in the tests, we initially experimented with just five annotated dialogues. Once the definitions were established we then tested the full set of thirteen annotated dialogues.

In all of the contingency tables, the counts are restricted to utterances in which there is a furniture entity annotated with a coreference relation. In the case of the furniture entities, we examine which of the mutually known properties are expressed in the redescription. In some cases, we look at the property usage relative to the property usage in a previous utterance or turn and in some cases we look for a particular property usage to be related to the contextual situation we are examining.

5.4.1 Domain Constraint Changes Hypothesis

For this hypothesis we want to examine utterances for constraint change annotations in the cases where there is a coreferential relation annotated for at least one of the entities discussed in the utterance. We then compare features of the constraint change to the properties that were expressed for the entity with the coreferential relation.

We annotated the corpus with features indicating whether a constraint change was communicated. When a constraint change was communicated, we also annotated whether this was accomplished implicitly or explicitly. Recall too that we annotated which properties were directly communicated in redescriptions. We can use these particular features to test the hypothesis that there is a difference in property usage when a constraint change is communicated implicitly, explicitly or not at all. We examine each utterance for every constraint change that is generally possible for the domain when populating the cells of the contingency table.

We only count property usage for properties that are arguments in the constraint equations being changed. For example, we only look at the usage of the color property for the color match constraint or price for the constraints having to do with putting an upper limit on the price allowed for an action type argument or setting a subjective filter for the price (e.g. cheap). In Table 5.2, we list each of the constraint types that we examined and the property that we expected would be useful for inferring that change. Our expectations derive from the instructions given to the COCONUT dialogue participants (see Chapter 2). For instance, the instructions ask participants to color-coordinate the rooms. So if we know the colors of the items already selected for a room and the color of an additional item being considered for that room, we can infer whether there has been a change in the setting of the color match constraint. The relevant property for the property limit constraint is indicated in the annotation for the constraint change. We expect that this constraint would rarely happen in conjunction with a redescription since it generally creates a type of search template. For example, (8) applies a color property limit and (9) applies a price property limit. However, in (10), we see that the color property limit could be used in conjunction with redescriptions if we assume that "your green chairs" and "your yellow chairs" are both redescribing entities that have already been presented in the dialogue. Note that with (8) the speaker is not redescribing furniture items; instead he is checking to see if items with a certain subset of features exist. In (9), the speaker could be checking to see if an item he thought existed did indeed. However it seems more likely that he does not know whether the item exists making it similar to (8).

(8) Do you have any green or yellow chairs?

(9) Do you have a sofa for $100?

(10) Let's get either your green or yellow chairs.

Also, we don't expect redescriptions and the price upper limit or the price evaluator constraint to co-occur often. Both are typically found with questions and can create search

Changes	Related Properties
Room Color Limit	color
Price Upper Limit	price
Price Evaluator	price
Property Limit	color, price

Table 5.2: Associated properties and constraint changes

templates. For example, (11) is a case of a price upper limit constraint and (12) and (13) are cases of a price evaluator constraint. However, note that (13) could contain a redescription and still place a price evaluator constraint on the action. That is the price evaluator helps to identify the object that was introduced earlier.

(11) Do you have a sofa that we can afford?

(12) Do you have any cheap yellow or green chairs?

(13) Let's get your cheap chairs.

	Property Used	Property not Used
Implicit change	9	0
Explicit change	2	11
No change	401	649

Table 5.3: Domain Constraint Change Hypothesis: relating property usage to changes

When we examine, implicit changes, explicit changes, and no changes as separate factors, (see Table 5.3), the expected frequency for one cell is too low to assess the results using the standard χ^2 distribution. First, we combine explicit changes and no changes since we expect those to behave similarly and we leave implicit changes as a separate factor.

	Property Used	Property not Used
Implicit change	9	0
Other	403	660

Table 5.4: Domain Constraint Changes Hypothesis: relating property usage to implicit changes

Table 5.4 shows that when we look at redescriptions (annotated as coreference), properties that are related to implicit constraint changes are more likely to be used in the description than not ($\chi^2 = 12, p < 0.001, df = 1$). When there are either explicit constraint changes or no constraint changes, related properties are less likely to be included. This

correlation offers positive support for the hypothesis that a goal to communicate constraint changes makes the inclusion of related properties more likely.

However, since the *Other* category in Table 5.4 is so large, it may skew the results. We also consider only the cases where there was a constraint change. Table 5.5 shows that in the context of an implicit constraint change, properties related to the change are more likely to be used in the description than when the change is explicit (Fisher Exact Test, $p < .0002$). Here we used the Fisher Exact Test [SC88] (p. 103-111) since the expected frequency of one cell in the table is too small for the standard χ^2 distribution to be a good approximation.

	Property Used	Property not Used
Implicit change	9	0
Explicit change	2	11

Table 5.5: Domain Constraint Changes Hypothesis: relating property usage to implicit vs explicit changes

Because there is a correlation between implicit vs explicit constraint changes and property usage, we included this hypothesis in our selection algorithm (see Chapter 6). If the annotated features indicate that the speaker implicitly communicated a constraint change to which the redescription is related, we will include the property that could help with that inference in the redescription. Note that this selection decision does not prove that the change is communicated by including the property. However, if the match with human performance is the same or better for our description selection algorithm than for other description selection algorithms, we will have an indication that such a hypothesis might be true.

5.4.2 Informational Relation Hypothesis

For the Informational Relation hypothesis, we wish to test whether a property value that is made salient by an informational relation associated with an effect, constraint or goal is suppressed in a redescription indicating that it may already be available for identification purposes without having to be explicitly included. In this case, we expect to see that the property is less likely to be used in a redescription of the item whenever the property is made salient via a nearby informational relation. For this test we examined a variety of annotated features, proximity relationships, and other features that could be easily and directly extracted from the dialogues. The features we considered included whether the redescription and the informational relation were issued by the same speaker, whether the

informational relation is before, after or in the same utterance with the redescription, and the type of the informational relation. We combined cells in the contingency table until we came up with one for which the expected frequencies for each cell was 5 or more.[6]

We examined utterances with redescriptions that have informational relations associated with them. The annotations for the informational relations indicate which, if any, object property is made salient by the relation, what the informational relation is and what other utterance is involved in the relationship. Because we have two utterances indicated in each informational relation annotation, we can compute the relative proximity of the utterances to one another (i.e. forward, backward or same). We can also directly extract from the dialogue the speaker for each utterance. We also have information about which action in the utterance carries the information relation, allowing us to match up the informational relation to the appropriate redescriptions in the related utterance. For example, (14) is annotated with an effect informational relation on the select table and select chairs actions and makes the price property salient.

(14) I have the chairs and table.
That will cost us $500.

Table 5.6 shows the final contingency table after combining cells to meet the condition on the expected frequency for each cell. For contingency table 5.6, there is an overall $\chi^2 = 15.5, p < .001, df = 2$. However this only tells us that some of the factors are dependent: it doesn't identify the factors. To find this out, we partition the degrees of freedom ([SC88]pp 194-199). The first partition (forward and backward utterances only) has $\chi^2 = 1.92, p < .2, df = 1$ and the second partition (other and same utterance) has $\chi^2 = 13.58, p < .001, df = 1$. From this, we can see that there is a correlation, but only when the informational relation and the redescription are in the same utterance. This means that in the description selection algorithm, we will suppress the property made salient by an informational relation in the redescription when the informational relation is within the same utterance.

5.4.3 Persuasion Hypothesis

For the Persuasion hypothesis, we wish to test whether expressing a property in a redescription is related to whether the problem solving situation shows the item to be in

[6]Using an expected frequency of 5 is a general guideline for deciding whether the χ^2 distribution will be a good approximation. Authorities differ on how to interpret the expected frequency relative to the size of the contingency table. It appears that the greater the number of rows and columns, the smaller the expected frequency can be. [SC88]

	Salient property Used	Salient property Not Used
relation in forward utterance	11	9
relation in backward utterance	54	21
relation in same utterance	4	14

Table 5.6: Informational Relation Hypothesis: salient property usage relative to proximity of an informational relation

an extreme relationship to some of the alternatives being considered for the solution. By an extreme relationship, we mean that there is a contrast between the furniture items that make the redescribed item more desirable as a solution for an action. For example, the cost of the item being redescribed might be lower than any of the alternatives that have been discussed so far.

Creating a contingency table in this case requires that we interpret some of the annotated features. First of all, we need to identify the cases in which the speaker might be trying to convince the hearer of a particular choice. We also need to approximate what the alternative solutions are at each point in the dialogue.

We base our analysis of what might be a persuasion context on the work described in [DJTM00]. One clear persuasion context arises when an item is presented in a proposal utterance. If other options have been discussed for the action then there may be the possibility for a contrast. A persuasion context could also arise when an item is presented in an unconditional commitment utterance. We must look at what preceded the unconditional commitment utterance since not all are cases of persuasion. [DJTM00] found that unconditional commitments often follow partner decidable options. For example, in (15), A presents his best option and B presents hers. Here B unconditionally commits to A's sofa which is cheaper than her best option.

(15) A: I have a blue sofa for $200.

B: I have a yellow sofa for $250. Let's go with your $200 sofa.

In another pattern, which was not fully analyzed in [DJTM00], the speaker commits to something the partner indicated he did not endorse (i.e. he said something to indicate he thought it was a bad choice). The speaker may feel that the conditions that made it a bad choice are unimportant or are too constraining (i.e. she changes the goals or the constraints of the problem). For example, in (16), B does not endorse the option he presents but A overrides his objection.

(16) A: We have $100 left. I still have that $50 blue chair.

B: I have a rug for $100, but it is yellow.

A: We don't need to match. Let's get your $100 rug.

Finally, an unconditional commitment could also be a persuasion context when lists of items were previously presented. In the COCONUT annotation scheme, these lists are annotated as Statements instead of ActionDirectives. This is because no action is clearly being intentionally addressed [DJP98]. However in our annotation scheme the furniture items are associated with specific actions whenever possible. This is because the items become part of the dialogue participants' shared knowledge and all the items can be used in problem solving and are alternative options for the actions they are related to. For example, in (17), A lists all of the items he has available and so it is annotated as a statement. However B's second utterance is annotated as an unconditional commitment. In this case there were two possibilities for what sofa to select and so a persuasion context arises.

(17) A: I only have 2 red tables for $200,
1 green table for $350 and 4 $50 blue chairs.
I don't have any rugs or lamps
but I have 1 yellow sofa for $200.

B: I have yellow rug for $75 and a blue sofa for $200.
Let's buy your yellow sofa and my rug.

To identify utterances that are either proposals, unendorsed options, unconditional commitments or partner decidable options, we use the definitions that are presented in [DJTM00], which we list below. These definitions use the annotation features described in [DJP98].

- proposal: Action Directive + offer + determinate solution size
- unendorsed option: open option + determinate solution size
- unconditional commitment: Action Directive + commit
- partner decidable option: indeterminate solution size + (open option or (Action Directive + offer))

Once we have identified possible persuasion contexts, we need to check for contrasts with alternatives. We will describe in section 5.5.1 how we track the alternatives for an action. The contrast possibilities are shown in table 5.7. The contrast cases arise from

Contrast	Related Property
Matches other selections when alternatives do not	color
Cheaper than alternatives	price
More expensive than alternatives near end of problem	price

Table 5.7: Associated properties and contrasts between alternative solutions

the COCONUT problem description (see Chapter 2), where color matches, buying all the necessary items and spending out the budget all earn extra points. There is one other possible contrast that we were not able to model accurately by interpreting the features that were recorded in the COCONUT annotation: buying as much as possible. To recognize this kind of contrast we would have to model the problem solving process to see that buying a number of cheaper items for the same total price is an alternative to a smaller number of more expensive items.

For the contingency table 5.8, $\chi^2 = 5, p < .05, df = 1$. We interpret this result as positive evidence in support of the persuasion hypothesis and we incorporate the hypothesis into our description selection algorithm. Any time there is a persuasion context and one of the contrast cases in table 5.7 is present, then the property associated with the contrast will be selected for the redescription.

	Property Not Used	Property Used
no contrast	18	9
contrast	13	24

Table 5.8: Persuasion Hypothesis: property usage relative to shows of contrast to alternatives

5.4.4 Commitment Hypothesis

We wish to test the hypothesis that commitment to a proposed action can be communicated by repeating the properties used in the last description that sets up the expectation for a commitment. As with the persuasion context, we also base our analysis of what constitutes a commitment context on the work described in [DJTM00]. A commitment context occurs when the last description of an item was presented in an utterance in the immediately previous turn and the utterance is identified as a proposal, a partner decidable option or an unconditional commitment and no other items or utterances related to the action intervened. Clearly one likely response that a hearer will make to a proposal or a partner decidable option is to commit to selecting that item. Another pattern found in [DJTM00] is that a partner will sometimes unconditionally commit to his partner's uncon-

ditional commit as part of the agreement process. A final case of a commitment context is when the speaker is unconditionally committing to his own unconditional commitment from a previous turn. In this case, the partner said nothing about the unconditional commitment and the speaker may be trying to verify that his commitment was understood.

When determining repeated properties, we discount the type and owner properties. The type property is excluded because it involves pronominalization and zero anaphora; issues we are not addressing in this research. We exclude the owner property because its only function is identification in this domain.

Table 5.9 indicates that in contexts where a commitment is predicted and occurs, all mutually known properties are more likely to be included in redescriptions (Fisher Exact Test, $p < .0171$).

Given that there is a correlation, we implemented the commitment hypothesis and included it in our selection algorithm. If we are expressing a commitment to an item in a context where a commitment could be expected, then we will repeat the features that were presented in the previous description with the idea that this could communicate our commitment. Note that we are not attempting to confirm that commitment was communicated during interpretation as this will be left for future research (see Chapter 8).

	Not Repeat Properties	Repeat Properties
No Commitment	7	8
Commitment	2	20

Table 5.9: The Commitment Hypothesis: repeating properties from a description in a previous turn vs. level of commitment

5.4.5 Summarization Hypothesis

Recall that the Summarization Hypothesis states that when a problem or subproblem involving the entity to be described has been completed, its redescription will include all of its mutually known properties. To test this hypothesis, we first look at redescriptions after an agreement has been reached for the action. The idea is that the designers are summarizing old decisions to make sure they are coordinated. As part of this process, they want to be sure they both have the right furniture items in their representations of the solution.

As with the Persuasion and Commitment Hypotheses, we base our analysis of what characterizes a summarization context on the work described in [DJTM00]. A summarization context arises when an agreement has been reached on the parameter assignments for

an action, the agreement was reached more than one turn ago and no intervening utterances changed that agreement. We can identify many cases in which an agreement is reached on an action by requiring that either a partner decidable option or a propose utterance occurred more than 2 turns ago. We require more than 2 turns to intervene because we want to allow for the case where the partner left the decision pending by moving on to a dependent action (e.g. a final table decision may be left pending until the chair options are explored). We are estimating that if the action is not revisited after 3 turns, then it was not put on hold pending work on another action and that the partner agreed by moving on to another independent action.[7] This test for agreement is motivated by Clark's types of evidence of understanding; one is the initiation of the relevant next contribution[CS87]. While [CS87] addresses understanding rather than mutual commitment, it appears that the same signals can also address mutual commitment as with the levels of joint actions in conversation discussed in [Cla96]. We can identify other cases where an agreement is reached by requiring an unconditional commitment two or more turns ago. We require that there be an intervening turn so that the partner is able to show that he has moved on to some other problem.

As with the Commitment and Verification Hypotheses, the type and owner properties are excluded when determining whether mutually known properties are repeated. We also observed that the quantity property is not usually used in redescriptions unless the entity has more than one element in it. The contingency table 5.10 shows the relationship between the usage of the quantity property and whether the discourse entity being redescribed is a chair. Chairs are the only discourse entities in the COCONUT domain that typically have more than one element. For table 5.10, $\chi^2 = 90, p < .001, df = 1$. Because of this, we included quantity in the count of mutually known properties only when the number of elements in the entity being redescribed was greater than one.

	Quantity Not Used	Quantity Used
chair	13	32
other	146	8

Table 5.10: Usage of the quantity property

Table 5.11 indicates there is no correlation between a summarization context as we have characterized it and whether all the mutually known properties get repeated ($\chi^2 = 1.49, p < .3, df = 1$). The definition of a summary that we built up from a combination of annotated features and the current dialogue situation as modeled on the basis

[7] In the initial version of our annotation scheme, we had a feature for indicating dependent actions but we dropped the feature because we were unable to get good intercoder reliability measures for it.

of these features may not be accurate enough or the definition may only apply for certain styles of interaction. We test our selection algorithm with and without this hypothesis being incorporated. To implement the hypothesis, we will repeat all the mutually known properties whenever a summarization context arises and there is either no agreement state associated with the current utterance or the agreement state is a statement. Any other agreement state cancels out the agreement for the action.

	All Mutual Properties Used	Not All Mutual Properties Used
Not End of Agreement Process	54	117
End of Agreement Process	8	8

Table 5.11: Summarization Hypothesis: relating end of agreement process to repeating all mutually known properties

5.4.6 Verification Hypothesis

The Verification Hypothesis is that a hearer will repeat all the properties presented in the previous turn where the item was initially introduced, possibly to verify that they were all correctly understood. In this case we collect all the properties that were presented in the turn where the item was first described and check whether the first mention of the item was in the immediately previous turn or further back in the dialogue. As with the Commitment Hypothesis, the type and owner properties are excluded when determining whether properties are repeated.

For contingency table 5.12, $\chi^2 = .06, p < .8, df = 1$. This result shows that we cannot reject the null hypothesis and should assume that the factors tested are independent. It is possible that the verification context we tested was not precise enough and needs more refinement or it may be that the verification function of repetition does not hold for the COCONUT communications setting. We test our selection algorithm with and without this hypothesis to confirm that it does not improve the match to human performance. To implement this hypothesis, we will repeat all the properties presented in an immediately previous turn if the item was initially described there.

5.5 Preparing the Annotated Dialogues for Testing

In order to test our hypotheses, we also need to interpret some of the annotated features in the dialogues to create the discourse entities that are associated with each dialogue. The discourse entities and their location in the discourse structure are also the

	Properties Not All Repeated	Properties All Repeated
not initial in subsequent turn	1	0
initial in subsequent turn	44	28

Table 5.12: Verification Hypothesis: repeating all properties from the turn where last described relative to proximity of last description

input for the context selection and distractor set building algorithms that we will describe in the next chapter. With respect to data for testing the output of the description selection algorithms, we use the annotated furniture entity property features in those same dialogues to determine the desired description that should be generated for an input discourse entity and distractor set. Rather than processing the annotations once to create the higher level information we need for the corpus analysis and the description selection algorithms, we interpret the annotations each time we test our hypotheses or an algorithm. This is because we need to interpret the annotations in slightly different ways for some of the algorithms we test. We could create one file with the desired output but it is not clear where a feature must be expressed in the utterance for it to count in the evaluation of the description selection algorithm. By using the original annotated files for each test case, we can vary what counts as a match in the evaluation process.

5.5.1 Preparing the Input Test Data

The discourse entities that we wish to construct from the dialogue are what results from the utterance having been said in its current context. For each annotated discourse entity in a dialogue, we either create a new discourse entity or update an existing one and link it to a copy of the unmodified entity. We update entities when a coreference relation is indicated in the annotation and otherwise we create a new discourse entity.

When we create a new discourse entity, we start by adding its usage information to the representation; the utterance in which it occurred, how it was expressed and who used it in the dialogue. Next we add the property information that is locally annotated for the current containing utterance. We have three types of property annotations for each entity:

- noun phrase level (e.g. "red" and "chair" in "the red chair"),
- utterance level (e.g. "red" in "the chair is red"),
- locally inferred (e.g. "price" in "I can buy a blue chair with the money we have left.")

To complete the discourse entity, we must next consider any relationships between the new entity and existing discourse entities. There are four discourse entity relations that allow us to infer additional information about the entity we are creating: set, class, predicative and common noun anaphora. These inferential discourse relations are based on the work presented in [Pri81, Kar76, Gro77, Hei82, Haw78, Pas94]. One or more of these relationships may be annotated for a new discourse entity and there is some overlap in the information that they provide for the discourse entity representation we are building.

The set relation captures both subset and set relations (e.g. "3 of the chairs" and "the table and chairs") and class captures instance/type relations (e.g. "I have several red tables. One costs $200 and the other costs $250."). We had few part/whole relations and so we do not discuss this relation further. The common noun anaphora relation captures elliptical, non-coreferential types of relationships (e.g. "I have a red table for $200 and a blue for $300."). The predicative relation "is" indicates an similarity relation between two distinct discourse entities (e.g. "My cheapest table is a blue one for $200."). The two noun phrases create two distinct entities and we treat the more specific entity as the argument entity in our processing.

For each of these relations, property values are inherited. The inheriting entity is the more specific of the two and property values get copied over from the less specific entity when either the inheriting entity has no value for that property or has a less specific value (e.g. "superordinate" is a less specific value for the property type than "chair" is). The inheriting entity becomes the final discourse entity.

All but the predicative relation can involve more than two entities. When multiple entities are involved, we combine the entities and generalize values whenever property values differ (e.g. "red" and "blue" generalize to "range", "sofa" and "table" generalize to "superordinate" and "$50" and "$100" generalize to "$150" when the component entities do not have all the same property values). Inheritance takes place between the combined entity and the entity being created. With common noun anaphora, only the type property is inherited and with the class relation, we further generalize the newly created discourse entity when it was created from several component entities (i.e. we are creating the "type" entity in the instance/type relationship). To generalize a "type" entity, we clear out the property slots with generalized values along with clearing out the quantity slot.

We only update an entity when a coreference relation has been annotated. When a coreference relation is annotated, none of the other types of discourse entity relations should be annotated. When we update the entity, we save a copy of the original entity and link it to the updated entity. By linking a copy of the original entity to the current one,

we can trace how the entity evolved during the course of the dialogue and how and when it was used. Last, we update the usage information and make any changes in the property values that are indicated in the annotation.

To provide the necessary input to the distractor set algorithms, each action in the dialogue has a separate stack onto which we push discourse entities related to that action. If an entity is related to multiple actions a copy of the entity is pushed onto the stack of each of these actions. As we indicated earlier, the actions approximate the discourse segmentation and some of the distractor set algorithms require this as input. The notion of the action representing the non-linguistic task structure is motivated by [Ter85] and is also similar to the conversational threads discussed in [Poe93].

To check the agreement state for the Summarization Hypothesis, we also maintain a separate agreement state stack for each action. An entry in the stack records the agreement state, the speaker, the discourse entities mentioned in the utterance and the turn in which the utterance appeared. We need the discourse entities mentioned in the utterance to check for such things as intervening entities being discussed. When examining intervening entities we must also consider whether there are subsumption relationships with any of the entities in the utterance. For example, when "they" refers to a table and 4 chairs that previously only appeared as two separate discourse entities, we need to be aware that "they" subsumes the entity for "4 chairs" and the entity for "the table" when checking the agreement state of the select chairs action or the select table action.

To compute alternative options for the Commitment Hypothesis, we maintain a separate stack of items discussed for each action. As we encounter utterances that propose or unconditionally commit, we clear the action stack before pushing the items from the current utterance onto the stack. Anything that was initially added to the stack, as well as anything added after a propose or unconditional commit utterance, are considered alternative options for the current action.

5.5.2 Preparing the Output Test Data

We require little interpretation of the annotation features to create the output data we need for evaluating the performance of the description selection algorithms. As we mentioned earlier, there are three categories of property information that are annotated at the entity level, NP-level properties, utterance-level properties and locally inferred properties. The NP-level properties are always included in the representation of what is made explicit for the entity while the locally inferred properties are not.

It is debatable whether the utterance-level properties should count for description selection. Although we annotated whether or not the utterance-level property information is syntactically and semantically required, we have no way of capturing whether the requirements were met opportunistically or not. If we assume that the content for the nominal was selected first, then it is possible that syntax and semantics may have *moved* the property to the utterance level. For example, if the description selection algorithm chose:

```
((owner A)(color red)(type chair))
```

then the syntactic and semantic requirements of "have" would make it unnecessary to also express the owner property value at the nominal level in "I have the $25 red chair that we could use in the LR". However, if the description selection algorithm chose ((color red)(type chair)), the same sentence might result with the owner property value present solely to meet syntactic and semantic requirements. In this last case, we do not know whether the person might further modify the original description selection output in light of the linguistic need to express the owner property. The speaker could conceivably discard a property that was originally selected because the owner property value had the same power for uniquely identifying the target object.

Chapter 6

Description Selection Algorithms

In the previous chapters, we developed six hypotheses about domain and dialogue influences on attribute selection for object redescriptions that would interact with the speaker's ever present goal of the target object being identified by the hearer. Our hypotheses explore the functions of repetition at the argument level within utterances. Previously, these functions have been studied only at the utterance level. Recall that our hypotheses are now:

- DOMAIN CONSTRAINT CHANGES: Properties related to constraint changes are expressed in a context where the change must be inferred by the hearer.
- INFORMATIONAL RELATIONS: When the effect of, goal of, or the constraint imposed on an action is communicated within the same utterance as the action then the properties made salient by such a relation are **not** expressed.
- PERSUASION: Property values that are pivotal for deliberation are expressed in the context of goals to communicate a proposed action.
- COMMITMENT: In the context of a commitment to a proposal, all the properties expressed in the proposal will be repeated.
- SUMMARIZATION: In the context of a previously completed problem or subproblem, all the mutually known properties for an item will be repeated.
- VERIFICATION: In the context of a newly introduced entity, all the properties expressed will be repeated by the hearer in his/her next turn.

Figure 6.1: Intentional Influences hypotheses

All of the hypotheses appeal to the *consequence* repetition function defined in [Wal93] since they strengthen evidence about how inferences are licensed. The first three hypotheses—Domain Constraint Changes, Informational Relations and Persuasion—also appeal to the *attention* communicative functions of repetition defined in [Wal93], since they make propositions salient that may be needed for inferencing. The last three hypotheses—Commitment, Verification and Summarization—also appeal to the *attitude* communicative function of repetition defined in [Wal93] since they help coordinate understanding and acceptance.

In the previous chapter, we performed a corpus analysis both as an initial test of our hypotheses and to provide feedback on our factor definitions. The factors expressed in the hypotheses are combinations of annotation features and also sometimes involve computing approximations of the problem solving situation (e.g. what alternative solutions are currently being entertained).

Rather than attempting to directly annotate for the contextual situations that we hypothesized would be influential, we instead annotated for lower-level indicators. One advantage in doing this is that it seems more difficult to achieve good intercoder reliability for higher-level features than for lower-level ones. We conjecture that this is because more complicated definitions are difficult to apply consistently than are the component definitions. Another advantage of using lower-level features is that we can verify the completeness and consistency of our higher level definitions. Our definitions aren't always as complete and precise as we first think. In addition, the annotations assigned are themselves only approximations of what is going on in the dialogues since the annotators are "overhearers" and cannot test or negotiate their understandings as suggested by [SC89].

To further test our six hypotheses about what influences the content of object redescriptions, we implemented a description selection algorithm that embodies these hypotheses. We are interested in testing two things when checking the output of our description selection algorithm. First, we want to see how well the algorithm matches the human performance in the corpus. We don't expect to achieve a perfect match because there could be many equivalently good descriptions, humans may not always perform optimally, and there may be contextual effects and effects related to general world knowledge that the algorithm doesn't model. Second, we want to see how our algorithm performs relative to algorithms that only consider the identification goal for redescriptions. We compare against our four identification-only algorithms.

Besides comparing our outcome to that of only addressing the identification goal, we want to explore multiple strategies for the identification goal so that we can incorporate

the best performer into our description selection algorithm. In this way, we can see how close we come to matching human performance in addition to simply improving performance. To establish additional benchmarks we also implemented some extreme selection algorithms. These algorithms fix an upper bound on poor performance since we expect them to perform badly relative to the humans in the COCONUT corpus.

In implementing the various algorithms, there are a number of unknowns to contend with. We will expand upon the unknowns as we describe each of the algorithms we used in testing the performance of our algorithm.

In this chapter we will concentrate on describing all the algorithms involved in our tests and in the next chapter we will describe our experimental design and the outcomes of our experiments.

6.1 Addressing the Identification Goal

Recall that in section 3.2 we discussed three cognitively motivated models for object identification:

- gestalt: properties are selected because they make the physical search for the target object more efficient,
- lexical focus: previous descriptions are repeated because they have been established as mutually agreeable ways of identifying the target object,
- incremental minimality: properties are considered according to their domain saliency and are selected only if they rule out distractors.

In this section we will describe our implementations of gestalt and lexical focus and two implementations of incremental minimality, Dale & Reiter's **IDAS** algorithm and a variant that we call **differences**.

6.1.1 Gestalt Selection Algorithm

The idea behind the **Gestalt** selection algorithm is that it is easier for listeners to identify an overspecified referent than a minimally specified one. This is because people are thought to create a *gestalt* search template [Lev89] relative to the object that is to be described. In the COCONUT domain, all the objects have the same attributes so we need only compute a gestalt search template once.

During the corpus analysis, we gathered information about the frequency of property types in descriptions in order to determine what might be a *gestalt* search template for

the COCONUT domain. Type, color and owner are the three properties that were most frequently used in the redescriptions. After trying a variety of property groupings, we found that type and color were the best choices for a gestalt template in that they provided the closest match to the human data for this selection strategy. See chapter 7 for the details of this particular performance test.

Since the idea of the gestalt search template says nothing about uniquely picking the target object out of a distractor set, we tried several approaches for dealing with adequacy and selected the one that produced the best performance results.

6.1.2 Lexical Focus Selection Algorithm

Passonneau describes three possible sources of attribute saliency [Pas96]:

- lexical focus: the attributes in the last mention of the discourse entity are salient,
- location focus: the location attribute in the most recently mentioned location is salient,
- event focus: the attributes of the last key event in which the object participated are salient.

Passonneau suggests that the event and location focus categories are possibly genre dependent whereas lexical focus is not. She assumes that the attributes indicated by the three sources are all equally salient and that the salient attributes for an entity are updated as the discourse progresses.

Event focus is not relevant for the COCONUT domain since there are no distinguishable differences between the select actions that could help distinguish furniture items. Although location is relevant (i.e. what room does the furniture item go in), it was generally part of the action description (e.g. "let's put the rug in the living room"), wasn't generally changed (e.g. "let's move the chair in the dining room to the living room") and wasn't generally used to identify a furniture item (e.g. "the rug in the living room is yellow and the one in the dining room is red."). This means that we only have lexical focus to indicate what attributes are salient.

Passoneau's algorithm enforces economy rather than efficiency (as with **IDAS**). It uses a generate-and-test method to arrive at a distinguishing description. It tries three approaches and if the first fails to produce a distinguishing description then the next approach is tried. The three approaches for what to express about the entity are to generate:

1. a pronoun,

2. a minimal NP with just the type and a determiner,

3. a full NP with salient attributes added to the minimal NP.

The algorithm does not indicate how to select a particular salient attribute. We are only concerned with lexical focus for COCONUT and it is not clear what to do if the previous mention includes multiple attributes (in addition to type). Given the rationale behind lexical focus, in cases where the attributes in the last description of the entity are in focus and are possibly an agreeable way to label the object, we would expect all of these attributes to be of equal saliency and they should be used as a group.

The algorithm also does not say what to do with the object attributes that are not salient. We assume that the other attributes are available to help select the target object if the lexical focus attributes do not uniquely select the object. As with **gestalt** we tried several approaches for dealing with adequacy and selected the one that produced the best performance results.

6.1.3 Dale & Reiter's IDAS Algorithm

Dale & Reiter's **IDAS** algorithm [DR95] is the next generation for the algorithms that they developed in their respective dissertations [Dal92, Rei90a]. Instead of trying to find a minimal distinguishing description which is NP-hard, the algorithm relies on a prioritized list of attributes. They envision a customized attribute list for each domain and assume for now that the list does not change during a dialogue. The algorithm works through the list and adds each attribute that eliminates an object from the distractor set. As distractors are eliminated, the distractor set is modified to reflect this. This algorithm is a more computationally plausible approach and because of this perhaps also more psychologically plausible.

IDAS checks an ordered list of properties and chooses to include the property if it rules out any of the distractors in the focus set. The list of properties is ordered by perceptual saliency and, as we said earlier, we expect this saliency to be task specific. While it is believed that the perceptual saliency may change as the task progresses, it is generally taken as a simplifying assumption in most implementations that this saliency remains static for the entire dialogue. By examining the frequencies with which these properties are used in the corpus, we determined the perceptual saliency hierarchy for the COCONUT domain to be: type, color, owner, price and quantity.[1]

[1] We assume that the discourse entities are collectives or plurals with homogeneous properties. When non homogeneous property values are indicated by set discourse entity annotations, we generalize the property values (e.g. "ours" for owner and "range" for non homogeneous colors and prices) to create the new discourse entity. However, these generalized property values are never used for coreference in the COCONUT dialogues.

This algorithm assumes that a distractor set is provided as input. The definition of the distractor set will be discussed in section 6.1.5.

6.1.4 Adjusting Perceptual Saliency for the Current Focus

We also tried a variant of **IDAS** that we call **differences**. It allows us to experiment with a simple way of having the context of the dialogue influence perceptual saliency.

Differences is similar to **IDAS** in that it uses an ordered list of properties, but the selection criterion is different. **Differences** selects the property if the target object's property value is different from the most salient value for this property in the distractor set. The first few selections of attributes for describing an entity will be just like **IDAS** but overall it is more likely to include more properties than **IDAS** since prominence for each property is based upon the entire distractor set.

The **differences** selection criterion is meant to capture the idea of perceptual prominence observed in experiments by [CSB83]. The experimenters showed one of two photographs to students and asked them to describe the color of the flowers in the photograph. In the two photographs there were four sets of flowers and each set was of a different color. In one photograph the daffodils were made more prominent and in the other nothing was prominent. With the first photograph, the students more frequently described the color of the daffodils and in the other they more frequently asked which flowers the experimenter meant. In the first photograph the daffodils stood out or were somehow different from the others and so they didn't need a unique identifier.

Differences also relates to the idea discussed in [Lev89], which contrasts the object with the last item in focus. It is similar in that it is working with differences or contrasts. What distinguishes the two ideas is what is being contrasted; either the entire focus space (as with **differences**) or the most salient entity in the space.

For the **differences** algorithm we find out what property values are perceptually prominent in the distractor set and then contrast the target object with those prominent values. If any of the target object's properties are different from the prominent values then those properties are included. IDAS selections, on the other hand, are influenced only by what is currently in the distractor set. A property that contrasts with a prominent one may not get selected because all the distractors with that prominent property have already been eliminated. Prominence is based on counts of property values in the distractor set. If most of the objects are "red" then it is a prominent value. We experimented with a threshold for what percentage of values are needed for a value to be called prominent. We found that a

threshold of .45-.65 provided the best results. We will discuss this experimentation in the next chapter.

Like **IDAS**, this algorithm assumes that a distractor set is provided as input. The definition of the distractor set will be discussed in section 6.1.5.

6.1.5 Distractor Set Definitions

It is widely believed that some partitioning of a discourse defines the distractors from which the target object must be singled out ([Hei83, Rei85, PS84, GS86, Kam93] *inter alia*). As [Dal92] points out, there are many unanswered questions about how the discourse should be partitioned and how subsequent reference relates to the partitioning. In particular, once one settles on a partioning scheme two questions arise:

- What is the saliency between partitions?
- Are all the distractors in a partition of equal saliency?

Since the distractor set definition is an unknown we tried a variety of definitions with the above questions in mind. We tried two extreme approaches, one that is between the two extremes and two that are related to Grosz & Sidner's theory of the attentional and intentional structure of discourse [GS86]. We briefly introduced these distractor set definitions in section 4.1, but we will describe them in more detail here.

The first extreme, which we called **ALL**, includes every furniture discourse entity that has been mentioned thus far. This is similar to the global working set approach used in [Dal92] for the domain of generating cookbook recipes. The global working set at a particular point in time contains all the identifiably distinct ingredients. Initially the set contains all the ingredients that are listed at the start of a recipe and then, as actions are described, the ingredients in the set are updated. For example, if the initial list contains flour, sugar and milk and an instruction indicates that they be mixed together, then the three ingredients will be removed from the working set and batter will be added. For the COCONUT domain, there are no actions that change the furniture items but the items are incrementally introduced.

The second extreme includes only the discourse entities that were discussed in the previous utterance. We call this definition, **1UTT**. This definition assumes that only the most recently mentioned entities are salient regardless of the partitioning of the discourse. If we wanted to allow for partitioning, then there would be no salient distractors at the start of a new partition.

Between the extremes of **ALL** and **1UTT** we have **5UTT**. It places all of the entities discussed in the last five utterances in the distractor set. Like **1UTT** and **ALL** it assumes that there is no partitioning of the discourse but that the relevant distractor set lies somewhere between those of **ALL** and **1UTT**.

The first distractor set definition, based on Grosz & Sidner's theory is, **SEG**, which we described earlier in Chapter 4. It attempts to duplicate Passonneau's distractor set definition in that it is the union of all the entities described in the current segment and all the entities for the related action when that action and entity were last described. This assumes that intervening segments are no longer accessible, and that the saliency between entities in a segment and across the two most accessible segments are equal.

The second definition that is based on Grosz & Sidner, we call **SEG+SOLN**. It is the union of all the distractors in the current segment and the discourse entities that are thought to be in the solution set for the problem at that point in the dialogue. The solution set is heuristically determined by assuming that if one of the dialogue participants stops talking about an action then it has probably been solved. It assumes that of all the previous utterances we only retain accessibility to the most highly salient entities described by them. If the entities are part of the solution set then we assume that they would always be accessible. The only time the solution set would provide additional distractors that influence the attributes selected (since our redescriptions always include the type) is in the case of backtracking to change the solution.

SEG and **SEG+SOLN** also require that we define a partitioning of the dialogue. In sections 5.1 and 5.5.1 we described how we defined this segmentation. Instead of marking segment boundaries we push copies of discourse entities onto stacks for each of the actions to which they are related. The focus space is then the union of all the action stacks for each action mentioned in the current utterance. If no actions are discussed in an utterance, the focus space is carried over to the next utterance. When a previously discussed action is not continued in the next utterance then the stack for that action becomes the focus space for its previous segment.

6.2 Extreme Selection Algorithms

The first extreme algorithm that we will examine is **random**. **Random** loops for the maximum number of property choices that can be made and randomly selects a property value to include regardless of whether the value for that property is known. This tends to skew the number of property values selected to a smaller number and is more like the distributions found in the corpus. Only 4% of the redescriptions use all the properties.

The other two extreme selection algorithms that we tested are, **always** and **never**. **Always** selects all the mutually known properties for every entity to be described and **never** always chooses to express just the object type.

6.3 Intentional Influences Algorithm

The intentional influences algorithm first determines whether the addition of particular properties might help contribute to the satisfaction of goals other than unique identification. For example, if adding a property could help the hearer infer that a constraint change is intended then that property should be included to help satisfy the goal of achieving joint commitment to a constraint change. Once these other goals have been addressed then the goal of unique identification is addressed. This means that prior to a call to satisfy an identification goal the distractor set may have already been reduced to some extent and the attributes already included in the description will have been removed from the prioritized list of attributes. (These details assume that **IDAS** is the best algorithm for trying to satisfy the identification goal.)

6.3.1 Integrating Intentional Influences with the Identification Goal

Our hypotheses in the previous chapter are independent of one another and so were their contingency tables. However to create a selection algorithm, we need to combine the goals expressed in our hypotheses and include the identification goal when appropriate. Recall that we do not have a single annotation to indicate what goals the speaker might have in mind. Instead we use a combination of annotation features and the discourse entities we created to infer that we are in a context where a particular goal might arise.

To decide how to combine the goals, we reasoned that summarization, commitment and verification were independent of identification since they cause either all the mutually known properties to be expressed or they cause all properties from an immediately previous description to be repeated. In both of these selection cases, there should be no confusion about what the target object is and so we do not separately address the identification goal when one of these three goals is active. Likewise, we should be able to treat each of these goals as being independent of one another. We assume that these three goals do not co-occur, although they could. For example, one might want to verify her understanding of a description and at the same time commit to selecting the object (although it would be contingent upon the correct understanding). However, the goals have nearly the same outcomes for description selection. So, we test for these goal contexts separately and we order the context checking by considering the most constrained context first. We test first

for the summarization context, then the commitment context and finally the verification context. If none of these contexts apply then we test for the persuasion and domain constraint change contexts since neither of these should co-occur with the other three contexts we just discussed.

We jointly test for the persuasion and domain constraint change contexts since we reason that they could co-occur. Finally we follow-up the selection decisions from the persuasion and domain constraint change context checks with an attempt to satisfy the identification goal. If the description does not pick out the target, then we add more properties to try to make it so. At the end of our description selection decisions we use the informational relations context to suppress properties when appropriate.

Given our findings about the usage of quantity in section 5.4.5 (i.e. quantity usage correlates with the object being a chair), we start the entire selection process by including the quantity whenever the type of the object being described is a chair and the quantity is greater than 1. We also always include the type. Another preprocessing step includes a check for objects that participate in information requests and responses. In these cases, a specific property is requested by the speaker. We assume that the speaker answers by repeating all the properties from the request along with the requested property. In this case, none of the other context checks are tested or applied.

Note that we could test for each context every time, but by reasoning about what may be independent we save some effort at run time. Below, in Figure 6.2, we show the general algorithm as described above. Recall that although Verification and Summarization were not supported by the correlational study, we have parameterized their inclusion. The parameterization is not reflected in the algorithm description we present here. Next we will describe each of the algorithms needed for deciding whether a particular goal context exists.

We should note that we do not advocate using this algorithm as it is stated in a generation application. The algorithm is oriented towards determining what goals are influential rather than providing a general mechanism for combining the influence of multiple goals on description selection.

6.3.2 Context Checking

We need to check for the following contexts: summarize, commitment, verification, persuasion, constraint change and informational relation. To understand these context tests, recall that the following states are associated with utterances and the actions they express, base on a combination of annotation features:

- partner decidable option: The speaker describes or offers an item but the current solution state is indeterminate because the speaker believes the hearer may have better options that he has not yet presented.
- propose: The speaker offers an item in a context in which he has already heard what the hearer has to contribute as an alternative.
- unendorsed option: The speaker offers an item in a context in which he has already heard what the hearer has to contribute as an alternative and he believes the alternative is better.
- unconditional commit: The speaker indicates a commitment to using an item.

The summarize context is the most restrictive of the six. An agreement must have been reached for the action without the action being readdressed in between the agreement and the summarization. The achievement of an agreement state is approximated when a propose or partner-decidable option was the last state for the action and it happened more than two turns ago or an unconditional-commit was the last state and it happened two or more turns ago. In a propose or partner-decidable context, the agreement gets inferred and in the other case the agreement is more explicit. This is an approximation of the summarization context since there may be other patterns for establishing an agreement that we do not yet know about or cannot capture with the annotations schemes we used. For example, we cannot deal with contingencies between actions. If a contingent decision allows one to infer a new state for a related action, we will miss the outcome of the inference with our context check. For example in Figure 6.3, the participants are considering a yellow and green table of equal cost. They defer their final decision until they consider the chairs that are available. Since they decide on the green chairs and the last state for the table action was an unendorsed option, we will miss the agreement about the table unless the participants explicitly return to the table decision.

The commitment context requires that there be a previous proposal or unconditional commitment for the entity in the immediately previous turn. For this to be so, no other items must have been discussed for the action in between the current utterance and the proposal or unconditional commitment. Also we allow a speaker to unconditionally commit to his own previous unconditional commitment as long as it was in his immediately previous turn.

The persuasion context arises when a proposal is to be made and alternate solutions exist where there is a contrast between the colors or prices that make the proposed item clearly a better choice. The times when a proposal is to be made can only be approximated

since we do not know with certainty what the speaker had in mind. Given the analysis of the agreement process in [DJTM00], we look for either a propose utterance, or an unconditional commitment utterance where the previous state was an unendorsed option or a partner decidable option. The alternatives are approximated by saving the items discussed for each action. After a propose or unconditional commitment are made then all the items for the action get flushed and the proposed item gets added. Next we must check for contrasts. For color we compare the color of the proposed item to those items already selected for the room and the alternate items. If the proposed item matches the items in the room and none of the alternates match them, then we have a persuasion context. For prices we have two possibilities. If we are near the end of the problem solving effort and the price of the proposed item is greater than that of all the alternate items then we have a persuasion context. In this case the item may be helping to spend out the budget. If the price of the proposed item is less than that of each of the alternate items this is another persuasion context. Here, the cheaper item may be preferred to help reserve money for other purchases.

The remaining contexts are simpler to recognize. The verification context occurs when an item was described in the immediately previous turn. The constraint change context is actually signaled by a single annotation feature. If there is an implicit constraint change annotated then we have a constraint change context. The informational relation context is signaled by an informational relation annotated as applying to itself. For example, in (18), the action and effect are expressed in the same utterance unit.

(18) Getting the green chair will spend the rest of our money.

In the next chapter we will describe our experimental design for evaluating these algorithms and the performance results.

```
Select the type of the object

If the description is an answer to an inforequest on a property for
     a mutually known object
then repeat all the properties from the request
     include the requested property

ElseIf the summarize context applies
then select all the mutually known properties

ElseIf the commitment context applies
then select all the properties from the offering utterance

ElseIf the verification context applies
then select all the properties expressed in the previous turn

Else
   If the persuasion context applies
   then select the properties that make the item a better solution

   If the constraint change context applies
   then select the properties that can be used to infer the change

   If the description doesn't uniquely select the target object
   then apply an identification oriented selection algorithm

If the informational relation context applies
then deselect the properties made salient by the relation
```

Figure 6.2: Intentional Influences algorithm

```
G>
[15]: I have table-high for $200 green.
[16]: Do you have any cheaper?

S>
[17]: The sofa I bought is blue,
[18]: I messed you up before.
[19]: The cheapest table high I have is the same $200
[20]: and is yellow.
[21]: Who should buy the table high?

G>
[22]: Do you have any cheap yellow or green chairs?

S>
[23]: Yes, I have 3 green chairs
[24]: that cost $50 each.
[25]: Do you want me to buy some?

G>
[26]:  yes buy 3 green chairs for $50 each
```

Figure 6.3: Contingencies between actions

Chapter 7

Evaluating the Performance of the Description Selection Algorithms

In the previous chapter we developed a new description selection algorithm; the **intentional influences** algorithm. It incorporates our hypotheses about how intentions related to domain problem solving and how intentions related to the dialogue can influence the content of object redescriptions. These hypotheses are based on the functions of repetition that have been identified and studied at the propositional level. The difference with our work is that we are associating the repetition functions with the argument level as well. Our goal in this chapter is to evaluate the performance of **intentional influences** relative to the performance of the human dialogue participants in the COCONUT corpus and to the algorithms that address only the identification goal.

7.1 The Experimental Design

There are two levels of experiments we used to evaluate **intentional influences**; one that explores the unknowns internal to each of the algorithms we used in the evaluation and one that compares description selection algorithms. In all of our explorations and comparisons we use the same response variable: the match to human performance.

7.1.1 Defining the Response Variable

To compare the performance of the algorithms to that of humans, we use a measure of the degree of match between the human's and the algorithm's selection of properties for the same discourse entity in the same dialogue context. Inclusion and exclusion of a property both count in the degree of match. We only consider four of the five properties associated with a furniture entity in the COCONUT domain. These properties are; *color*, *price*, *owner*, *quantity*. We exclude the property *type* from the measure since it is generally assumed to be represented in the expression and since we are not studying the question of when to use

pronouns and zero anaphors. So a perfect match means that the algorithm chose to include or exclude the same properties as the human did for a particular entity. The measure is calculated with X/N, where X is the number of attribute inclusions and exclusions that agree with the human data and N is the number of attributes that can be expressed for an entity ($N = 4$ for COCONUT). This means the value of the measure ranges between 0 and 1 inclusive. This response variable is called *match* in the experiments that follow.

7.1.2 Statistical Tools

We first do an analysis of variance (ANOVA) [Mat98, Coh95] on the results of the experiments to find out if there are any significant differences in the performance as we vary the experimental factors. However, this only tells us whether there are significant performance differences. To determine where and how large the performance differences are, we do multiple pairwise comparisons (MCA) [Hsu96] and at the end of the chapter we use all-to-one comparisons, sometimes called multiple comparisons with a control (MCC) [Dun64] [1]. The results of the multiple comparisons will be displayed as 95% confidence intervals, which can be displayed in graphs like Figure 7.1. These intervals are always of the form:

(estimate)±(critical point)×(standard error of estimate)

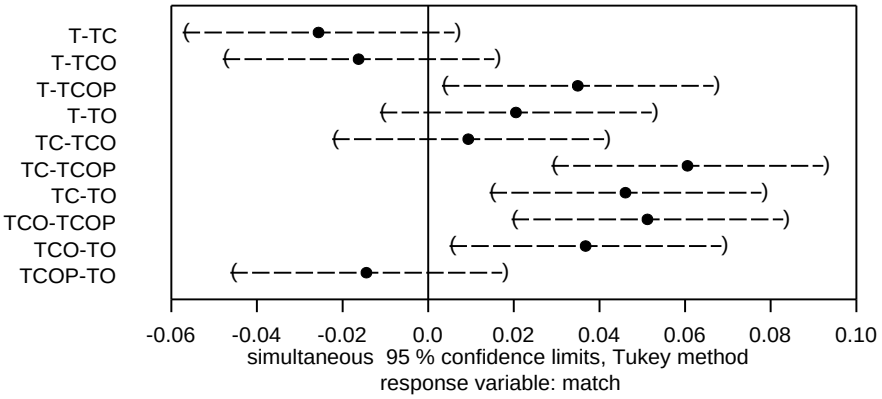

Figure 7.1: Example of confidence intervals

The critical point depends on the multiple comparison method that is specified (e.g. Tukey, Dunnett, Bonferroni). In the comparisons we report here, we chose the method

[1] We used the Splus4.5 multicomp function to perform the multiple comparisons [Mat98].

that created the smallest critical point. For this reason, the methods indicated in the graphs may vary some [2].

Intervals in the graphs that exclude zero indicate statistically significant performance differences. (Note: it doesn't follow automatically that the performances of two algorithms are identical if there is no significant difference between them. It is a matter of judgement to determine whether to treat them as identical.) The labels on the y axis indicate the two levels or experimental factors that were compared. If the interval is to the right of zero then the first member of the label pair performed better and if the interval is to the left then the second member of the pair performed better. In Figure 7.1, the following pair differences are significant: T-TCOP, TC-TCOP, TC-TO, TCO-TCOP and TCO-TO.

We frequently discuss non-significant differences in terms of performance trends as part of making a judgement call about equality of performance. If the center point of the interval is to the right of zero then we say that the first member of the pair in the label has a trend towards performing better, and vice versa if the center point is to the left of zero. However, we must take into consideration the size of the interval in order to judge whether the opposite conclusion may be true or whether the two performances may be equivalent. In cases where we must make a decision for setting an internal algorithm factor we are lax in this judgement but for comparing algorithms we are conservative about judging whether there is a trend towards better performance.

7.2 Exploring the Unknowns within the Algorithms

To evaluate **intentional influences**, we compared its performance to that of the description selection algorithms that only try to satisfy the identification goal. In the previous chapter, we discussed a number of problems with implementing the identification-oriented algorithms. One problem that is common to all these algorithms is their dependence on a notion of saliency. An appropriate way of determining saliency has not yet been thoroughly explored and decided by the research in computational linguistics or cognitive psychology. This means that there are factors that we can vary within each of the algorithms and our goal is to coax the best possible performance out of each algorithm before comparing the algorithms to one another. We did not vary the features or their values exhaustively when we conducted our experiments. Instead we focused on the higher level issues and tried variants that have been suggested in the literature.

[2] The Splus4.5 multicomp function has an option to select the best method and so automatically finds the smallest critical point by considering all the valid methods.

Recall the identification-oriented selection algorithms that we described in the previous chapter:

- **Gestalt**: properties are selected because they make the physical search for the target object more efficient.
- **Lexical focus**: previous descriptions are repeated because they have been established as mutually agreeable ways of identifying the object.
- **IDAS**: properties of the greatest saliency for the domain are chosen on the basis of whether the target object's property value will uniquely select it from a set of distractors.
- **Differences**: properties of the greatest saliency in the domain are chosen on the basis of whether the target object's value will either contrast with the most salient value in the space or uniquely select the target object from a set of distractors.

For each of these algorithms, we will describe the unknowns that we experimentally decided.

For the gestalt algorithm, we didn't know what the search template should be. Although there are studies about perceptual saliency, there are no similar studies for abstract or imagined objects. Although a new template should be created for each item according to the ideas expressed in [Lev89], for COCONUT we assumed that we could use the same template for the entire corpus. We believe this assumption is valid because the furniture objects all have the same attributes. As a result, we needed to establish one template that was appropriate for COCONUT. Recall that the search template equates to the description selected by the **gestalt** algorithm.

We also didn't know what to do when the gestalt description fails to uniquely identify the target object. So we treated the question of adequacy as an unknown for **gestalt**. It was also an unknown for the **lexical focus** algorithm.

For the **differences** algorithm, the open questions are how to determine what the most salient properties and values are for the current distractor set. We did not experiment with different property saliency orderings. Instead we used the same ordering as for **IDAS**. Recall from 6.1.3 that the property frequency distributions found in the corpus gave us this saliency ordering: type, color, owner, price, quantity. We did, however, experimentally determine how to identify salient property values. In this case we used a threshold to indicate how many objects in the distractor set must have the value for the value to be considered salient. For example, how many more red objects than green, blue or yellow are needed to make "red" the most salient value for the color attribute? It is this saliency threshold that we experimentally determined.

Finally, for **IDAS** as well as all the algorithms, we didn't know how to determine an appropriate distractor set. We varied the distractor set definition to find out which offers the best performance for each of the algorithms.

For the **intentional influences** algorithm we needed to determine which of our hypotheses (see Figure 6.1) to actually include in it. Although we had guidance from the correlational studies we described in Chapter 5, this sort of testing does not allow us to infer causality between the factors we studied. Since all but two of our hypotheses were supported, we simplified our testing by either including and excluding only the unsupported hypotheses: Summarization and Verification. We always included all the rest of the hypotheses.

In addition, we didn't know what to do when the selections did not uniquely identify the target object. In one case, none of the contexts associated with the hypotheses may apply and in the other the selections made by the applicable goals may not be enough to uniquely identify the target object.

7.2.1 The Experimental Factors in the Identification Algorithms

We will first describe the factors that we varied for the collection of experiments on the identification algorithms. These factors include:

- distractor set (any algorithm that incorporates **IDAS**)
- search template (**gestalt** only)
- adequacy (**gestalt**, **lexical focus**)
- saliency threshold (**differences** only)

We will describe each of the factors separately from the results of the experiments since we varied multiple factors in many of the experiments but then examined them separately during the analyses. After each factor value we will list in parentheses the abbreviation we use for it in the experiments and the confidence interval labeling.

The distractor set factor varies which distractor set definition we use. Recall from 6.1.5 that we have these five distractor set definitions:

- **ALL**: all entities previously mentioned in the discourse (ALL)
- **1UTT**: all entities mentioned in the previous utterance (1UTT)
- **SEG+SOLN**: all entities in the current discourse segment and all the entities currently in the solution set for the action being addressed (SEG+)

- **SEG**: the union of the entities in the current discourse segment and the last discourse segment in which the target object occurred (SEG)
- **5UTT**: all the entities mentioned in the previous 5 utterances (5UTT)

The search template factor applies only to the **gestalt** algorithm. We tried the following search templates out of all possibilities because they are the attribute sets that were more frequently used in object redescriptions and also include one or more of the three most frequently used properties (type, color, owner):

1. Type Color (TC)
2. Type Color Owner (TCO)
3. Type (T)
4. Type Owner (TO)
5. Type Color Owner Price (TCOP)

The adequacy factor varies what we do when the algorithm's selection does not uniquely select the target object from the distractor set. This problem was not clearly addressed in the literature that suggested the **gestalt** and **lexical focus** algorithms. It could be that the selection should be abandoned when it is inadequate in favor of using some other selection approach or that the selection should have additional attributes added in order to make the description adequate. We tried three possible values for the adequacy factor:

- Use the main algorithm's unaltered description without considering unique identification (GEST, LEX)
- Use the **IDAS** algorithm to supplement the main algorithm's description with additional attributes in an attempt to make the description adequate (GEST+, LEX+)
- Substitute **IDAS**' description whenever the main algorithm's description is inadequate (GESTOR, LEXOR)

The saliency threshold factor applies only to the **differences** algorithm. The threshold helps to determine whether there is a salient value for a particular attribute in a given distractor set. A particular setting represents what percentage of objects must have the attribute value for that value to be considered salient. We tried the following settings in our experiments:

- .2 (LOW)
- .45 (MEDL)
- .65 (MEDH)
- .9 (HIGH)

7.2.2 Determining the Unknowns for Gestalt

7.2.2.1 The Distractor Set Definition

For both GEST+ and GESTOR we need distractor set definitions in order to judge adequacy but this does not apply for GEST. Because of this we experimented with the five values for the distractor set definition (1UTT, 5UTT, SEG, SEG+, ALL) for only GEST+ and GESTOR.

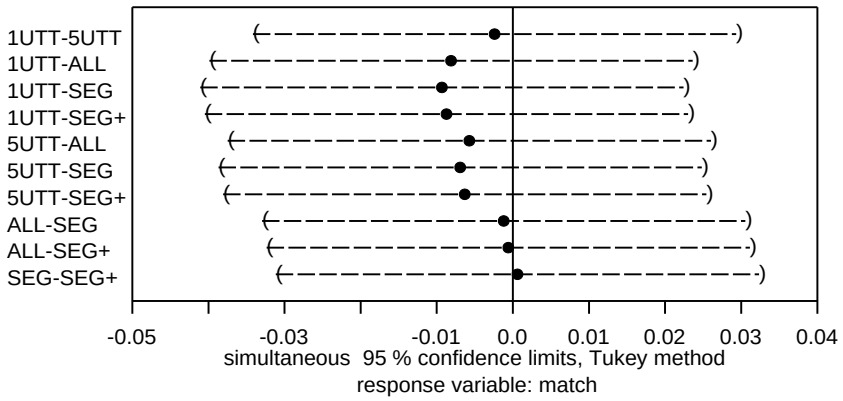

Figure 7.2: Multiple comparisons of distractor set for Gest+

For GEST+, an ANOVA of ($F = .265, p < .9$) shows that there are no significant performance differences between the five distractor set definitions. However, since the MCA comparison shown in Figure 7.2 indicates there is a trend for SEG to perform slightly better[3], we used SEG as our distractor set definition for the GEST+ setting of the algorithm.

For GESTOR, an ANOVA of ($F = .465, p < .761$) also shows that there are no significant performance differences between the distractor set definitions. This time the MCA comparison shown in Figure 7.3 indicates that ALL has a slight trend towards better performance. We used ALL as the distractor set definition for the GESTOR setting of the **gestalt** algorithm.

[3]To see that this is so, note that the median point that is furtherest away from zero is a pair involving SEG and also that SEG is the better performer in all its pairwise comparisons.

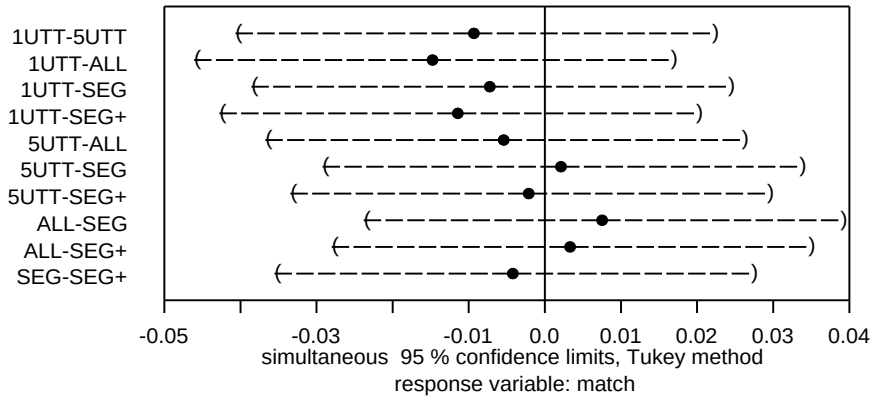

Figure 7.3: Multiple comparisons of distractor set for GestOR

7.2.2.2 The Gestalt Search Template

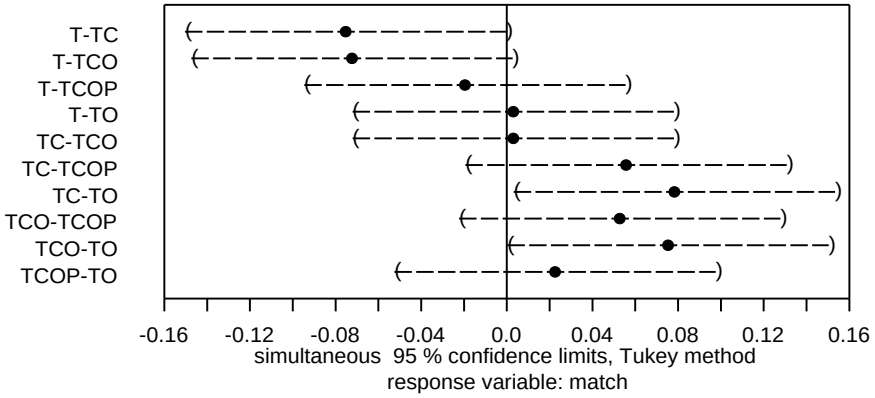

Figure 7.4: Multiple comparisons of gestalt search templates

In conjunction with the above experiment for the distractor set definition we also varied the gestalt search template setting in order to find out which of the search templates provided the best performance (TC, TCO, T, TO, TCOP).

ANOVAs of the results of the experiment showed that there were significant performance differences for all the variants of **gestalt** adequacy as we varied the search template. For GEST ($F = 3.9, p < .0037$), GEST+ ($F = 9.32, p < .0000002$), and GESTOR ($F = 6.18, p < .00006$). From the MCA comparisons we get the confidence intervals shown in figures 7.4, 7.5, and 7.6. We see that template TC is significantly better than TO for GEST, and TCOP and TO for GEST+ and GESTOR. The TC template also has a trend towards better performance compared to the remaining templates. In our other experiments we used type and color (TC) as the setting for the GEST, GEST+ and GESTOR templates.

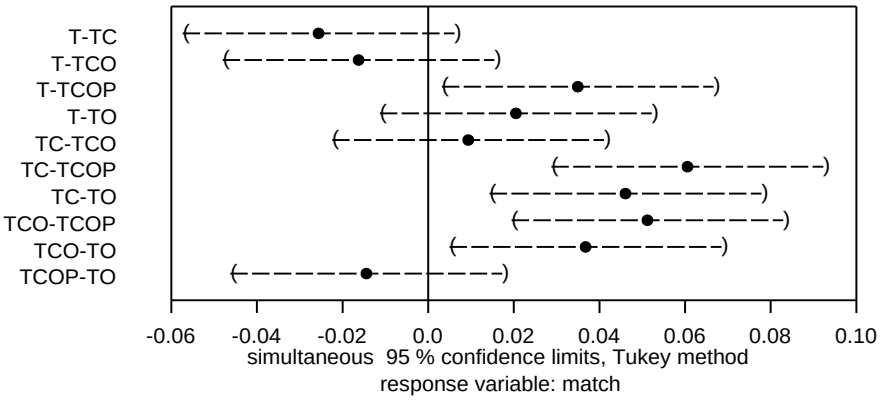

Figure 7.5: Multiple comparisons of gestalt+ search templates

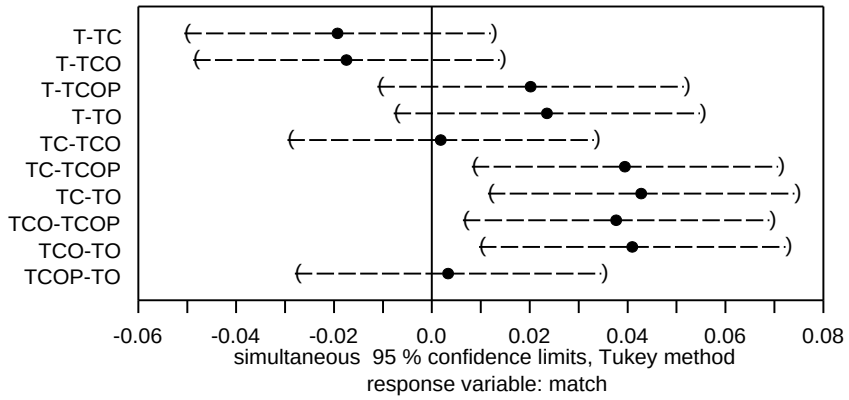

Figure 7.6: Multiple comparisons of gestalt or IDAS search templates

7.2.2.3 Adequacy

For this experiment, we used the best settings for the distractor set and the template as indicated above when we considered the influences of varying adequacy (GEST, GEST+ and GESTOR). The ANOVA for this experiment showed that there were no significant performance differences between the three ($F = .123, p < .88$). Since we wanted to make the best choice possible for this setting we did an MCA comparison to see which setting has a trend towards better performance. The results of the MCA comparison are shown in Figure 7.7. From this we can see that GEST+ has a trend towards being the better performer of the three. We used this adequacy setting for the **gestalt** approach. This means it uses the IDAS approach to add additional attributes to its initial description (the template) until it finds a description that uniquely identifies the target object.

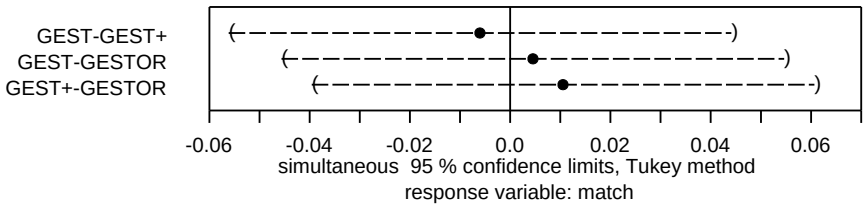

Figure 7.7: Multiple comparisons of gestalt, gestalt with unique identification and gestalt or unique identification

7.2.3 Establishing the Unknowns for Lexical Focus

7.2.3.1 Distractor Set Definitions

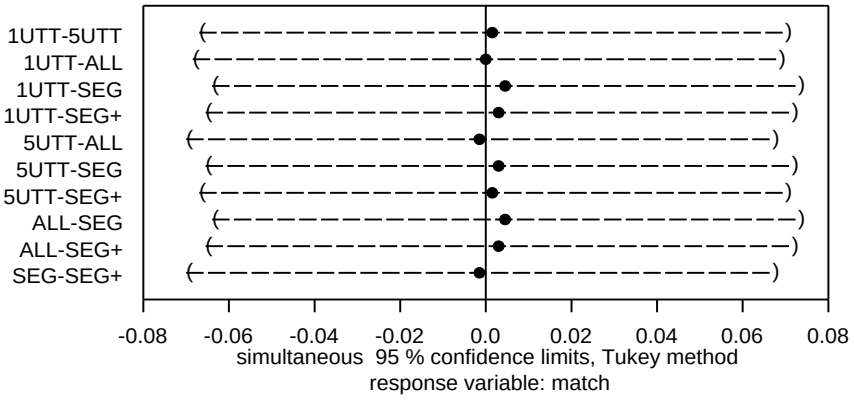

Figure 7.8: Multiple comparisons of distractor set for Lex+

We can disregard the distractor set definition for the base level **lexical focus** algorithm LEX since it does not consider adequacy, but we cannot for the other two; LEX+ and LEXOR. Both of these need a distractor set definition to make a judgement on adequacy. For LEX+, our ANOVA result of $(F = .0123, p < .9997)$ shows that there are no significance performance differences between the distractor set definitions. From the MCA comparison results shown in Figure 7.8 we can see that 1UTT and ALL appear to be equivalent and have a small trend towards performing better than the others. For the LEX+ algorithm we arbitrarily picked ALL over 1UTT as the distractor set definition. For LEXOR, our ANOVA result of $(F = .129, p < .972)$ again shows that there are no significance performance differences between the distractor set definitions. From the MCA comparison results shown in Figure 7.9 we can see that ALL has a trend towards performing better than the others. For the LEXOR algorithm we used ALL as the distractor set definition as well.

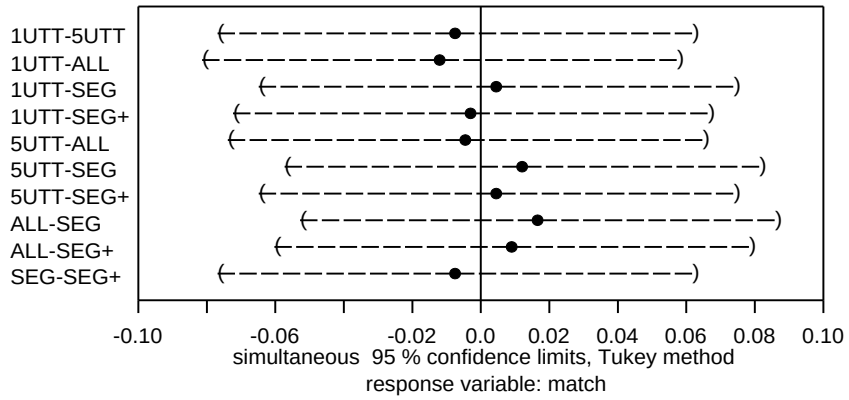

Figure 7.9: Multiple comparisons of distractor set for LexOR

7.2.3.2 Adequacy

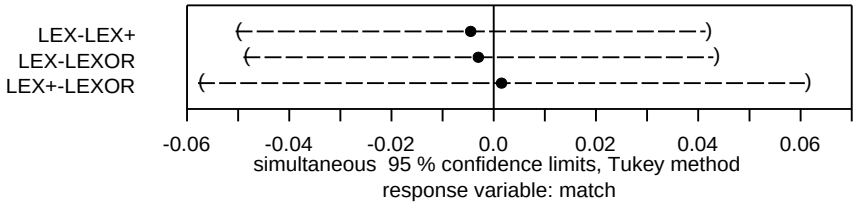

Figure 7.10: Multiple comparisons of lexical focus and lexical focus with unique identification

For this experiment, we used the best distractor set settings as indicated by the above experiment. An ANOVA comparing (1) the experiment where LEX does not consider adequacy, (2) the experiment where LEX+ considers adequacy by adding on attributes as selected by **IDAS**, and (3) the experiment where LEXOR considers it by replacing the description with the **IDAS** selections, showed that there were no significant performance differences ($F = .03, p < .97$). However, to see what the performance trends are we did an MCA comparison, the results of which are shown in Figure 7.10. From this we can see that there are no significant differences between the three but there is a small trend for LEX+ to perform better. We selected LEX+ for the adequacy setting.

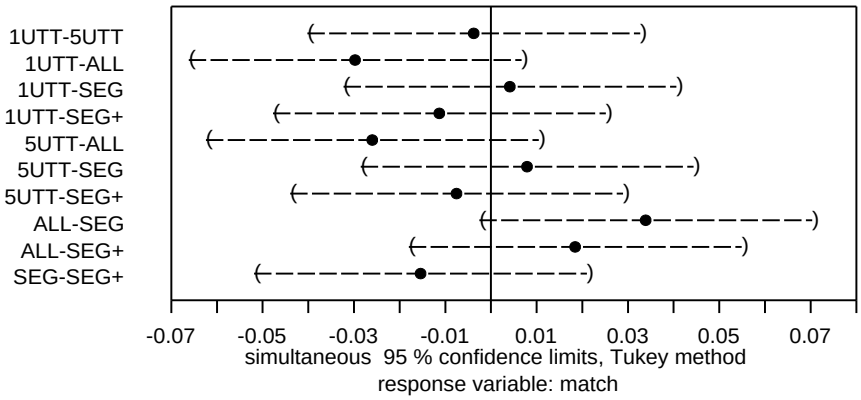

Figure 7.11: Multiple comparisons of distractor set for Diff

7.2.4 Establishing the Unknowns for Differences

7.2.4.1 Distractor Set Definitions

Experimenting with the five values for the distractor set definition (1UTT, 5UTT, SEG, SEG+, ALL), the ANOVA for the **differences** algorithm was ($F = 1.999, p < .09$). This shows that there is no significant difference between the five distractor set definitions. Still we wanted to decide on one setting. From the MCA comparisons shown in Figure 7.11 we see that ALL has a trend to perform better than the others. We selected ALL as the distractor set definition for the **differences** algorithm.

7.2.4.2 The Saliency Threshold

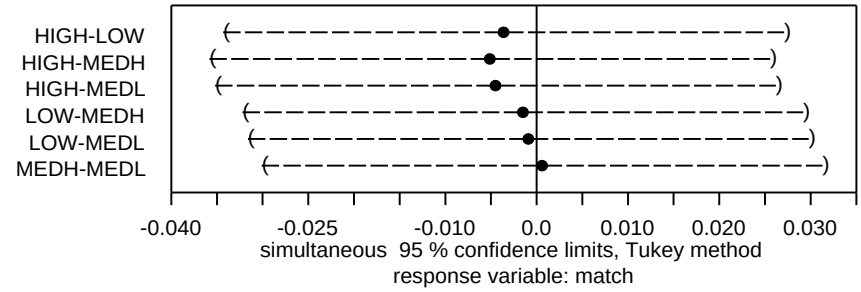

Figure 7.12: Multiple comparisons of saliency thresholds for Diff

When we ran an experiment in which we varied the threshold for determining which if any value was most salient, the ANOVA results were ($F = .07, p < .97$). Although varying the saliency doesn't change the performance significantly, we can see from the MCA comparisons in Figure 7.12, that MEDH (.65) and MEDL (.45) have trends towards better

performance with MEDH perhaps being slightly better. We used MEDH as the saliency threshold setting for the **differences** algorithm.

7.2.5 Establishing the Unknowns for IDAS

7.2.5.1 Distractor Set Definitions

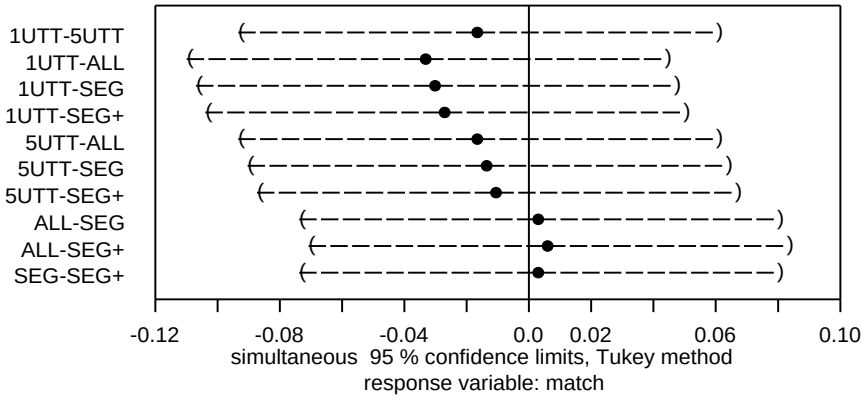

Figure 7.13: Multiple comparisons of distractor sets for IDAS

Finally, we need to establish which distractor set definition works best for the **IDAS** approach. From the ANOVA of ($F = .461, p < .764$), we see that there are no significant performance differences. But from the MCA comparison shown in Figure 7.13 we can see that ALL has a trend to perform better than the others and so we used this definition of the distractor set for **IDAS**.

7.2.6 Putting Together the Best Combinations for Identification

Here we will summarize the settings we used to get the best performance for each of the identification only algorithms:

- **Gestalt**: distractor set=SEG, template=TC, adequacy=GEST+
- **Lexical focus**: distractor set=ALL, adequacy=LEX+
- **Differences**: distractor set=ALL, saliency=MEDH
- **IDAS**: distractor set=ALL

In section 7.3, when we compare algorithms, we will use the following abbreviations to represent the best performing versions of the algorithms with the above settings (GEST+, LEX+, DIFF, IDAS).

7.2.7 The Experimental Factors in the Intentional Influences Algorithm

Here we will describe the factors that we varied for the experiments on the **intentional influences** algorithm. These factors include:

- identification algorithm
- hypotheses inclusion
- adequacy

When none of the contexts of the hypotheses hold, at minimum we want to satisfy the identification goal but we don't know which identification algorithm works best within **intentional influences**. For the identification algorithm factor, we re-examined the internal settings for the identification algorithms since they work in concert with **intentional influences**. We expected to get similar results since the identification algorithms are working on a subset of the redescriptions for which none of the contexts tested by **intentional influences** apply. We did not reconsider adequacy for **lexical focus** and **gestalt** simply for reasons of economy and assumeed that adequacy is necessary.

For the hypotheses inclusion factor, we did not attempt to evaluate each of the six hypotheses separately to see which has the most impact. This is because we expected that there would not be enough instances of any one of the context factors that the hypotheses describe to show meaningful differences in performance. However, in Chapter 5, we found correlations that clearly supported only four of the six hypotheses; Domain Constraint Changes, Informational Relations, Persuasion and Commitment. The remaining two algorithms, Verification and Summarization, were not supported but we wanted to act skeptically since the correlational statistics used assume that the descriptions are independent from one another and they are not. This means that we grouped the first four hypotheses and experimented with whether adding in Verification, Summarization or both improves the performance of **intentional influences**. For both hypotheses we will use the following values:

- include (T)
- exclude (NIL)

As with the identification algorithms, the adequacy factor varies what we do when the algorithm's choice of description does not uniquely select the target object from the distractor set. For this experiment we varied **intentional influences** between considering adequacy, where we add on attributes using the **IDAS** approach, and not considering

adequacy at all. It is not appropriate to try substituting the **IDAS** selection whenever the **intentional influences** selection is inadequate since the two are not addressing the same goals. We tried two possible values for the adequacy factor:

- Use the main algorithm's unaltered description without considering unique identification (INF)
- Use the IDAS algorithm to supplement the main algorithm's description with additional attributes in an attempt to make the description adequate (INF+)

We divided the experiments up according to which of the four identification algorithms we are using internally within **intentional influences**. We find the best settings for all the internal identification factors concurrently with the hypotheses inclusion factor and the higher level adequacy factor for **intentional influences**. Once we had the best settings for each combination we compared the four selection algorithms (i.e. **intentional influences** combined with either LEX+, GEST+, DIFF or IDAS) to see which is the best performer. We will not show the ANOVA results and confidence intervals for the internal settings of the identification algorithms here, as the process is the same as for the identification algorithms when used in isolation. Instead we will report the best settings and the reader can see Appendix B for the details. We will however show the results for each of the four identification algorithm settings for the hypotheses inclusion and the higher level adequacy factors.

Contrary to our expectations, the best internal settings for the identification algorithms were different from those we found when the algorithms were used in isolation:

- GEST+: distractor set=SEG+, template=T(ype)
- LEX+: distractor set=5UTT
- IDAS: distractor set=SEG+
- DIFF: distractor set=1UTT, threshold=MEDH

7.2.7.1 Exploring Which Hypotheses to Include in Intentional Influences

For this experiment we looked separately at the four combinations of **intentional influences** and identification algorithms. An ANOVA in which we examine the effects of varying whether we include the Summarization Hypothesis shows that there is a significant difference in performance of ($F = 25.71, p < .0000004$) for GEST+, ($F = 6.66, p < .0099$) for LEX+, ($F = 15.2, p < .000097$) for DIFF and ($F = 11.03, p < .0009$) for IDAS. From

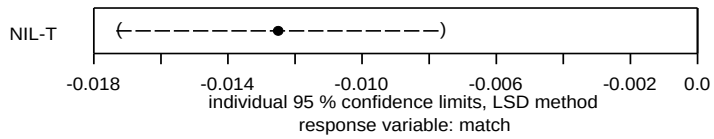

Figure 7.14: Multiple comparisons of Summarization for **Intentional Influences** and GEST+

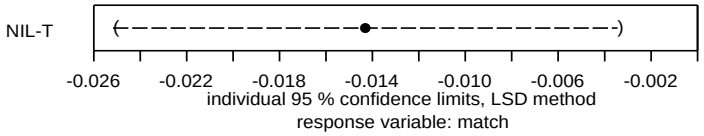

Figure 7.15: Multiple comparisons of Summarization for **Intentional Influences** and LEX+

the MCA comparisons shown in Figures 7.14, 7.15, 7.16, and 7.17 we see that it is better to include the Summarization Hypothesis in the **intentional influences** algorithm.

An ANOVA in which we examine the effects of varying whether we include the Verification Hypothesis shows that there is a significant difference in performance for GEST+ ($F = 18.71, p < .00002$) and DIFF ($F = 7.08, p < .0078$) but not for LEX+ ($F = .96, p < .328$) or IDAS ($F = .007, p < .94$). From the MCA comparisons shown in Figures 7.18 and 7.19, we see that there is a trend for it to be better to include the Verification Hypothesis for these two cases, and from Figures 7.20 and 7.21 we see it is better not to include the verification hypothesis for these other two cases.

7.2.7.2 Adequacy for Intentional Influences

For this experiment we varied whether adequacy was checked and resolved in the cases where some of the **intentional influences** contexts applied. An ANOVA of the experimental results for GEST+ of ($F = .631, p < .427$), LEX+ of ($F = .267, p < .61$) and DIFF of ($F = .489, p < .485$) shows that there are no significant performance differences for these three cases. However there is a significant performance difference for IDAS ($F =$

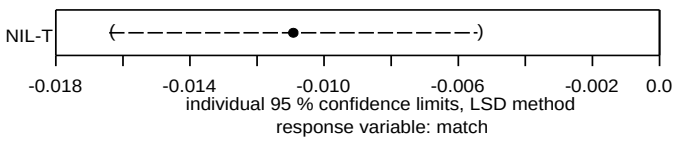

Figure 7.16: Multiple comparisons of Summarization for **Intentional Influences** and DIFF

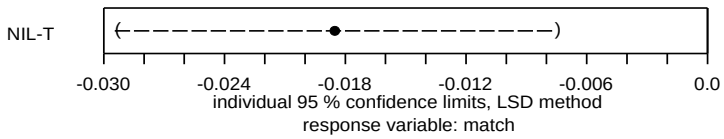

Figure 7.17: Multiple comparisons of Summarization for **Intentional Influences** and IDAS

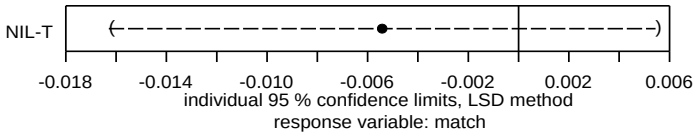

Figure 7.18: Multiple comparisons of Verification for **Intentional Influences** and LEX+

$6.72, p < .0095$). To see what the performance trends are we did MCA comparisons for each of the four versions of **intentional influences**, the results of which are shown in Figures 7.22,7.23,7.24 and 7.25. From this we can see that INF+ is the setting that provides the best performance or has a trend towards being the better performer for all four cases.

7.2.7.3 Exploring Which Approach to Use to Satisfy Identification within Intentional Influences

Next we compared the performance as we varied the identification algorithm between LEX+, GEST+, DIFF and IDAS. For this, we used the best settings for the identification algorithms when they are used within **intentional influences**. An ANOVA of this experiment ($F = 2.71, p < .044$) shows that there is a significant difference in **intentional influences**' performance. To see what the performance trends are we did an MCA comparison, the results of which are shown in Figure 7.26. From this we can see that GEST+ is better than LEX+ and has a small trend towards better performance than the others. We used GEST+ to satisfy the identification goal within **intentional influences**

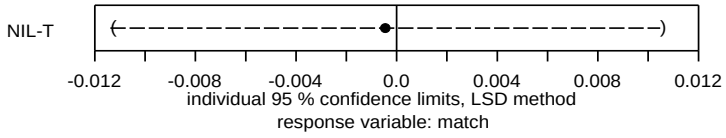

Figure 7.19: Multiple comparisons of Verification for **Intentional Influences** and IDAS

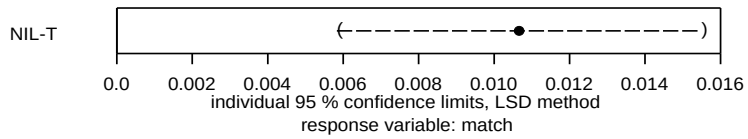

Figure 7.20: Multiple comparisons of Verification for **Intentional Influences** and GEST+

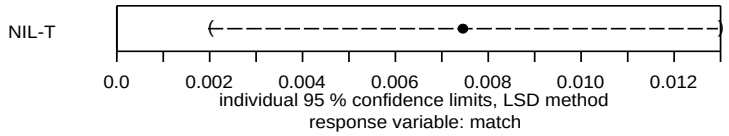

Figure 7.21: Multiple comparisons of Verification for **Intentional Influences** and DIFF

7.2.8 Putting Together the Best Combinations for Intentional Influences

We will call the best performing **intentional influences** algorithm INF+ in the comparisons to the identification algorithms that follow. To summarize, it has the following internal settings:

- identification algorithm: adequacy=GEST+, focus=SEG+, gtemplate=T(ype)
- hypotheses inclusion: Verification=NIL, Summarization=T
- adequacy=INF+

Although all the identification algorithm settings are the best ones for GEST+, they are not independent of its use within INF+.

7.3 Comparing Algorithms

Now that we have established the best internal settings for all of the algorithms in an attempt to extract the best possible performance from each, we can compare them to one another. In Table 7.1, we list the means for the *match* variable, to give an intuitive feel

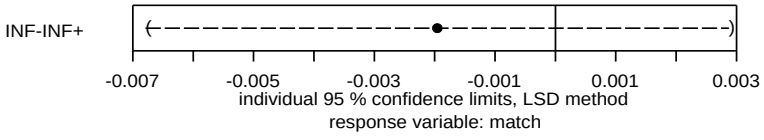

Figure 7.22: Multiple comparisons of adequacy for **Intentional Influences** and GEST+

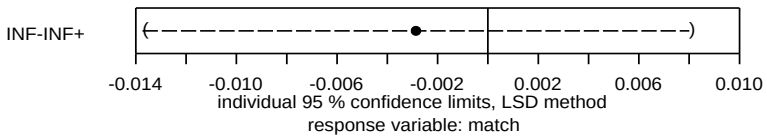

Figure 7.23: Multiple comparisons of adequacy for **Intentional Influences** and LEX+

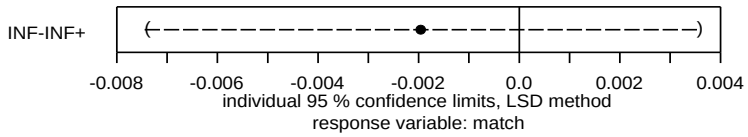

Figure 7.24: Multiple comparisons of adequacy for **Intentional Influences** and DIFF

for the performances ($F = 18.56, p < 0$). First we compare the identifications algorithms with three extreme algorithms:

- Always express every property (ALWY)
- Never express any property other than the type (NVR)
- Randomly select properties to express (RAND)

We expect all of the identification algorithms to perform better than these extremes. We do this comparison to confirm our expectation, to establish what poor performance looks like, and to determine how much better the identification algorithms are relative to the extreme algorithms.

7.3.1 The Identification Algorithms vs. the Extreme Algorithms

When we did an ANOVA we found that there were significant performance differences ($F = 16.95, p < 0$). From the MCC comparisons shown in Figures 7.27, 7.28, and 7.29, we find that all the identification algorithms are significantly better than RAND and ALWY but only DIFF and GEST+ are significantly better than NVR. However IDAS and LEX+ do have trends towards being better than NVR.

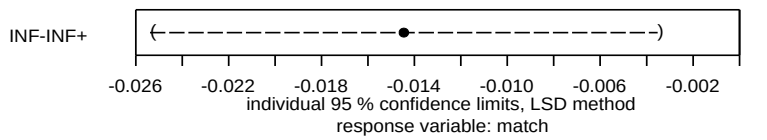

Figure 7.25: Multiple comparisons of adequacy for **Intentional Influences** and IDAS

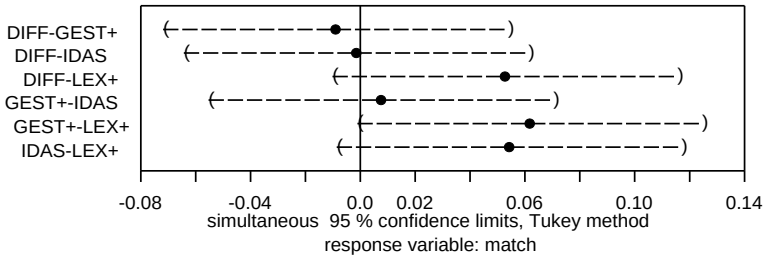

Figure 7.26: Multiple comparisons of identification algorithms within **Intentional Influences**

Algorithm	Mean Match
INF+	.6958
DIFF	.6762
GEST+	.6611
IDAS	.6476
LEX+	.6130
NVR	.5798
RAND	.4970
ALWY	.4774

Table 7.1: Mean algorithm performances

7.3.2 Comparing the Identification Algorithms

When we separate out the identification algorithms from the extremes, we have no predictions about which identification algorithm will perform better. An ANOVA of ($F = 2.72, p < .044$) shows that there are significant differences in the performances of the identification algorithms. In Figure 7.30 we see that DIFF performs better than LEX+ and has a trend to perform better than the other algorithms.

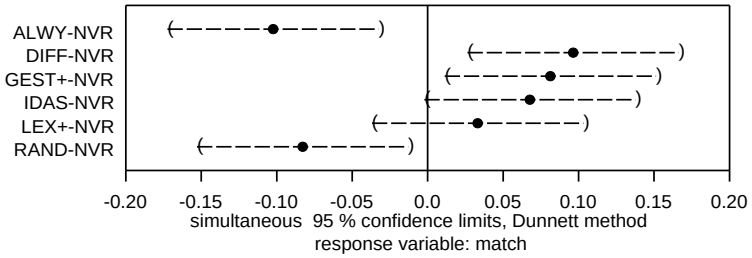

Figure 7.27: Multiple comparisons of all identification algorithms to NVR

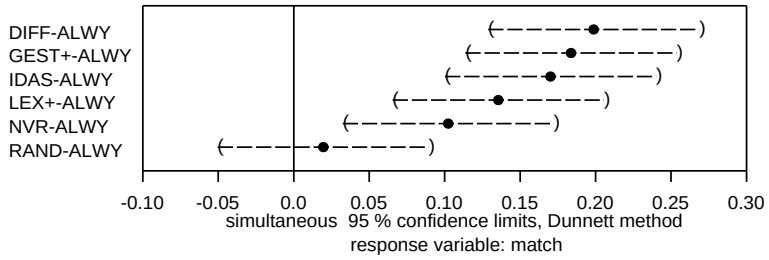

Figure 7.28: Multiple comparisons of all identification algorithms to ALWY

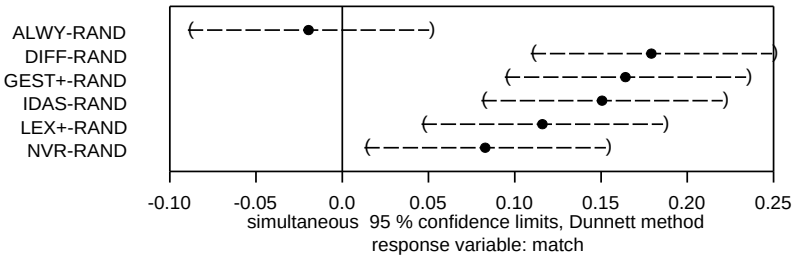

Figure 7.29: Multiple comparisons of all identification algorithms to RAND

7.3.3 Intentional Influences vs. Identification Only Algorithms

Now we compare the **intentional influences** algorithm to the Identification algorithms to see if there are any significant performance differences between them. An ANOVA of ($F = 3.6, p < .006$) shows that there are significant performance differences. From the MCC comparisons shown in Figure 7.31 we see that INF+ performs significantly better than LEX+ and has a trend to perform better than the other identification only algorithms. If we compare all these algorithms to just IDAS (the standard) we see from Figure 7.32 that it only has a trend to perform better than LEX+ and tends to perform less well than the others.

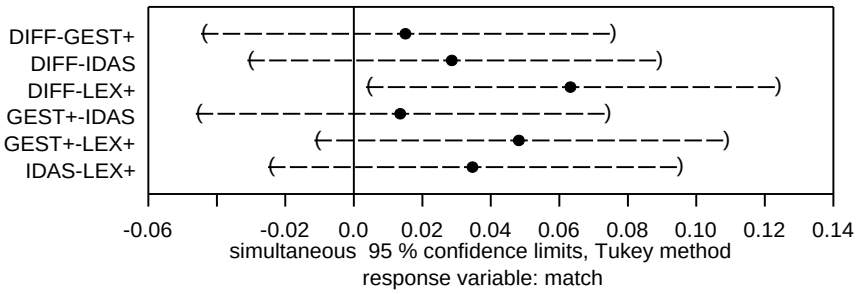

Figure 7.30: Multiple comparisons of identification only algorithms

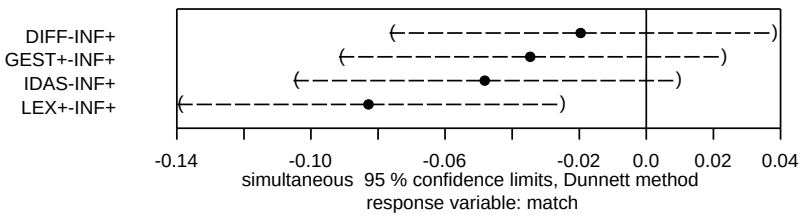

Figure 7.31: Multiple comparisons of INF+ with identification only algorithms

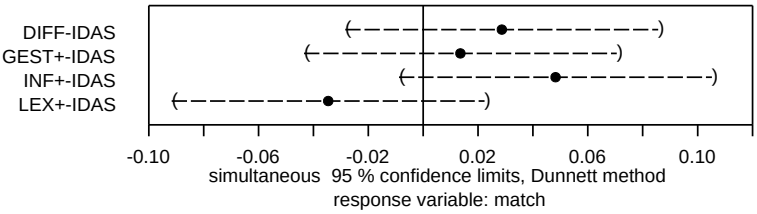

Figure 7.32: Multiple comparisons of IDAS with INF+ and other identification only algorithms

Next we want to explore why there are no significant performance differences between INF+ and the identification algorithms other than LEX+. The degree of match could be similar because there may be many allowable ways to express a description for identification purposes and the overloaded selections could intersect some of these allowable ways. However, if the overloaded goals are not generally allowable as influences on description selection then INF+ should perform similarly to a randomization of some of the selections motivated by the identification goal.

To test this idea, we created a selection strategy, RINF, that randomly selects a random number of attributes. As with INF+, it then uses IDAS to determine if additional attributes are needed to rule out distractors. Furthermore, since SEG+ was the best distractor set definition for INF+, we also use this setting for our randomized selection algorithm. We also include the performances of the non-randomized IDAS algorithm with the SEG+ setting for the distractor set definition in this new experiment to confirm that its performance is statistically similar to INF+.

We found significant differences between the three selection algorithms, INF+, RINF, and IDAS ($F = 6.05, p < .003$). As shown by the MCA confidence intervals in Figure 7.33, INF+ matches human descriptions significantly better than RINF and as well as IDAS. In addition, as before, INF+ also has a trend towards better matching compared to IDAS. This suggests that the goals we overloaded onto redescriptions are valid influences since they did not randomize the descriptions generated.

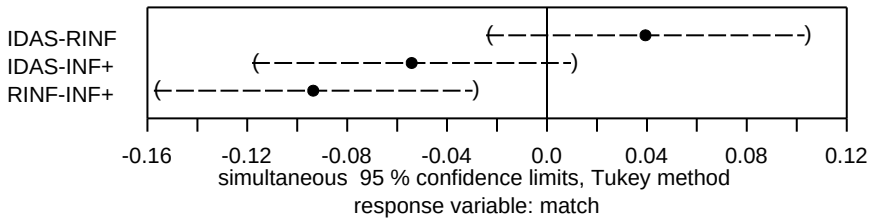

Figure 7.33: Multiple comparisons of RINF, INF+ and IDAS

Although we do not have enough data to determine which, if any, of the contexts in our hypotheses is most influential for attribute selection, we can show in Table 7.2 the relative contributions of each of these contexts and the contribution of the identification goal within the **intentional influences** algorithm. This gives us an informal view of the relative contributions without considering the frequency with which a particular context arises. The contribution made by the identification goal includes both the cases in which identification was the only predicted goal and the cases in which additional attributes had to be added to ensure unique identifiability after the initial selections made by the other goal contexts. Although the contribution from identification is smaller than one might expect, this does not mean that the identification goal was invalid for some redescriptions. Instead it indicates that the problem of identification was already addressed by the attributes that had already been selected by the other goal contexts. This reflects the economy that can be achieved with goal overloading.

Hypothesis	Percentage Contribution to Descriptions
Persuasion	16.67%
Domain constraint changes	5.33%
Summarization	22.67%
Commitment	26%
Informational relations	0%
Identification	29.33%

Table 7.2: Contributions of contexts and goals to attribute selection

Although we do not see a contribution from the Informational Relations Hypothesis to suppress properties that would have otherwise been described, we did see it play a role in some of the other variations that we tried for the **intentional influences** algorithm. Informally it seems to apply whenever the contribution from identification approaches 60%. This potentially indicates that when given a prominent role in the selection, identification is likely to add in the properties that later need to be suppressed.

Our results show that the **intentional influences** algorithm is significantly better at matching human data than the **lexical focus** algorithm and while it performs similarly to the other identification only strategies, we saw that the overloading was more than a slight randomization of the attributes selected to satisfy an identification goal and that the embedded identification strategy was making only a small contribution to the attributes selected. In the next chapter, our concluding chapter, we will review and further analyze our results.

Chapter 8

Conclusion

In this dissertation, the central question that we explored is whether other communicative goals, besides identification and communicating new properties as required by the task, can influence the content of discourse anaphoric expressions. We used corpus analysis and comparisons of parameterized algorithms as our forms of exploration. Some of the underlying questions that we identified and addressed with respect to current approaches for generating referring and discourse anaphoric expressions [App85c, Kro86, Dal92, HH95, Loc95, Rei90a, Pas96], included:

1. Is the potential redundancy of attributes at the level of object redescriptions an accident of nature or can it also serve an intentional purpose?

2. What goals can influence the choice of attributes?

3. Does consideration of multiple goals result in choices more like what humans would make?

4. How does discourse structure influence the content of discourse anaphoric expressions?

5. How do we determine the saliency of non-perceptual properties (e.g. price) relative to perceptual ones (e.g. color) [Lev89] for use in satisfying identification goals?

In answering the above questions, we found evidence of multiple influences on description selection and that the extra attributes are not all by-products of the cognitive architecture. We found that many propositional level goals, particularly those related to the functions of repetition, did influence the choice of attributes. There was indicative, but inconclusive evidence that consideration of multiple goals makes the selection of attributes more like what humans would make. We instead found that by overloading [Pol92] multiple goals onto redescriptions, we produced what are possibly acceptable alternative descriptions. Next we found that when considering the entities within a discourse segment [GS86] as

the distractors for identifying a target entity [Dal92], there was a trend for the choice of attributes to be closer to that of humans when we allow goals besides identification to influence attribute choices but not when we only considered the identification goal. However, it was a surprise to find that there were no significant performance differences when we varied the definition of the distractor set by using either discourse segmentation, global focus or simple frequency. We also found a trend towards better performance when we allow a static saliency hierarchy that gives preference to perceptual properties to be overridden by non-identification goals.

8.1 Summary and Discussion of Results

We began to explore question (1) by analyzing the degree of non-minimality with respect to identification for the COCONUT corpus. Recall that non-minimality means there are more mutually known attributes included in the description than are necessary for re-evoking or identifying the appropriate discourse entity. We found the COCONUT corpus to have a high degree of non-minimality (46% compared to 6% for the Pear stories corpus [Pas96]). First, it seems less plausible for such a high degree of non-minimality to be accidental (i.e. due entirely to the cognitive architecture [Lev89, DR95, Pas96, Cre96]) although this measure does not prove or disprove the question. It does indicate, however, that the COCONUT corpus is a good resource for studying whether attributes not needed for identification can serve some other purpose. We anticipated that this characteristic of high non-minimality would make influences from non-identification sources measurable relative to the postulated strong influence of identification.

While we first sought a corpus with a high degree of non-minimality with respect to identification, in the analyses that followed we didn't merely consider the non-minimal expressions. We considered all the redescriptions with the aim of getting the best possible match to human performance in order to address question (3) about whether choices which also included non-identification goals would be more like the human choices in the corpus. Also for the domain of the COCONUT corpus, non-perceptual properties were predominant in two different respects. First, all the task objects existed only as artifacts of the task and second, some of the attributes used for problem solving (e.g. price) and identification (e.g. owner) were non-perceptual. This use of non-perceptual properties allowed us to consider question (5) about how to interpret saliency for a mix of perceptual and non-perceptual properties when attempting to satisfy the identification goal.

To further address question (1) and to start on question (2), about what goals can influence the choice of attributes, and question (3), about whether the choices are more

like what humans would make, we first described some candidate communicative goals that potentially could be partly satisfied by the content of an object redescription. The communicative goals that we discussed derived from work on the functions of repetition at the utterance or propositional level [WS88, CS89, MP89, Bre90, Wal93], from observations about task goal changes that were not directly communicated, and about how intentional and informational relations influence content selection [McK85, MM95]. We then formulated the following six hypotheses about how each communicative goal might influence redescriptions:

- DOMAIN CONSTRAINT CHANGES: Properties related to constraint changes are expressed in a context where the change must be inferred by the hearer.
- INFORMATIONAL RELATIONS: When the effect of, goal of, or the constraint imposed on an action is communicated within the same utterance as the action then the properties made salient by such a relation are **not** expressed.
- PERSUASION: Property values that are pivotal for deliberation are expressed in the context of goals to communicate a proposed action.
- COMMITMENT: In the context of a commitment to a proposal, all the properties expressed in the proposal will be repeated.
- SUMMARIZATION: In the context of a previously completed problem or subproblem, all the mutually known properties for an item will be repeated.
- VERIFICATION: In the context of a newly introduced entity, all the properties expressed will be repeated by the hearer in his/her next turn.

To verify these hypotheses, we undertook a two part corpus investigation. First, we did correlational studies on the corpus using factors derived from two sets of annotation features: one existing scheme from the COCONUT project [DJTM00] and one that we developed specifically for this dissertation research. Our correlational study showed that goals besides identification can influence redescriptions—the goal contexts and the influences we expected those goal contexts to exert on the redescriptions positively correlated for all but the Verification and Summarization goal contexts. This indicates that redundancy in redescriptions can be intentional and gives us some guidance on which of the goal contexts were likely to influence the choice of attributes.

In the second part of our investigation we analyzed how well computer simulated selections for the COCONUT corpus matched human selections. We simulated selections for

the COCONUT dialogues by using annotations[1] about the discourse entities to be described and the contexts in which they appeared as input to the selection algorithms we wished to test. We then used the human generated descriptions in the COCONUT corpus to evaluate the descriptions created by the selection strategies we wished to test. To compare the performance of a selection strategy to that of humans, we used a measure of the degree of match between the human's and the strategy's selection of attributes for the same discourse entity in the same dialogue context. Inclusion and exclusion of a property both count in the degree of match. A perfect match means that the strategy chose to include or exclude the same properties as the human did for a particular entity.

Simulating the redescriptions in the COCONUT corpus and comparing the results to the actual redescriptions helped us get a clearer idea of the possible answers to question (2), about which goals actually influence the redescriptions. This is because we could see whether the hypothesized influences produced choices that were closer to human performance and also helped us in answering question (3).

However, to see if the goals in question made a difference, we needed a baseline that showed how choices without those additional influences would compare to human performance. So next we had to explore how to best satisfy identification goals and this led us to address question (4), how the content of discourse anaphoric expressions are influenced by discourse structure and question (5), how to determine saliency for a mix of perceptual and non-perceptual properties. While current approaches for satisfying identification goals all agree that discourse structure and attribute saliency influence the choice of attributes for identification, none agree on the specifics ([Hei83, Rei85, PS84, GS86, Kam93] *inter alia*). For this reason we experimented with a variety of approaches to find which ones led to selections that are more like those made by humans.

First, current approaches say nothing about how to mix perceptual and non-perceptual attributes when determining saliency [Lev89], and there are multiple ways in which saliency can be used to influence description selection. We realized some of the possible ways of interrelating saliency and the satisfaction of the identification goal in the form of four algorithms that are parameterized for how they let the discourse structure influence description selection: **gestalt**, **lexical focus**, **IDAS**, and **differences**. Recall from Chapter 6:

- **Gestalt**: attributes are selected because they make the physical search for the target object more efficient [Lev89].

[1] We used the same annotation features as for the correlational studies.

- **Lexical focus**: previous descriptions are repeated because they have been established as mutually agreeable ways of identifying the object [Pas96].
- **IDAS**: attributes of the greatest saliency for the domain are chosen on the basis of whether the target object's attribute value will uniquely select it from a set of distractors [DR95].
- **Differences**: attributes of the greatest saliency in the domain are chosen on the basis of whether the target object's value will either contrast with the most salient value in the space or uniquely select the target object from a set of distractors.

By varying the identification algorithm we were able to see what happens to performance when computing and applying saliency in different ways. While **differences** was significantly better than **lexical focus**, there were no other significant differences between the identification algorithms. This indicates that **lexical focus**, which shares similarities to [CWG86, GA87, OC91], is a bad strategy for the COCONUT corpus. Attribute saliency is not influenced by the last description of the discourse entity for a non-interruptible communications setting.

But by looking at the trends towards better performance between the identification algorithms, we found some interesting tendencies. We found that when adequacy was considered, the **differences** algorithm which considers salient values as well as salient properties and the **gestalt** algorithm which uses a gestalt of "type" and "color", tended to perform better than the well known **IDAS** algorithm. The main conceptual difference between the best **gestalt** and **IDAS** settings was the gestalt template. **IDAS** always includes the "type" whereas the best **gestalt** setting was to always include both "type" and "color". Since the best **gestalt** setting also considered adequacy, additional attributes were included in the description until the target was uniquely identifiable for the given distractor set. This meant that the main difference was that **gestalt** always included "color" in the description. These results indicate that the "type" attribute alone does not make a good gestalt for all situations and that saliency should be considered at more than just the level of attribute types as was assumed by **IDAS**.

To make use of both perceptual and non-perceptual properties, we treated all the properties identically and used the frequency of occurrence to establish a static saliency hierarchy for **IDAS** and **differences** and for **gestalt** and **lexical focus** when adequacy was considered. Given the frequency of occurrence, the saliency hierarchy gave preference to the perceptual properties (i.e. the ordering was "type", "color", "owner", "price", and "quantity"). Note that while "owner" is a non-perceptual property, its only role in the CO-

CONUT domain is identification[2]. Only in the case of **lexical focus** could non-perceptual properties take precedence since it used the properties from the previous description in the dialogue that a human participant had selected. It appears, based on frequency of occurrence and **lexical focus**' poor performance that perceptual properties do tend to be favored in descriptions.

However, to better address the question of how to integrate the treatment of perceptual and non-perceptual properties, we would need to try a variety of saliency hierarchy orderings. But doing so at this time does not seem fruitful. Although the static ordering is recognized by everyone as an oversimplification, the mechanism by which the saliency should be altered has not yet been adequately explored. While some studies from cognitive psychology give us some insight into the machinery needed for altering saliency (i.e. memory models), we need to know how knowledge is represented in memory and how problem solving goals and reasoning should be represented in memory. And these are still difficult, open, research questions. For example, how should furniture concepts be represented so that salient values and properties can be determined?

The second area of exploration in our search for the best approaches to satisfying the identification goal is the relationship between discourse structure and the content of discourse anaphoric expressions. There are many theories for partitioning the discourse but no empirical studies that conclusively support one over another. For this exploration, we varied the definition of the distractor set by varying the discourse partitioning approach. By doing so, we could see which distractor set definitions and which identification-only algorithms worked better together. Recall the distractor set definitions from Chapter 4:

- **ALL**: all entities previously mentioned in the discourse
- **1UTT**: all entities mentioned in the previous utterance
- **SEG+**: all entities in the current discourse segment and all the entities currently in the solution set for the action being addressed
- **SEG**: the union of the entities in the current discourse segment and the last discourse segment in which the target object occurred
- **5UTT**: all the entities mentioned in the previous 5 utterances

First, it was surprising that there were no significant performance differences between these distractor set definitions. However, we again saw some interesting tendencies

[2]The reason for this is that ownership of a furniture item had no influence on the problem solving given the task goals that were presented to the dialogue participants represented in the COCONUT corpus.

when looking at the performance trends. It was surprising that **ALL** had the trend for better performance for a majority of the identification algorithms and that the more widely assumed and more restrictive definition, **SEG**, only tended to work better for the **gestalt** algorithm.

Again, these trends indicate that always including "type" and "color" and not just "type" as with **IDAS** was a better strategy since it both performed better and agreed with the standardly assumed way of partitioning a discourse (i.e. partitioning according to discourse segment purpose as with [GS86]). The **gestalt** identification algorithm perhaps didn't need to overdo the description (i.e. a larger distractor set correlates with increased ambiguity and the need for more complex descriptions). In contrast, we suggest that the other algorithms may have needed a reason to include "color" and a large distractor set would be more likely to justify the inclusion. However, since the performance of the identification-only algorithms was poor relative to humans, one would certainly want to consider memory models in future work as a mechanism for determining the distractor set.

After establishing the better approaches for each of the identification algorithms and the better performers of the four, we next needed to integrate various approaches for satisfying the identification goal with the other goals that we hypothesized would be influential in selecting the content of a redescription. We created a parameterized algorithm called **intentional influences** for this purpose. It was parameterized for which other goals were allowed to influence the content and for which identification algorithm was included within it. We found that the better parameter settings for satisfying the identification-only goal were not the same as when identification was one of many goals that could exert an influence, contrary to what we had expected.

The identification goal was always considered for each object that was to be redescribed by the **intentional influences** algorithm. Sometimes identification was the only goal associated with an object and at others one or more additional communicative goals were also present. We investigated these two situations separately.

For the cases where multiple goals were applicable, we simply considered which attributes in the saliency hierarchy should be added in order to make the target object uniquely identifiable. Since in effect the other communication goals had altered the saliency hierarchy, trying different identification algorithms was pointless. We instead focussed on whether addressing adequacy was still necessary. It was possible that the other goals could have selected enough attributes for the object to be uniquely identifiable. We did find a tendency for it to help the match to human performance when we added attributes to make

the object uniquely identifiable. This agreed with our expectation that the identification goal was always applicable.

For the cases where only the identification goal was applicable, we found we could not assume that the settings for the identification-only selection algorithms would be the ones to use within the **intentional influences** algorithm, since only a subset of all the objects that were to be redescribed had identification as the only goal. We made one exception and assumed that adequacy would matter but in retrospect, we should have retested this setting as well. We found that for **IDAS** and **gestalt**, which were nearly identical, that the slightly less restrictive **SEG+** tended to be the better setting and that in general the more restrictive distractor sets tended to be the better choices for the redescriptions where the only applicable goal was identification. **Differences** was also nearly identical in performance to **gestalt** and **IDAS** but in exploring distractor sets for it, we found there was a significantly better result when looking back just one utterance for distractors. Looking one utterance back may have been better in this case possibly because combining attribute and attribute value saliency helped to localize the influences.

It is interesting to note that when identification was the only applicable influence, the standard approaches to satisfying the identification goal had a tendency to work better. This lends more credibility to our claim that there are multiple influences in that when we account for other influences the standard theories tended to work better. In further support of our claim it is also interesting that type tended to be the better gestalt template, making the better **gestalt** approach nearly equivalent to **IDAS**. This means that there were other mechanisms to allow the inclusion of color and so it did not have to be included in the gestalt as when only the identification influence was considered.

Finally we could address again the question of which goals were influential. By making a parameter out of the goals that were considered within **intentional influences**, we were able to see which of our hypothesized goals affected the performance of the **intentional influences** algorithm. Although we should have tested **intentional influences** both with and without each of the goals we singled out as influential, we chose to accept the positive correlational results and only test the negative ones. We found that Summarization did have a significantly positive influence on the match to human performance but that Verification had a significantly negative one.

Overall, we found that the **intentional influences** algorithm was significantly better than **lexical focus** and had a trend towards better performance in comparison to the other three identification-only algorithms. The **intentional influences** algorithm in effect let domain and discourse goals alter saliency by making attributes associated with these

goals become more salient. We found that when only the identification goal is considered, the perceptual properties tend to take precedence (e.g. a saliency ordering of "type", "color", "owner", "price", "quantity", and a gestalt of "type", "color"). However, when the persuasion goal exerts its influence before identification, it can cause non-perceptual properties such as price to sometimes become more salient than a perceptual property such as color. In this way, we were better able to integrate the saliency determination for perceptual and non-perceptual properties.

The evidence all suggests that there are multiple influences on attribute selection for redescriptions indicating that overloading may occur for redescriptions. Getting a significantly better performance for **intentional influences** when compared to all of the identification-only algorithms would have conclusively answered our questions about whether overloading communicative goals makes the selections more like that of humans. However, having **intentional influences** perform significantly better than one identification-only algorithm and finding that it possibly performs similarly to the others is encouraging. We considered that the performance could be statistically similar because there may be multiple allowable ways of redescribing some of the objects and the identification-only and **intentional influences** selection algorithms could both be intersecting similar sized subsets of the actual expressions in the corpus. To account for this possibility, we reasoned that if the overloaded goals were not allowable influences, then a selection strategy that randomly decides to select a random number of attributes before addressing adequacy should perform similarly to **intentional influences**. The **intentional influences** strategy was significantly better than this randomization while **IDAS** was statistically similar to the randomization. This indicates that the influences we tested are legitimate ones and the **intentional influences** selection strategy is not simply adding a small amount of noise to the selections motivated by identification.

There are several possible reasons why we did not see a significant performance difference when we overloaded multiple goals onto redescriptions. One is because there are very likely to be errors in our approximations of the contexts in which the communicative goals arise. Many of the context checks we tested against the corpus of dialogues are approximations of the dialogue participants' knowledge and intentions. A language generation application, on the other hand, would know with certainty some of the things we can only approximate. Also an application would have direct access to the problem solving state and settings that we are trying to approximate. For example, it would have an accurate list of alternative options and it would know what it is actually committed to doing. Here

we could only make a reasoned guess as to whether the human dialogue participants were actually committing to an option.

A second reason why the results may have been inconclusive is that we may have selected an incomplete set of hypotheses. We may have overlooked some goals and included others that were not strong influences.

A third reason, which we mentioned earlier, for why the performance of overloaded redescriptions compared to identification-only redescriptions may have been statistically similar is that there may be alternative allowable ways to redescribe some of the entities. An interesting question that arises in using human performance as the ideal measure is how well humans would agree with one another if asked to describe a particular entity in the context given in the corpus. [YM97] found evidence that there are multiple possible solutions when deciding whether to use a zero anaphor, pronominal, full nominal, or nominal with just the head noun to redescribe entities. There was only moderate agreement between human subjects about which form to use in a set of test texts (Kappa of .41). Although we examined a subset of this issue it is reasonable to expect that we would find a similar result.

Given that the best mean performance measure for matching the human data was .7 for the **intentional influences** algorithm (see Table 7.1 for the mean performances of the selection strategies), the argument for multiple solutions could mean that we have topped out on the performance measure. There could be speaker preferences with respect to the degree and type of overloading attempted. If this is true, we would also expect these preferences to change as the partners get acquainted and adapt to one another given the principle of LEAST COLLABORATIVE EFFORT [CWG86]. Preferences and adaptation imply that we would always fall short of perfect agreement since we are looking at data from more than one speaker pair in our measures. In that case, we may be as close in agreement with the humans represented in the COCONUT corpus as other humans would be. If we are near the topline in performance, then we would not expect to find significant performance differences.

This does not necessarily mean that the influences of overloaded goals on attribute selection is not useful to consider in natural language generation work, instead it could mean that the sort of measure we are using is not the best or only methodology for learning details about constraints on and potential benefits of overloading. A better, but more expensive methodology would be to test this type of overloading in an application by measuring whether overloading helps or hinders interactions with users. For instance, we could measure whether overloading or lack of overloading on redescriptions leads to a change in the number

of inferences made and in the number of misunderstandings. Rather than expecting our current methodology to provide definite answers about constraints on overloading, we expect it to instead serve as guidance on what types of overloading would be most fruitful to test in interactions with users.

8.2 Generalizing the Results

Since we studied only one corpus and argued that it was a good candidate for study because of some of its special characteristics, an important question is the degree to which our findings about intentional influences on object redescriptions apply to other tasks and other discourse genres. It seems plausible that in any dialogue situation we will find that repetition is an important communicative tool given that it helps signal such things as understanding and closings of discussions (e.g. summarizations). Since the repetition functions are general, we expect our findings to generalize to action redescriptions as well. But the applicability of our findings will depend on the degree to which multiple object or action attributes are important to the task. What characteristics of the language setting make repetition at the object level useful remains an open question.

In addition, we expect that the persuasion and commitment goals will also depend upon the degree of collaboration in the problem solving task. We expect that they will generalize to all highly collaborative problem solving tasks. This is because we expect that such tasks require agreement regarding objects or actions and their attributes. However, the generality of the commitment goal may be limited by the communications setting. The dialogue participants may be less willing to assume things and just move on to the next problem when they don't have fine-grained feedback [OC91, KW67, CWG86, GA87, IC87, Whi95, OWW93]. This means that the participants may seek more explicit shows of commitment in non-interruptible settings such as with COCONUT than with interruptible ones.

We do expect the lack of influence by the verification goal to be specific to non-interruptible dialogues and to text. Without the ability for fine-grained interactions, the verification goal could not serve its purpose well, and we would expect it to arise seldom in non-interruptible settings. We also found the informational relations goal only to be applicable locally, making its application more of a sentence planning issue. Because of this we would not expect its influence to generalize in the same way as the other goals.

Finally, although the domain constraint changes goal is task specific, we still expect it to generalize to design tasks. But its applicability will be limited by the flexibility of the task goals. We would expect it to apply more readily to real world problems than to toy

ones since the goals of toy problem domains are frequently non-negotiable. Its applicability will also depend on the relationship between constraints and other problem state changes and attributes of descriptions. If the attributes do not influence the constraint settings or the problem state then this goal will not be a candidate for overloading.

We also expect our findings with respect to performance trends for distractor set definitions and the treatment of saliency to generalize as well. Since discourse partitioning and the relationship between the partitioning and the distractor set agreed with the widely accepted Grosz & Sidner theory [GS86] when we overloaded goals, we expect this result to hold for other genres as well. With saliency, although the saliency hierarchy itself is corpus dependent and we wouldn't expect the hierarchy to generalize, we would expect the overrides of the hierarchy by other communicative goals to hold in general. This is because the relationship between saliency and goals that are in focus is supported by the research work on memory models (recall 3.3.4).

8.3 Future Work

There are two main areas for future work: extending the evaluation of the description selection algorithms, and using machine learning techniques to identify additional influences on description selection and to find out how to better integrate multiple influences.

A first step in extending the evaluation is to do an experiment similar to that of [YM97] and collect data on alternative descriptions from human subjects. We could generate lists of alternative descriptions for each redescription in the corpus and have human subjects select which description they would use at a particular point. If there is only moderate agreement between the humans, then we will know that there are a significant number of alternatively acceptable ways of redescribing some of the entities in the COCONUT corpus. In addition, we can use this new data set to calculate performance on the basis of perfect matches. If a selection strategy's choice matches a description selected by one of the humans for an entity at the same point in the dialogue then it would count as a perfect match. We could then determine which selection strategies are better at producing acceptable descriptions instead of how good they are at matching particular speakers.

Another step in testing the description selection algorithms is to embed them in generation applications and see if humans are able to recognize the application's intentions. While there is no substitute for a simulation approach when evaluating performance, it only addresses part of the evaluation question. Sometimes we want to find models that explain human performance and at others we focus on finding computationally feasible

models that will suffice. We want to know if the extra effort to find a better model is worthwhile from a system building perspective. To make such a determination, we should measure the intelligibility of dialogue agents who use different approaches for selecting the content of redescriptions. As a first step, we could substitute redescriptions in the human data with ones generated by an algorithm and then annotate these revised dialogues to see how well the new annotations agree with the old (with respect to everything but the NP level properties). We would want to compare the agreement that results using **intentional influences** compared to the best identification-only approach (e.g. **differences**). This test would also give us an idea of whether overloading redescriptions really matters for applications. If the agreement is high then the goals are getting adequately addressed elsewhere.

However, even if the agreement is good, we have no measure of how much additional work the annotators had to put into the interpretation. We could measure the length of time spent on the annotations but this would force us to re-annotate the original dialogues too. It would be more effective to incorporate the selection algorithm into a generation agent that converses with a human agent to try out the different approaches to description selection in an application. We would then compare the quality of the interactions as the selection algorithm is varied. We could use, for example, the PARADISE [WLKA97] framework in order to evaluate and compare the identification-only and the **intentional influences** approach. This framework takes user satisfaction into account; and this would reflect how difficult the dialogues were to interpret.

As part of embedding the description selection algorithm in an application, we also want to combine it with the sentence planning component of a natural language generation system. (In Reiter's consensus pipeline architecture for generation, sentence planning is the typical location in which referring expression processing is handled [CDE+99]). We would combine our description selection work with sentence planning approaches, such as [SW98], that provide a mechanism for goal overloading.

Another area of future work is to find out whether there other influential communicative goals besides the ones we tested. It is also possible that we need to consider different ways of combining these influences. In future work we will apply machine learning techniques to the annotated corpus to see if these techniques will provide a more sophisticated, context sensitive algorithm.

It is also possible that our distractor set definitions are flawed or that distractor set definitions vary with the communications setting. It was surprising that there were no significant performance differences between distractor set definitions except in one case. In

that one case, using only the entities mentioned in the previous utterance as distractors was significantly better for the **differences** strategy when it was used in conjunction with **intentional influences**. This lack of performance differences between the distractor set definitions potentially indicates that there is no relationship between the discourse structure and distractor sets or at least not in the ways envisioned so far.

It was also surprising that the **ALL** distractor set definition had a trend towards being the best performer in the case of the identification-only algorithms. Although we suggested that the multiple influences caused the identification algorithms to need an adjustment that could be met by including more distractors in the distractor set, another possibility is the communications setting for the COCONUT task. Some participants create and maintain graphics icons for all the items that have been discussed. These graphics icons could define the distractor set if both participants assume that the other also keeps a record of all the items discussed. We expect that we may be able to learn more about distractor sets with machine learning experiments as well.

Finally we saw that some algorithms worked better for some speaker pairs than others, whereas for some pairs, none of the algorithms were good matches. In future work, it would be interesting to examine whether there are factors about these particular interactions that might make one algorithm a better match than another. It could be that there are multiple strategies that lead to the multiple solutions we discussed earlier and the choices may be due to the preferences or other characteristics of the dialogue pair. Again, we may be able to find this out by applying machine learning techniques to our annotated corpus.

APPENDICES

APPENDIX A

The Annotation Manual

(see http://www.isp.pitt.edu/ jordan/diss/coding-man/diss-man.ps)

A.1 Introduction

The coding scheme described in this manual was developed for the dialogues collected as part of the COCONUT project (http://www.isp.pitt.edu/~intgen/). The goal of this coding scheme is to investigate the influences of cognitive constraints, and domain and discourse goals on descriptions of domain entities. In particular we are focusing on how discourse entities get expressed in dialogue and how this interacts with the expression of action and constraint changes. We are not addressing issues regarding when action and constraint changes are accepted or rejected. We are limiting the scope to when actions and constraint changes are introduced or when actions are taken up in the discussion.

A.1.1 The COCONUT Corpus

The COCONUT corpus is a collection of computer-mediated dialogues in which two subjects collaborate on a simple task, buying furniture for the living and dining rooms of a house. (The task is based on those in [Wal93, WGR93]). Each subject is given a separate budget and inventory of furniture that lists the quantities, colors, and prices for each available item. By sharing this information during their conversation, the subjects can combine their budgets and can select furniture from each other's inventories. The problem is collaborative in that all decisions have to be consensual; funds are shared and purchasing decisions are joint. The subjects in our collected conversations are equal in status: they are both briefed on the domain knowledge needed for problem solving and neither is an expert at the task.

The subjects' main goal is to negotiate the purchases; the items of highest priority are a sofa for the living room and a table and four chairs for the dining room. The sub-

jects also have specific secondary goals which further complicate the problem solving task. Subjects are instructed to try to meet as many of these goals as possible. The secondary goals are: 1) Match colors within a room, 2) Buy as much furniture as you can, 3) Spend all your money. Subjects are enticed to try to arrive at a solution that achieves the maximum number of goals by a task score that associates points with primary and secondary goals.

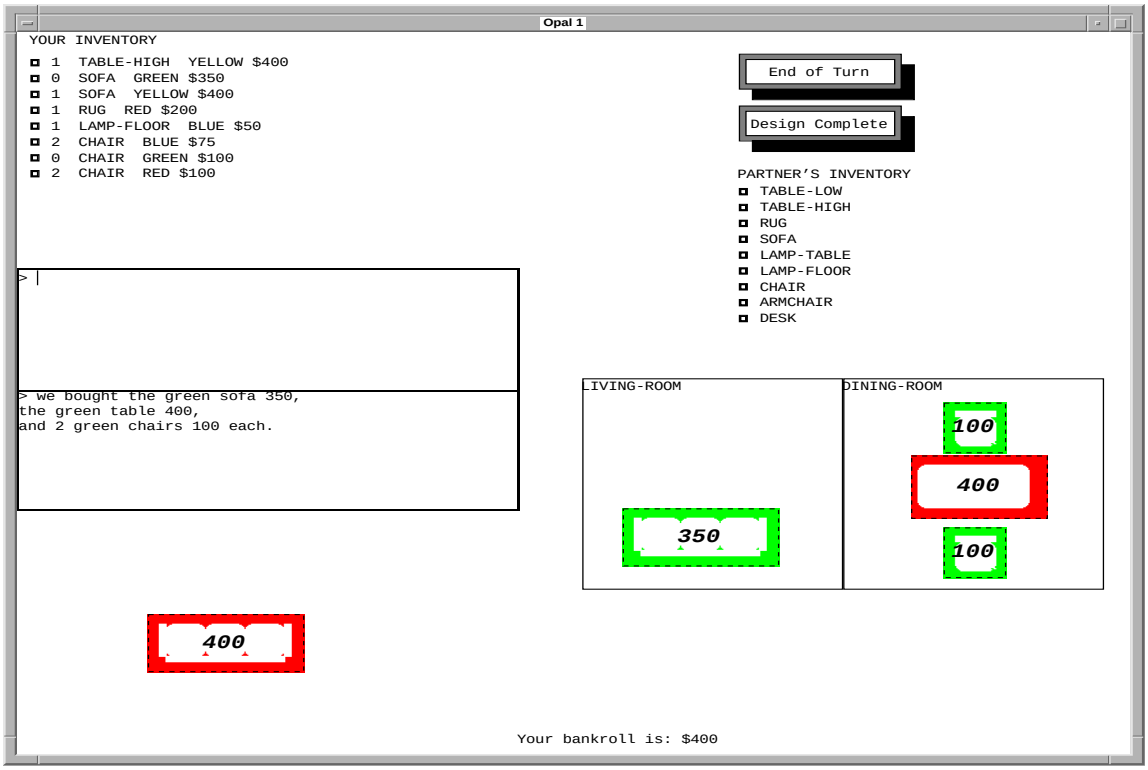

Figure A.1: A View of the COCONUT Interface

The subjects are in separate rooms and can communicate via the computer interface only. They are asked to maintain private graphical representations of their discussions and incremental agreements. We can use this private information as partial evidence of what S's utterance meant and what H understood. Subjects share dialogue windows but the inventories, budgets and updated floor plans are private and show up only on the owner's color display. Figure A.1 shows the interface as it looks in the middle of a design session.

The buttons in the upper right corner of Figure A.1, `End of Turn` and `Design Complete`, enforce turn-taking and initiate the incremental recording of the conversation and the graphics updates. During an incremental recording, the most recently transmitted message is recorded as well as the state of the sender's graphics display. The graphics display record is a description of the furniture icons in the two rooms as well as what has been allocated but not assigned to any room. An additional feature of the interface is that

each furniture icon is initially displayed with a dashed outline around it. The subjects are given the option of turning off the dashed outline to signal that agreement has been reached on using the item in the solution (i.e. the item has been selected as part of the solution).

Note. The examples that are excerpts from the actual Coconut corpus may have uncorrected typographical errors since we present them unaltered.

A.1.2 Coding Schema Approach

The coding scheme combines elements of the DRI coding schema (DRAMA) for discourse reference, and an extended set of informational relations based on the RDA coding schema. The annotation tool we use is *Nota Bene (NB)* [Fla95]. Pointers to both discourse reference coding and NB can be found at http://www.georgetown.edu/luperfoy/Discourse-Treebank/dri-home.html, under *Tools and resources.*

A.2 Top Level Menu and Overview

Figure A.2 represents the menus and submenus set up for NB and shows all the features and feature value choices for the coding scheme as it applies to the COCONUT corpus. A line out of a box shows the submenu for all the selections in that box. The diamond represents the choice *none of the others applies.* Each submenu and selection will be described below. The ordering of the selections in a menu and the ordering between submenus is not necessarily significant.

The top level menu (which is not shown in Figure A.2) consists of two major categories of tags; Utterance-Level and NP-Level tags. In addition there is a Comment tag. Every utterance that is a minimal unit (see Section A.5) must be annotated for the Utterance-Level category and every bracketed noun phrase (see Section A.5) must be annotated for the NP-Level category, except when the utterance unit is not relevant to the task or the utterance is part of a summary or repair[1]. In cases where more than one tag in a category applies or if the same category applies more than once for a particular annotated unit, then separate annotations should be made for each application. In some cases, the tags within a category are mutually exclusive. Unless noted otherwise, the annotator should assume that tags are not mutually exclusive.

[1] We do this to limit the scope of the current research project.

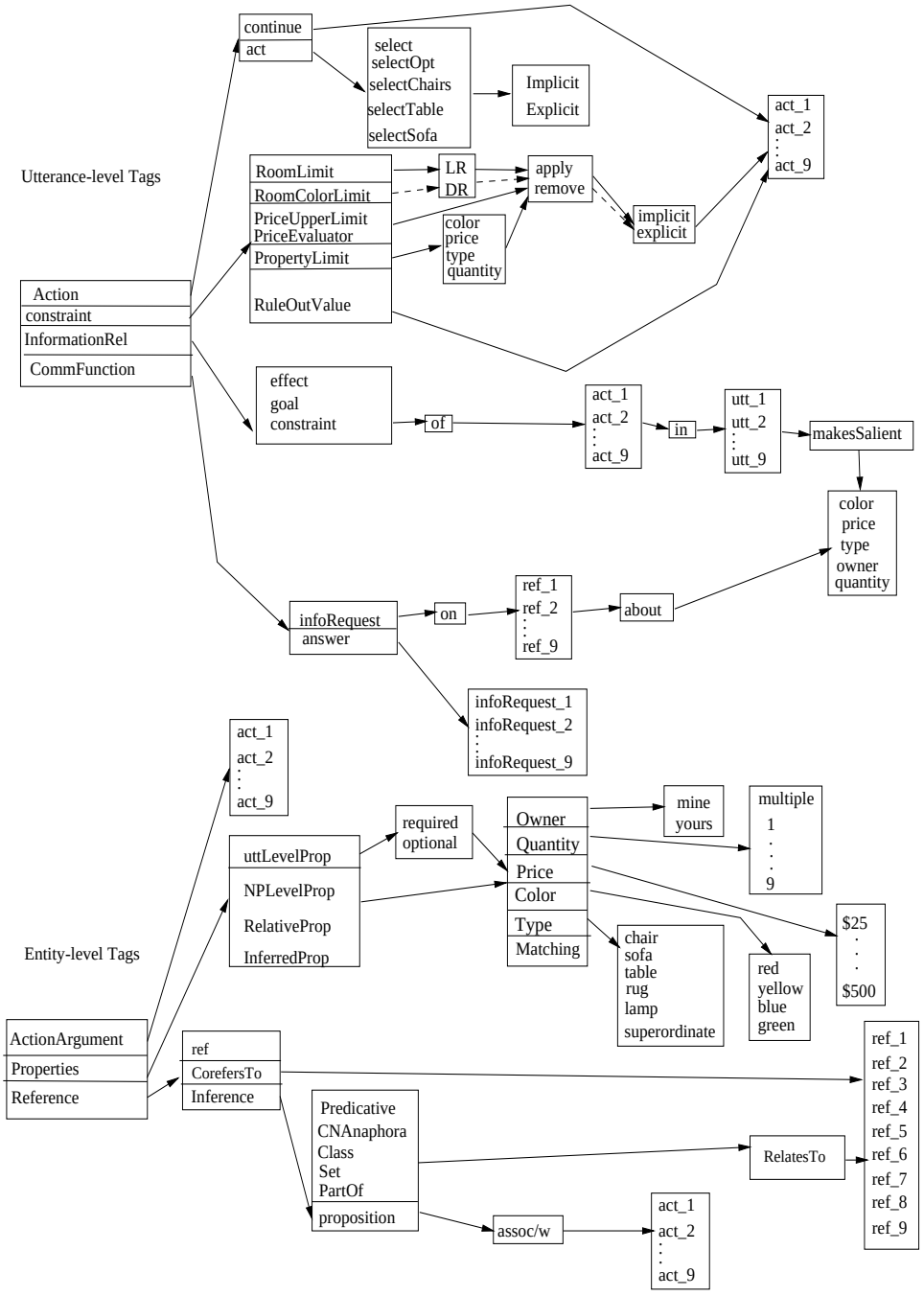

Figure A.2: Overview of Coding Scheme

A.3 Utterance-Level Tags

The utterance-level tags capture information about the domain actions and constraints that an utterance communicates and the communicative and domain relationships between the utterances.

A.3.1 Information about Domain Actions - Action

There is one general action type of interest for us in COCONUT: *select furniture for a room.* But there is a hierarchy of *select* action types for annotation purposes that ranges from this most general action type to the most specific action types that are one level up from fully defined executable actions, such as *select a table for the dining room.* **For each utterance we will code the most specific action types that are unambiguously recognizable**. For example, if an utterance unit discusses tables and chairs and it is inferrable that the primary goal of selecting a table and chairs is being addressed, then the utterance will be coded as having two action types: *select chairs for the dining room* and *select a table for the dining room* instead of the more general *select furniture for the dining room.* On the other hand if it is not inferrable that a primary goal is being addressed, the utterance would be coded as having one action type; *select optional items for a room.*

Coding for the most general action type of *select furniture for a room* would be appropriate when a participant lists all of the furniture that he has available and it is not clear how these would define an unambiguous set of more specific action types. For example, a listing of all of a participant's furniture that includes multiple sets of chairs could be either *select chairs for dining room* or *select optional items for a room.* Because it is ambiguous in the initial context, it should be coded as a more general action type to represent the ambiguity. However, if initially a speaker lists all the items they have of only the types that are most strongly associated with particular actions then it is reasonable to infer that the speaker is addressing those actions. For example, if the speaker lists all the chairs and tables he has then it is reasonable to infer he is addressing *select chairs for the dining room* and *select table for the dining room.* However if he also listed some rugs or lamps in the same utterance then the situation is ambiguous and should be annotated as a more general action type.

Note that if the items of highest priority according to the problem description (see Section A.1.1) have already been decided by the participants, then one can typically infer that any other tables, chairs and sofas that are not replacements are additional optional items. When no decision has been made yet about the priority items, one can infer that

the location for sofas is the living room and the location for tables and chairs is the dining room.

If the most specific action type description for an utterance is the first occurrence of that description then the annotator should begin the coding for the utterance by first selecting `act` from the `Action` menu. This selection causes the tool to assign a unique identifier (e.g. `act_40`) to the new action type.

The action type hierarchy is represented as a combination of choices under the `Action` and `constraint` menus. Under the `Action act` submenu the relevant choices are `select`, `selectOpt`, `selectChairs`, `selectTable` and `selectSofa`. Each will be discussed in more detail below. Under the `constraint` menu the relevant choice is `RoomLimit` and this will be discussed in more detail below in Section A.3.2.

The menu choice `select` is used to indicate the most general action type of *select furniture for a room*. If the location is more specific, then this will be indicated by the `RoomLimit` choice under the `constraint` menu to be described below in Section A.3.2. The choice `selectOpt` should be used when the furniture under discussion is clearly meant to be an optional item. If it is clear which room is meant then this can be indicated with the `RoomLimit` under the `constraint` menu as described in Section A.3.2. The choices `selectChairs` should be used when there is a specific action type of *select chairs for dining room*, `selectTable` for *select table for dining room*, and `selectSofa` for *select sofa for living room*.

Finally, when describing a **new** action type, the annotator should indicate whether the action type description required inferencing on the part of the annotator. Selecting `implicit` indicates that the `select` action and/or the furniture argument type were inferred. Otherwise, `explicit` should be selected. In general, the action type is not explicit. **To determine whether the action type was inferred, consider the utterance in isolation from the rest of the dialogue, but not from the initial problem description. If it is clear what the action type is without this additional context, then the action type is explicit.** For example, in (19), the action type could be either `selectSofa` or `selectOpt`. It is not clear without considering the preceding dialogue as well. However, in (20), it is clear that the action types are `selectTable` and `selectChairs` by considering just the utterance and the initial problem description. If something happened in the dialogue to alter the action type descriptions in this case then it should be annotated for the action type that is consistent with the dialogue. However, it would then have to be annotated as `implicit` since the dialogue has to be taken into account.

(19) Let's get my blue sofa.

The annotator should not consider whether the room is inferred when annotating inference for an action type. This will be taken care of as part of the constraint annotation. For example, (20) would be annotated as `selectTable Explicit` and `selectChairs Explicit`. A continuation of this discussion such as (21) would be annotated as a `continue` action for `selectTable`. On the other hand, if a dialogue started with (21), it would be annotated as `selectTable Implicit`.

(20) Let's decide on the table and the chairs.

(21) I have a $100 blue table for the dining room.

If the action type is a continued discussion of a previously described action type then the annotator simply annotates which action type is being continued by selecting `continue` under the `Action` menu and then indicating which automatically assigned unique identifier was given to the previously described action. Action identifiers appear as `act_[1--9]` in the menu. If the number portion of the automatically assigned identifier is larger than 9, then the annotator can select a close identifier from the menu and edit it to get the correct identifier. The correct action identifier can be found by looking for a previous utterance with the same action type description and placing the mouse cursor on any part of the utterance other than an NP. This placement of the cursor causes the tool to show the annotation for the utterance-level.

Not all utterance units will directly address the domain action hierarchy of *select furniture for a room*. For example, a simple answer to a yes/no question to clarify a value or property of an item would not directly address any domain actions. **To determine whether an utterance directly addresses any domain actions of interest and should be annotated with an action type, check to see if there is a bracketed NP or a [0] in the utterance that has a discourse entity relation to another furniture discourse entity .** (See Section A.4.3 for annotating discourse entity relations and Section A.5 for an explanation of bracketed NPs and insertions of [0] in utterances.) For example, (22) does not directly address any of the action types in the *select furniture for a room* hierarchy since it has no discourse entity relations to a furniture discourse entity. It is only indirectly related since it is a side-effect of some collection of action types. (See Section A.3.3 for annotating side-effects of action types.)

(22) I have $50 left.

Consider another example. In (23), the first utterance contains the elliptical NP, *two*, that refers to some previously discussed chairs (see CNAnaphora in Section A.4.4)

and thus would be annotated as a continued action for *select chairs for the dining room* (assuming the proper contextual information). However, the *yes* response does not directly relate to any furniture discourse entities so it would not be annotated as having an action type.

(23) A: So we can only get [two]?

B: yes

If on the other hand the response had been as in (24), then it would have an action type annotation. This may seem inconsistent since the two responses don't seem substantially different in meaning, but we adopted this convention for the sake of achieving consistency between multiple coders and to eliminate coding information that can be recovered by other parts of the annotation scheme.

(24) B: We can only get [two].

A general rule of thumb is that if no furniture item or furniture template is mentioned in the utterance, then it should be coded for actions. For these utterances, the annotator should only consider informtation relations for the utterance level coding (see Section A.3.3).

Note that there should be no more than 6 specific actions annotated for a dialogue. These include `SelectSofa`, `SelectTable`, `SelectChairs`, `SelectOpt` constrained to the living room, and `SelectOpt` constrained to the dining room. If an optional item is decided on for a particular room, and then later it is decided to consider an additional optional item for that same room, it should be annotated as a continuation of that same action. If it is unclear which room the additional optional item is intended for then it should be coded as a different SelectOpt with no room constraint and then once the room constraint is determined then it should be annotated as a continuation of the general and a continuation of the more specific with the first occurrence of the addition of the room constraint. If the constraint is later changed to the other room and there is no optional item in that room yet then the general action will become that more specific action for that room.

A.3.2 Information about Constraints that Limit Actions - Constraint

The `constraint` menu, allows the annotator to indicate the constraints that are placed on an action type. **Constraints should be annotated relative to the changes that have been made since the most recent description or constraint change**

for the relevant action type (e.g. the last utterance with the same action type identifier), or when a constraint setting is made explicit. Also, since an utterance can specify multiple action types, each constraint change needs to be annotated as to which action type description it applies to. This is done by including the action type identifier as part of the constraint annotation.

There is one exception for including an action type identifier in a constraint annotation. This is when there is a constraint to match colors in a room and is explained below.

An action constraint should only be annotated if there is an action introduced or continued by the utterance. If the utterance constraints an action but does not introduce or continue an action, then it should be annotated as an informational relation (see Section A.3.3. Recall that for an utterance to introduce or continue an action, it must have a furniture discourse entity in it (see Section A.3.1).

There are six constraints to be annotated. In the general case, besides the details of each constraint (which will be explained below) and which action type description it applies to, whether the constraint is inferred or not should be annotated using the selections `implicit` and `explicit`. The only case in which it is not necessary to note whether a constraint is inferred or not happens when the constraint excludes a particular furniture item (see `RuleOutValue` below). In general, it can be easy to overlook an inference that changes a constraint. For example, if you annotate a property of `matching` (see Section A.4.2) and didn't already have the room location set for the action the "matching" item is an argument for, then you should be able to infer a room location for that action.

All the other details for annotating constraints vary according to the constraint.

Constraining the room of an action type. Selecting `RoomLimit` indicates that the room where the furniture item is to be placed has been constrained. This constraint always applies to action types that have been annotated with `select` and `selectOpt` since there is no default room. For all the other action types (e.g. `selectSofa`), the `RoomLimit` constraint should only be annotated if it is made explicit as in (25). When describing the room limit constraint, the annotator should indicate which room the action is constrained to (`LR` for living room and `DR` for dining room) and whether the constraint is being applied or removed.

(25) Let's put my blue sofa in the living room

To understand what has been described so far for annotating constraints, consider a context where the two previous utterances had the action type descriptions, `selectOpt` and `selectSofa`, and where no constraints changes were annotated. Assume that the next utterance, (26), is a continuation of the two action type descriptions, `selectOpt` and

`selectSofa`. In this case the only constraint change relative to the previous `selectOpt` and `selectSofa` action descriptions is an inference that the optional rug is to be constrained to the living room. We would just annotate that a constraint has been implicitly applied to the `selectOpt` action type to limit its location to the living room. Assuming that `selectOpt` has the action type identifier `act_2`, the annotation would be `RoomLimit LR apply implicit act_2`.

(26) Let's put the rug with the sofa.

Constraining items in a room to have matching colors. Constraints involving color matches are indicated by selecting `RoomColorLimit`. In the initial case it is understood that there is a color match constraint set for both the living room and the dining room. It is more general to place a color match constraint on a room rather than on particular action types since the room might not be specified until later in the dialogue. For this reason, this is the only constraint description that excludes an action type identifier.

Since the initial case is to have a RoomColorLimit set for the living room and a separate one for the dining room, this constraint is typically removed rather than applied. Note that in Figure A.2, the dashed arrows indicate the path through the submenus for describing the `RoomColorLimit` constraint.

To see how the room color constraint interacts with other constraints, consider a case where there are yellow items in the living room and blue items in the dining room, and a blue sofa has been indicated as an optional item. One can reasonably infer that the blue sofa is for the dining room. Otherwise the color match constraint in the living room would be violated. If the blue sofa were explicitly limited to the living room, then one should annotate that the `RoomColorLimit` for the living room has been removed.

Setting an upper limit on the cost of an action type. Selecting `PriceUpperLimit` indicates that an upper limit has been placed on the price of the item selected for the furniture argument of the action type. For example, in (27) there is a constraint that the sofa price be no more than $200 for the `selectSofa` action type. Assuming that `selectSofa` has the identifier `act_11`, the annotation would be `PriceUpperLimit apply explicit act_11`.

(27) We only have $200 left for the living room.
Do you have a sofa for that price?

Since this constraint is not indicated in the initial problem description, it is typically applied rather than removed when it first arises in a dialogue. (Select `apply` to

indicate this.) However, the constraint can also be removed later should the participants find it necessary to do some backtracking. (Select `remove` to indicate this.)

Note that conceptually, (28) could be considered a `PriceUpperLimit` but since it would not have a specific action annotated for the utterance, it should not have a action constraint annotated either. However, it should have informational relation annotations (see Section A.3.3).

(28) I have $500 left.

Setting a filtering function for the cost of an action type. Selecting `PriceEvaluator` indicates that a filtering function has been set that depends on the price of the item selected for the furniture argument of the action type. For example, in (29) the filter is that the chairs should be as inexpensive as possible. A filtering function differs from setting an upper limit on the price in that it is vaguer. **The annotator should choose the constraint definition that is the most specific one possible**. Again this constraint is not indicated in the initial problem description so it will first be applied (select `apply`) and may later be removed (select `remove`) should the participants decide to backtrack.

(29) Do you have any cheap yellow or green chairs?

Constraining the properties associated with an action type argument. The participants might choose to constrain the values for one property of a furniture type argument. This is annotated by selecting `PropertyLimit`. There are three properties that can be limited in this way: color, price and type. In general, this constraint acts to specify a query template for items that are being sought. The `PropertyLimit type` will not often occur and should only be set for a `SelectOpt` action.

For example, in (30), the color property for `selectChairs` is restricted to green or yellow. This differs from a constraint to match colors in a room in that it specifies a subset of colors that are ruled out and this constraint is independent of the colors of anything else. An example for limiting the price property is (31). This differs from a constraint to set an upper limit on a price in that only sofas that cost exactly $100 are requested. This circumstance might arise when the participants are trying to spend all of their remaining money. An example for limiting the type is (32) in the context of `selectOpt`. Note that any furniture type can suffice for the `selectOpt` action type.

(30) Do you have any green or yellow chairs?

(31) Do you have a sofa for $100?

(32) Do you have any rugs or lamps?

Since these constraints aren't in effect at the initiation of the dialogue, they will first need to be applied (select `apply`) and can later be removed (select `remove`) should the participants decide to backtrack in their problem solving effort.

Distinguishing between PropertyLimit price, PriceEvaluator, and PriceUpperLimit.

If you can infer a price range for an item but not a single price value, then the annotator should choose the `PriceUpperLimit` constraint. For example, (33) should be annotated as `PriceUpperlimit`, assuming it is mutually known how much money the speaker has left.

(33) I can get the red sofa with the money I have left.

With a `PropertyLimit` on price, there should be an item or items with a specific price value or total price value that are being sought. An example of this is (34). Note that with this example, an appropriate response could be that there are n items that total $50. The key difference between the `PropertyLimit` on price and `PriceUpperLimit` is that the speaker's intention is to spend out the budget and not to buy a certain quantity of items.

(34) Do you have anything for $50 to spend out the budget?

To distinguish between `PriceUpperLimit` and `PriceEvaluator`, `PriceEvaluator` is for cases when the speaker may not know any of the prices or it is not clear to the annotator which chairs are being referred to. For example, in (35), assuming that no chairs have been introduced prior to this utterance, then it should be annotated as a `PriceEvaluator`. On the other hand, (36) should be annotated as `PriceUpperlimit` since it relates to the mutually known price of an entity. In this example, if the chairs are too expensive, then it seems reasonable to infer that speaker wants to look for something cheaper. *Your chairs* is not completely ruled out unless a set with a lower price can be identified. One could use the rule of thumb that price = some value equates to `PropertyLimit price`, price > some value equates to `PriceUpperLimit` and otherwise it is `PriceEvaluator`.

(35) Let's get cheap chairs to go with your blue table.

(36) Your chairs are too expensive.

Ruling out a particular furniture item. Selecting `RuleOutValue` indicates that a particular furniture item has been ruled out for the furniture argument of the action type.

For example, in (37), there is a unique lamp that has been explicitly ruled out. In (38), the items of the same type that have been mentioned recently in the discourse, that don't cost $50, are implicitly ruled out.

(37) We will have to forget about the lamp.

(38) I think the 50 ones are better.

This constraint can be either implicit or explicit but the difference is not noted in the annotation .

A.3.3 Informational Relations - InformationRel

An informational relation between two utterances describes how the content of the two utterance are related in the domain. **Note that informational relations should only be annotated between two different utterances or two different clauses in the same utterance and must be associated with a particular domain action**. In deciding what informational relations occur between the two utterances or clauses, it is necessary to think about whether the information in an utterance or clause specifies actions or states and how the two actions, or an action and a state, are related. If one cannot associate the informational relation to an action then that informational relation should not be annotated.

In the following annotation choices for informational relations, an action type as described in Section A.3.1 is not equivalent to an action for purposes of annotating informational relations, nor is a constraint as described in Section A.3.2 equivalent to a state. However, the annotations for action type and constraints can offer some guidance in making decisions about informational relations. As we will see below, some informational relations provide more information about the further specification of one action type. All the informationally related units should have the same action type identifier when both have action type annotations. For example, utterance units annotated with `goal` should be limited to utterances where the goal and the associated action have matching action type identifiers.

We define an action, in the case of informational relations, as an utterance that has an action type and that specifies values or potential values for the action type. Included in the informational relation annotation is the action type identifier associated with the action types involved in the informational relation and which other utterance is part of the informational relation. In addition, if a furniture entity property is made more salient or brought to mind by the informational relation, it will also be

included in the annotation. An informational relation will not always make some property more salient than another. If this is the case, do not annotate any of the properties as being more salient than the others. Examples of annotations will be included below with the descriptions for each of the informational relations.

For standardization, the annotation should always go on the clause or utterance that plays the role indicated by the relation name. Note that an informational relation can point forward or backwards in the discourse.

Effect. This unit is an effect or consequence of the action expressed in another unit. An effect is something that happens because of an action but may not be a goal of the action. For example, in (39), the first clause is the action and the second is the effect. Assuming the action type identifier is `act_10` and the first clause is `utt_10` and the second `utt_11`, the annotation on `utt_11` would be `effect of act_10 in utt_10 makesSalient price`. Again, select the action utterance that is in closest proximity to the effect utterance. This will usually be in the same turn as the effect utterance.

(39) If we buy my rug, we will be out of money.

Constraint. This unit is the constraint on an action . **This is a state that exists or needs to exist that cannot be planned for**.

As an example of a constraint informational relation, consider (40). The first utterance is the constraint and the second is the action that is being constrained. **For purposes of standardizing the annotation, the constraint utterance should be related to the action that is in closest proximity to which the constraint is meant to first be applied**.

(40) We only have $100 left.
We could get your rug instead.

Goal. The situation presented in this unit is a goal for the action presented in another unit. A goal is a desired state or a higher-level action and the action in the other unit is part of or a way of achieving the goal. For example, in (41) the first utterance is a goal and the second an action that helps in achieving the goal. Note that both utterances have an action type of `selectTable`. The first utterance cannot also be the action for the goal since it does not specify any values or potential values for the table argument. Assuming the action type identifier is `act_10` and the first utterance has the identifier `utt_10`, the annotation on `utt_10` would be; `goal of [act_10 in utt_11] makesSalient price`. The question to use in testing for what property (if any) is made salient, is to ask whether the goal of

this action brings a particular furniture property more to mind than another. For example, in (41), the price of the table in the action is made more salient than it might otherwise be without the statement of the goal.

(41) We need to decide on a cheap table.
I have a blue one for $100.

Typically if an action type is explicit or becomes explicit, as in (42), then a `goal` information relation should be annotated. In this example, both utterances have the action type `selectTable`. The first introduces it and the second continues it. While the second utterance is the `goal` it would not be annotated as the action as well although it has an appropriate action type. The action annotated should be the most specific one that is in closest proximity to the goal. The first utterance in (42) is a more specific action type since values for the action type arguments are supplied. Assuming the action type identifier is `act_10` and the first utterance has the identifier `utt_10`, the annotation on `utt_11` would be; `goal of [act_10 in utt_10] makesSalient type`.

(42) I have a blue table and a yellow table.
So let's decide on the table first.

Although a goal utterance can be related to all actions with the same action type, the annotator should just mark one such relationship. Look for the action in closest proximity to the goal that the goal is clearly meant to apply to. In ambiguous cases, generally an action utterance in the same turn as the goal utterance would be considered to be in closer proximity than an action utterance by a different speaker.

Distinguishing goal, constraint and effect informational relations can be difficult. A goal is something that the participants are working towards trying to achieve and in this domain usually indicates which action they are going to address. A constraint, on the other hand, is something that an action must comply with to qualify as an executable action. To tell a goal from an effect, if the utterance is a motivation for an action, it is a goal. If the utterance is something that results from executing the action then it is an effect. One exception is that points for items should not be annotated with informational relations. This particular type of effect is not of interest to us. **Note that an utterance can have multiple informational relations associated with it.**

An utterance can be involved in more than one informational relation. Oftentimes, if there is a `constraint` informational relation there is also an `effect` informational relation. For example, in (43), the remaining $200 is an effect of some action but is also a constraint on the action of buying for the LR.

(43) This leaves us with $200 for the LR.

A.3.4 Communicative Functions - CommFunction

We are primarily interested in one pairing of communicative functions; questions and their answers. However, we are focusing only on question / answer pairings that relate specifically to properties and property values of domain object units. **Although we recommend annotating all of the utterance level tags first, the annotator should postpone the `CommFunction` annotation until all the referential indices have been assigned (see Section A.4.3), since the referential identifier is a necessary part of the `CommFunction` annotation.**

Any utterance that directly or indirectly asks for information about a property or property value for an ActionArgument should be annotated as an `infoRequest`. An utterance should not be annotated as an `infoRequest` if it is a general request for information as in(44). The `infoRequest` feature is meant to record information requests for values of particular properties for mutually known furniture entities. A question about the existence of an item should never be annotated as an `infoRequest`. The reason for the restriction to asking about property values of mutually known items is that we are focusing of the question of why a particular property value is mentioned in a subsequent reference and not why a particular item is subsequently referred to. So in the case of a request for a property value of a mutually known item, that is an obvious reason why the proprty value would be included in a subsequent reference (i.e. the answer).

Note that questions about the existence of particular items should most likely be coded as an action constraint. For example, in (44), there should be `PropertyLimit` action constraints on the type and color.

(44) Do you have a blue rug?

As part of the `infoRequest` feature, the annotator should also indicate which furniture discourse entity the information is requested. This is accomplished by indicating the furniture discourse entity's referential identifier. Finally, we want to know which properties information is being requested on. For example, in (46), assuming that *your chair* has the referential identifier `ref_10`, the first utterance should be annotated as `infoRequest on ref_10 about color`. When the annotation is made, the tool automatically assigns a unique identifier `infoRequest_[1-n]` to each request. If information on more than one property is requested, then each property request should receive a separate annotation. Note that if the annotator cannot indicate which property information is requested on, then the `infoRequest` tag should not be assigned.

(45) What did we just buy?

(46) A: What color is your chair?

B: It is red.

To annotate an answer to a property information request, the annotator should select the utterance in closest proximity that supplies the requested information and should indicated the unique identifier, `infoRequest_[1-9]`, for the request it responds to. For example, in (46), above, assuming that the first utterance was annotated with `infoRequest_23`, the annotator would tag the second utterance with `answer infoRequest_23`.

A.4 Entity-Level Tags

The Entity-Level tags capture information about a discourse entity described in an utterance. **Each bracketed object unit within an utterance unit should be annotated with an Entity-Level tag**. We define an object unit as an NP that denotes a discourse entity of type furniture or action. Note that an NP is not the only part of an utterance that contributes to a discourse entity's description. An utterance can convey additional information about the discourse entity (e.g. in (47) the ownership of the chair is conveyed by the utterance and not by the NP *a red chair*). **So although we are annotating just the NP forms in the dialogue we will actually be recording all the relevant information about the underlying discourse entities in the utterance**. For example, in (47), the annotation for *a red chair* will include the ownership information.

(47) I have [a red chair].

See the section on file format (Section A.5) for more on how object units and utterance units are identified.

A.4.1 ActionArgument

Any object unit that further specifies the furniture argument of any action type in the *select furniture for a room* hierarchy (as described in Section A.3.1), or refers to a specific furniture entity or corefers to an entity that is an `ActionArgument`, should be annotated as an ActionArgument. This means that any utterance annotated with an `ActionArgument` should also be coded with the action it is an argument of. The reverse does not necessarily

hold since the conditions for annotating an action are less stringent with regards to the type of furniture discourse entities that must be present.

We stipulate that coreferential entities (see Section A.4.3, must also be annotated with a separate `ActionArgument` because it is possible for an entity to become an argument to a different action. For example, a participant might discuss a rug as an optional item for the living room and then if it isn't selected for the living room, he might later discuss it as an option for the dining room solution.

If multiple object units in an utterance have a subsumption relation between them or are in a predicative relationship, only the most specific ones should be coded as ActionArguments. For example, in 48, *a blue one* and *a red one*, should both be annotated as ActionArguments for `select table for dining room`. The object unit *my cheapest tables* subsumes these other two entities. Note that *a blue one* and *a red one* do not subsume one another, they are instead representational siblings. **Also, although wh-pronouns are bracketed as object units, they do not typically count as action arguments since they do not further specify the furniture argument**.

(48) [My cheapest tables] are [a blue one] and [a red one] both for $200

The object unit further specifies a furniture argument if it denotes a discourse entity of type furniture and is not a generic NP. This includes for example, pronouns and full noun phrases that refer to specific furniture items as with *a blue chair* and *it* in (49), NPs that rule out values for the furniture argument as with *mine* in (50), and non-referential NPs that put constraints on the values that can be assigned to the furniture argument as with *any blue chairs* in (51) and (52). Generic NPs, such as *tables* and *rugs* in (53), would not be annotated as ActionArguments although they would indicate a particular action type. Instead generics, like those in (53), indicate that a particular goal is being addressed.

(49) I don't have [any blue sofas],
but I have [a blue chair for $150].
Let's get it.

(50) [Your chair] is cheaper than [mine].

(51) Do you have [any blue chairs]?

(52) I don't have [any blue chairs].

(53) [Tables] are more important than [rugs].

Every object unit that is identified as an ActionArgument should be annotated as to which action type description it further specifies. The applicable action type description is the most recent, specific action type description that has a furniture argument type that subsumes that of the object unit. The unique identifier for this action type description is associated with the ActionArgument by selecting the closest identifier in annotation submenu (i.e. `act_1--act_9`) and editing it to get the correct identifier.

We add the stipulation of subsumption of the furniture type because sometimes the entity denoted by the object unit will be related to a general domain action such as *get furniture for the living room* or *get furniture for the rooms*. For example, in the constructed utterance in (54), there is an implied action type of *get furniture for the rooms*. In this case, each of the object units will be individually related to this implied action type since it is the most recent action type and the furniture type *furniture* subsumes the furniture types *sofa*, *table*, and *chair* for the three ActionArguments.

(54) I have [a $200 red sofa], [a $300 yellow table] and [4 green chairs for $25 each].

In general, there should only be one `ActionArgument` assignment per object entity. If the annotator finds a case where the domain furniture entity is initially introduced where they think the entity is a argument to more than one action then the annotator should review whether they have coded the action itself at the right level. If the domain furniture entity is mutually known to the participants (this does not equate to cases of initial reference), then it is allowable for an object entity to be the argument of more than one action. This circumstance could arise if a dialogue participant refers to an entity that has a set discourse relation to a variety of other furniture items. The entity is an initial reference but it is made up of mutually known domain entities. For example, consider (55) in a context where there are 4 chairs selected for the dining room and 2 for the living room. In this case *the chairs* are an `ActionArgument` for the actions `SelectChairs` and `SelectOpt`.

(55) The chairs cost $200 total, the sofa $250 and the table $300.

Since the annotator needs to associate an action type identifier with each ActionArgument, utterance-level action information should be coded first in order to code more efficiently. (See Section A.3.1.) Also, recall that an utterance can have multiple action types associated with it. Because of this, care should be taken to examine all the action types defined for an utterance before selecting one.

A.4.2 Property Value Codings - Properties

The Properties category codes specific property values (e.g. the color is blue) of the object units annotated as ActionArguments, that are expressed by either the containing utterance or in the noun phrase itself. At the utterance level there are three possible choices: `RelativeProps` (relative properties), `InferredProp` (inferred properties), and `uttLevelProp` (utterance level properties). At the NP level there is one choice: `NPLevelProp` (NP level properties). All of these choices are discussed in detail below. For property value codings, the goal is to record any evidence that is local to the utterance that allows us to recognize a property value for a furniture discourse entity. The property values should be noted for every bracketed entity and not just cases of initial reference. We are particularly interested in seeing which properties are repeated either implicitly (`InferredProp` and `RelativeProps`) or explicitly (`uttLevelProp` and `NPLevelProp`).

The annotation scheme allows any property value to be expressed at either the utterance or NP level, although it is more likely that some property values will be preferentially expressed at one level or the other. For example, ownership of an object is frequently expressed at the utterance level as in (56), but can be expressed at the NP level as well as in (57). At the utterance level, property values can be expressed in two special ways that are of interest to us: relative to the property values of another object (`RelativeProp`) and by inferring the property value (`InferredProp`). Other ways of expressing property values at the utterance level are not distinguished and are grouped in the general category of property values expressed at the utterance level (`uttLevelProp`). If there is a discourse inference relation (see Section A.4.4) supplied by another bracketed NP in the utterance, then no property tag should be assigned for information that is gained via the discourse inference relation.

For the general category of `uttLevelProp`, the annotator should also indicate whether the property value information is obligatory according to the syntax or semantics of the utterance by selecting `required` in the menu and `optional` otherwise.

(56) I have [a blue chair for $200].

(57) [My blue chair] is too expensive.

`RelativeProp` is used to encode when a property value is defined relative to some other already known property value, e.g. in (58), the price property value of *your chair* is compared with the price property value of another.[2] Although it is mainly the prices and

[2]Note that with the right context, (57) could be annotated with a `RelativeProp` for price. It depends on whether there is another chair in the current context.

points (note, however, that we will not annotate anything regarding points) associated with the COCONUT furniture items that will be presented in this way, other property selections have been made available for coding.

For (58) to qualify as a `RelativeProp`, the property value for the object must be defined relative to a mutually known property value. **Typically, it will not be possible to annotate a single value as part of the `RelativeProp` annotation so no actual value selection should be made for this category**. Note that (57) should not be coded as a `RelativeProp price` since the price value for the chair is not defined relative to a known price value. In this particular example, the price related information should be annotated as an implicit `PriceEvaluator` constraint (see Section A.3.2).

(58) [Your chair] is cheaper than [mine].

The utterance level manner, `InferredProp`, is provided for cases when the property value is not explicitly mentioned but a specific value is inferrable locally from the content of the utterance. For example, in a context where it is mutually known that the agents have a certain amount of money remaining, then with (59) and (60), the price of the blue chair can be inferred[3].

(59) I have [a blue chair] which leaves us with $100.

(60) I can get [a blue chair] with the money I have left.

Another example, is where the number of items is locally inferrable from the NP article or the number of the noun (e.g. singular or plural), or both. In 60, we can infer that there is 1 chair in this discourse entity. Although this may seem to be explicit information about the number of items it is not. Compare (60) with an utterance that substitutes *1* for *a*. Note that with an inferred quantity, one will generally only be able to infer quantity property values of 1 or multiple.

Property values that are inferred on the basis of an discourse inference relation (see Section A.4.4), or a coreference relation (see Section A.4.3) should not be annotated. If two referents in the same utterance are related by a coreference relation or an inference relation, then the property value codings that are shared between the two by virtue of the relation or are provided at the utterance level, need only be annotated on the more specific of the two NPs. For example, in 61, the two bracketed NPs have two separate reference identifiers and are related by the `Set` inference relation (see

[3]Note that with (60), if you can't infer a value for the price of the chair, it should be annotated as a `PriceUpperLimit` constraint. See Section A.3.3.

Section A.4.4). The first NP is more specific so it would carry the utterance level price property annotation and the second would not need to have this specified.

(61) [The yellow chairs] are $75 for [each one].

Property values for furniture items include: type of furniture, color, ownership (i.e. whose inventory it is listed in), quantity (the number of identical entities grouped in the set represented by the entity, price, and whether the item explicitly has "matching" associated with it. There is such an association if the NP contains the word "matching" or the utterance has the word "match" in it. For example, (62), the table entity would be annotated with "matching" and would also have the `inferredProp` for color with a value of red.

(62) I have 2 red chairs and a matching table.

The values for the type of furniture also includes a choice of `superordinate` for NPs that have a category that is superordinate for the base-level types. For example, *furniture*, *items*, and *stuff* are superordinates for specific furniture entities in the domain.

If no property values are expressed at a particular level or in a particular manner, then it is not necessary to code that level (where the levels are; `uttLevelProp` and `NPLevelProp`, and the manners are; `RelativeProp` and `InferredProp`). When encoding the property values for a particular level or manner, the property value is either present or absent and a binary selection should be made for each possible object property value. Note that there for an object entity, there should be no more than one value annotated for each property. Finally, there is an option to escape out of the sequence of property value selections. Escapes will be interpreted as an indication that the remaining property values are not expressed at this level.

A.4.3 Reference Relations - Reference

The reference relations were adapted from the DRAMA coding scheme and manual.

If a noun phrase introduces a discourse entity that is new to the discourse, it should be assigned a new referential index.[4] If the entity refers to a discourse entity that was referred to previously, it should be assigned the same index that was used for all previous references to the relevant entity.

[4]Note that we are not limiting the assignment of referential indices to NPs that are referring expressions.

The menu selection, `ref`, assigns a new referential index to a new discourse entity while `CorefersTo` links a discourse entity to an existing referential index. The selection, `CorefersTo` has a submenu with the referential indices `ref_[1-9]`. These values should be edited to get the correct unique referential index.

In some cases, it may be unclear whether an object unit introduces a new discourse entity. **When the annotator is unsure whether an object unit reintroduces an entity, say X, the annotator should test whether the information in the new utterance is consistent with everything that has already been said about X.**

If a single entity cannot be identified for an object unit because there are multiple possibilities that are equally likely to be what the speaker or writer intended (i.e. if there is genuine ambiguity), then the annotator should indicate the list of possibilities in the Comment tag. Note that this should be done only in cases where the annotator finds the alternatives to be truly equally likely. Keep in mind that the more natural interpretation is that the speaker intends to refer to the most recently mentioned entity.

If there is vagueness or uncertainty that prevents the annotator from determining whether an object unit refers to a particular discourse entity then no annotation should be made for coreference.

There are many cases where the entity for an object unit is very closely linked with an entity already in the discourse context, sometimes so closely that the annotator may be tempted to use `CorefersTo` to capture partial identity. However, most of these relations are captured by the annotation of inference relations, discussed in the next section.

Each bracketed entity should have either an initial reference identifier annotated or a coreference annotation (this applies to generics as well). An object entity should only have one reference relation tag annotated. Should an annotator believe that more than one coreference relation exists they should instead be annotating an initial reference and some discourse inference relations (see Section A.4.4).

(63) I have [yellow chairs] that I can buy [0].

If there is a co-reference relation to an unbracketed NP, then the annotator should go back and annotate that particular unbracketed NP. **Note that not all levels of object units are bracketed. Only the smallest non-identical subcomponents are bracketed.**[5]

To annotate an unbracketed NP, simply drag the mouse across the part of the NP that defines the entity that was referred to. For example, in 64, note that the first sentence

[5]See Section A.5 for information on how object units are identified.

has 3 NPs but only the two lowest level ones are bracketed but not the higher level NP that is the two conjoined lower level NPs. In the next sentence, however, *those*, refers to a discourse entity that is defined by this higher level NP. In this case the annotator should assign a referential index to this higher level NP and then note the co-reference relation between those and this higher level NP.

(64) I have [2 $50 green chairs] and [2 $75 red chairs].
Let's get [those].

A.4.4 Inference Relations between Discourse Entities - Inference

The inference relations for discourse entities are taken from the DRAMA coding scheme and manual and were adapted by adding some examples from the COCONUT corpus. Note that this annotation category is under the Reference relation in the annotation menus although it is not a reference relation. It is placed there because reference relations and inference relations function to relate discourse entities.

The referential identifiers discussed in the preceding section make it possible to identify all object units that are related by strict coreference. Another relation among discourse entities that constrains how a referent will be referred to is whether it can be inferred from previously mentioned referents. The types of inferential relations proposed to play a role in discourse reference include, for example, part/whole, instance/type, set/subset and so on.

There are two types of inferences that relate distinct referents: linguistic and conceptual. We will code for both but we will not make a distinction between the two types in the annotation scheme. An inference is linguistic if it is derived directly from the linguistic semantic representation of an expression, linguistic knowledge about the elements of this representation, and general inference rules. In one type of linguistic inference, the progressive use of a verb implies that the event evoked by the clause extends indefinitely into the past and present. In contrast, an inference is conceptual if it depends on world knowledge, and commonsense reasoning. Excerpt (65) illustrates both types of inference. A linguistic inference rule associated with the verb *pick* implies a change of location for the argument corresponding to the picked referent. That is, the pears that were picked have one location prior to the picking event, and a new location afterwards. World knowledge supports the conceptual inference that the default location for pears prior to the picking event is in pear trees. Thus linguistic knowledge about the argument structure of pick and commonsense knowledge that pears grow on pear trees supports the derived logical form

illustrated in (65a), namely that the man picking a set of pears is picking them from a corresponding set of pear trees.

(65) a. A man is picking some pears(j).
man(i) & pear set(j) & event(k) & is picking(k i j) & pear trees(m) & is picking from(k i j m)
b. He puts some pears(n) he has picked into a basket.
c. A boy steals the pears(n).
d. The pears(j) make a sound when the man picks them.

We will not distinguish between the two types of inference relations in the annotation. Both the conceptual and the linguistic inference relations will be grouped at the same level in the menu and relations that are only distinguished by whether they are conceptual or linguistic will use the same tag.

If there is an inference relation to an unbracketed NP, then the annotator should go back and annotate that particular unbracketed NP. Note that not all levels of object units are bracketed. Only the smallest non-identical subcomponents are bracketed.[6] Note that NPs that denote events, lists, inventories and rooms are not bracketed but may participate in some of the inference relations. **A relation to an unbracketed NP should be annotated only when it contributes to the inference of some property value for the related furniture entity**. Remember that we will not include this inferred property value as part of the related furniture entity's annotation, the property value annotation will be on the unbracketed NP.

For each inference relation, the inference relation annotation should go on the second NP in the dialogue so that the relation always points backward. This convention standardizes which of the related NPs receives the annotation.

Set: If the referent X of a discourse entity can be inferred to have one of the following relationships with referent Y of a previous discourse entity, and this relationship between X and Y has not yet been recorded, the corresponding relation between X and Y should be assigned the Set feature.

$X \in Y$, $Y \in X$
$X \subset Y$, $Y \subset X$

An example of a set/subset relation is all the pears that have been picked by a pear picker, versus one basketful of these pears.

[6]See Section A.5 for information on how object units are identified.

We expect this relation to occur quite frequently in the COCONUT domain. For example, in 65, *the green set* is a new discourse entity and has a set conceptual inference relation to the three distinct discourse entities for *2 $25 green chairs*, *2 $100 green chairs* and *$200 green table.*

(66) A: I have [2 $25 green chairs] and [a $200 green table].

a. B: I have [2 $100 green chairs].
Let's get [the green set].

Plural linguistic inference relations will also be included as part of the **Set** annotation. If a plural NP is used to refer to a set of referents, and any members of the set are already known, then the relation of the set to the members should be coded. A variety of possibilities occur, which could be encoded using set notation. The referent of a plural NP may be inferentially related to the referent of a previous singular NP, as in (67) below. Or, vice versa, the referent of a singular NP may be inferentially related to the referent of a previous plural NP, as in (68).

For example, in (67a) and (67b), two referents have been discussed, a boy (j) and a girl (k). The plural pronoun subject (m) in (67e) refers to set with members (j) and (k).

(67) a. and the little boy's-j going along,
b. and um then you see this little girl-k.
c. Coming on a bicycle in the opposite direction,
d. and uh you wonder how she-k's going to figure in on this.
e. And uh they-m come,

In (68), a set of three boys (m) is introduced first, and then a member p of m, and a subset n of m (the three boys) is later referred to. In (68), the little boy who returned the bicyclist's hat (p) is described as taking the reward of three pears back to share with his two friends (n). The singular NP subject in (68d) is related to the referent of the earlier plural NP through set membership. The plural NP in (68i) is related to the original set of three boys by set difference.

(68) a. And then he picks up uh the li
b. the three little boys-m sort of go in the opposite direction,
c. down the road, : : :
d. And the little boy-p ,

f. who fetched his hat,

g. took the pears,

h. and goes back,

i. to his friends-n,

j. and /they/ each ;p gives them-n each a pear.

In (69), from the COCONUT domain, assuming that A's red chairs have the reference identifier `ref_1` and B's `ref_2`, *the chairs* in B's second utterance would be annotated as `Set RelatesTo ref_1` and `Set RelatesTo ref_2`.

(69) A: I have [2 red chairs for $25 each].

B: I have [a red table for $125] and [2 red chairs for $50 each].
Let's definitely get [the chairs].

As noted earlier, inference relation annotations should always point back to the related entities and never forward.

PartOf: If the referent X of a furniture discourse entity can be inferred to be part of some entity Y that has already been mentioned in the discourse then this relationship should be annotated. In the COCONUT domain this happens mainly between furniture entities and unbracketed entities such as *my list*, *my inventory*, *the living room*. The PartOf annotation should be used only when the entity type of the unbracketed entity is something the related furniture entity could be part of. If there is a subsumption relation between the two discourse entities in question then the PartOf annotation should not be assigned. For example, one could consider the entity type of *inventory* to be **list**. The list entity does not subsume a furniture entity but a furniture entity could be part of what defines a particular list entity. In general, this relation would not exist between two bracketed NPs since bracketed NPs should all be of type furniture. If the annotator is considering this tag for two bracketed NPs then it is most likely a Set of Class inference relation.

The PartOf annotation will generally go on the bracketed entity since it is usually later in the discourse. This annotation should only be done if the PartOf relationship enables the inference of some property about the furniture entity. For example, in 70, the ownership of the items is inferred via the PartOf relationship to *your inventory*.

(70) From your inventory, I have 3 green chairs for $50 each and a red sofa for $200.

Class: If the referent X of a furniture discourse entity can be inferred to have a subsumption relationship with referent Y of a previous discourse entity, and this relationship

between X and Y has not yet been recorded, the corresponding relation between X and Y should be assigned the Class feature. For example, in 71, *the table* and *your green one* have a subsumption relationship.

(71) Let's decide on the table for the dining room.
How about your green one?

The `Class` annotation should always point back to an entity and should only be noted for those in close proximity to one another.

CNAnaphora: If a common noun anaphora, such as one anaphora or null anaphora, or a pronoun, is used to refer to a discourse entity or element of a discourse entity, then the relation to that entity should be coded. A simple rule of thumb is that any bracketed entity that would not receive a property value annotation for its type is generally a case of CNAnaphora. For example, in (72), each of the NPs in the second utterance has a null anaphora relation to the NP in the preceding utterance. Assuming that *a variety of high tables* has the reference identifier `ref_10`, the annotations for *green*, *red*, and *yellow* would each be `CNAnaphora RelatesTo ref_10`. Note that this example also has a `Class` relation as well.

(72) I have [a variety of high tables]
,[green], [red] and [yellow] for 400, 300, and 200.

Note that not all `CNAnaphora` relations are also coreference relations (see (73)).

(73) I have a lot of [green chairs], I have [2 $100 ones], [3 $50 ones] and [2 $125 ones].

Proposition: The referent of a discourse referential NP may depend on the interpretation of a proposition. In the COCONUT corpus, this pertains primarily to references to actions in the `select furniture for room` hierarchy. For example, in (74), *That* refers to the action type `select sofa for living room`. The annotator should only annotate the NPs that refer to actions in the action type hierarchy. Assuming that `selectSofa` has the action type identifier, `act_12`, the annotation for *that* would be `proposition assoc/w act_12`.

(74) A: Let's get my blue sofa.

B: [That] sounds good.

The annotator would not annotate *it* as a proposition in (76) since it refers to *a decision* and not an action in the action type hierarchy. Also pleonastic *it* should not be annotated as a proposition, as in (76).

(75) It is up to you.

(76) It is better to buy [a cheaper table].

This annotation should be placed on the pronoun for the propositional entity and indicate the action identifier for the action it refers to.

Predicative: If the referent of a discourse entity is related by a predicative relationship such *is*, then the two entities should be coded as having this relationship. For example, in 76, the entities defined by *my cheapest table* and *a blue one for $200* are not the same discourse entities in the definitional sense but the information about one provides more information about the other. The annotation should be placed on the second entity, *a blue one for $200.* Note that this example also includes `CNAnaphora` and `Class` inference relations.

(77) [My cheapest table] is [a blue one for $200].

When coding inference relations, there is a submenu of numbers `ref_[1-9]`. The number selected indicates the unique reference id number of the related discourse entity. If identifiers with numbers greater than 9 are needed, they can be obtained using the editor. The inference relation should only be annotated when a new discourse entity is involved.

A.5 File Format

We preprocess the files to be tagged by defining the minimal units for tagging; object units and utterance units. First we define the object units. The object units are all the NPs that denote entities of type furniture or pronouns that denote entities of type event that are in the `select furniture for room` hierarchy. The aim is to note the distinct entities defined by an NP. A set counts as a distinct entity if all the members are identical. A complex NP can often be broken into simpler NPs that define distinct entities but sometimes the complex NP defines a more distinct entity. For example in the NP *2 of the green chairs*, the entire NP should be marked as the object unit since it is more specific or limiting than the subcomponent *the green chairs* (i.e. this entity would be defined as having more than 2 green chairs). On the other hand, the NP *the green table and chairs* should be marked as two object units *table* and *chairs.* The full NP should not be delimited

as an object unit in this particular example. The portion of the NP that is an object unit will be indicated with the delimiters [ ].

An implicit argument of a verb should also be inserted as [0] whenever it is syntactically a required argument of the verb and it denotes an entity of type furniture or an entity of type event that is in the `select furniture for room` hierarchy. Note that in (77), although there is a semantically implied argument it is not syntactically a required argument. Note that for the implied argument to be syntactically realized, "than" must be added as well.

(78) I do not have a sofa for a better price.

Wh-pronouns that have intended fillers of type furniture should also be marked as object units. Missing arguments for object units of type event will not be inserted. For example, *My rug matches*, should be formatted as *[My rug] matches [0]* since the verb *matches* requires two objects as arguments. However, the deverbal *the match* should not be bracketed as an object unit or have an implicit argument added since this NP refers to a matching event.

Also, NPs such as *my inventory*, should not be delimited as object units. In this case, the NP refers to an entity of type list and not an entity of type furniture. Although this entity has furniture items as part of it, it is not representationally a furniture item.

Next we segment the dialogue into functionally independent clauses following Passonneau's approach [Pas94]. This section is taken unaltered from the COCONUT manual[DJP98]. Each clause corresponds to a single line.

Passonneau's criteria for identifying a functionally independent clause are:

- It must contribute a proposition to the discourse that is semantically complete and fully specified, and if it is a full syntactic clause it must be syntactically independent and maximal.
- It must not be a formulaic clause serving the function of an interjection.

As a consequence, functionally independent clauses include, besides main tensed clauses, coordinate clauses, subordinate adjuncts, nonrestrictive relative clauses, clauses containing verb phrase ellipsis and response fragments.

We differ from Passonneau in considering backchannels, some gerundives and some interjections as separate units, as well.

Conventions

1. *Cue words*, such as 'yes', 'no' and 'alright'. When 'yes' or 'no' are part of a response, as in 'Yes, I have a blue rug', they are are not treated as independent utterances. Cues such as 'OK' and 'alright' can similarly share a backward tag with the following utterance. But this is a matter of analysis. Therefore, in preprocessing the files, we treat 'OK' and'alright' as separate units. If the annotator considers them part of a larger response, s/he should make a segment of the cue and the rest of the response. An example in which 'OK' is part of a response:

```
S1:  (a)  I guess you can buy the yellow rug for $150.

S2: <Segment_1><Backward_1=Accept>
     (b)  OK,
     (c)  I'll buy the rug for $150. <Coref=sameItem (a)>
    </Segment_1></Backward_1>
```

   An example in which 'OK' is independent:

```
S1:  (a) What do we have left if anything?
S2:  (b) OK.
     (c) We spent: 300 (sofa), 250 (lamp), 200 (table), and 300 on
         chairs...
```

2. *Gerunds.* There appears to be two different kinds of gerunds: we call one type *resultative*, as it describes the effect of a possibly complex action, and the other *depictive*, as it denotes a state. The excerpt below provides examples of both, *resultative* in (f), and *depictive* in (b) and (d):

```
S1: (a) That means we just bought two chairs
    (b) that are yellow costing 75 a piece
    (c) and two chairs
    (d) that are green costing 100 a piece
    (e) for a total of 675,
    (f) leaving us with  275
```

   The excerpt also shows the consequences of this distinction for segmentation: only *resultative* gerunds are considered as independent utterances.

   While the distinction is not totally clear, we think it may be related to the distinction proposed in [Stu85] between *weak* and *strong* free adjuncts.

(a) Having unusually long arms, John can touch the ceiling.
(b) Standing on the chair, John can touch the ceiling.

(a) Being a businessman, Bill smokes cigars.
(b) Lying on the beach, Bill smokes cigars.

Stump calls the adjuncts in both (a) sentences *strong*, because their actual truth is uniformly entailed. and those in the (b) sentences *weak*, because their actual truth can fail to be entailed. Our examples may correlate with Stump's *weak / strong* distinction: *costing 100 a piece* would be 'strong', as it is an unchangeable property in the context of the game, whereas *leaving us with 100* would be 'weak', as its truth depends on the chosen solution.

3. *List of NPs.* When a list of items is given via coordinated NPs — e.g. *I have a green table, two sofas, one blue and one yellow, and lots of chairs* — individual NPs are not considered as separate utterances.

APPENDIX B

Exploring Internal Settings for Identification within Intentional Influences

Here we report the results of the experiments we conducted in order to determine the best internal settings for the identification algorithms when they are used within the **intentional influences** algorithm. Recall from Chapter 7 that there are four identification algorithms that we must determine the internal settings for (GEST+, LEX+, IDAS and DIFF).

For each of these identification algorithms, when used within the **intentional influences** algorithm, we varied the following internal settings:

- GEST+: distractor set, template
- LEX+: distractor set
- IDAS: distractor set
- DIFF: distractor set, threshold

We will group the discussion of the experiments for these factors by the identification algorithm.

B.1 GEST+ algorithm

When we varied the distractor set definition for GEST+ within **intentional influences** we found an ANOVA of ($F = 2.13, p < .075$). This indicates that there is no significant difference between the distractor set definitions. Looking at the MCA comparison shown in Figure B.1 we found that SEG+ has a trend towards better performance.

When we varied the search template definition for GEST+ within **intentional influences** we found an ANOVA of ($F = 57, p < 0$). This indicates that there is a significant difference in performance between the search templates. Looking at the MCA comparison

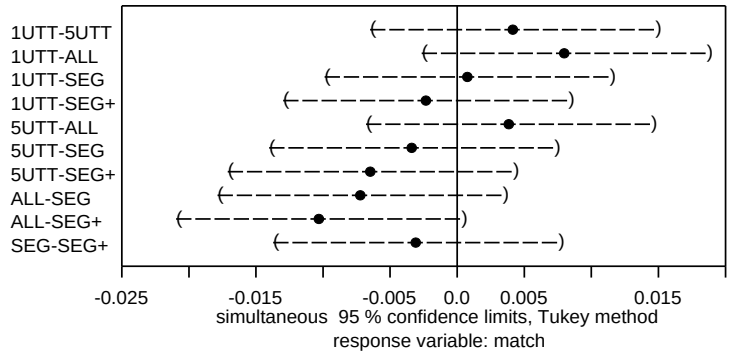

Figure B.1: Multiple Comparisons of Distractor Set for Gest+

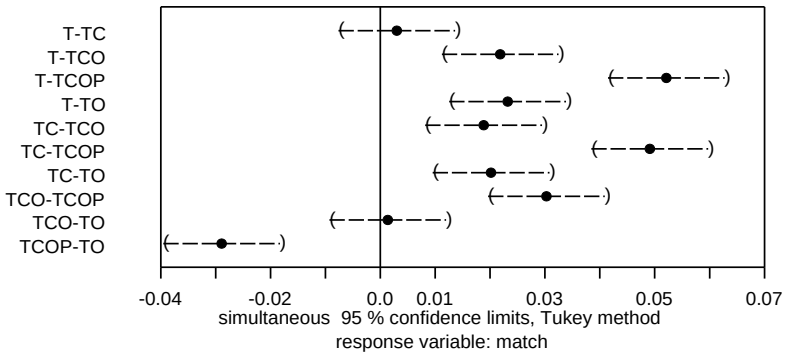

Figure B.2: Multiple Comparisons of Search Template for Gest+

shown in Figure B.2 we found that a search template of T (type) either performs better or has a trend towards better performance when compared to all the other templates.

B.2 LEX+ algorithm

When we varied the distractor set definition for LEX+ within **intentional influences** we found an ANOVA of ($F = .37, p < .83$). This indicates that there is no significant difference between the distractor set definitions. Looking at the MCA comparison shown in Figure B.3 we found that 5UTT has a trend towards better performance.

B.3 IDAS algorithm

When we varied the distractor set definition for IDAS within **intentional influences** we found an ANOVA of ($F = .38, p < .82$). This indicates that there is no significant difference between the distractor set definitions. Looking at the MCA comparison shown in Figure B.4 we found that SEG+ has a trend towards better performance.

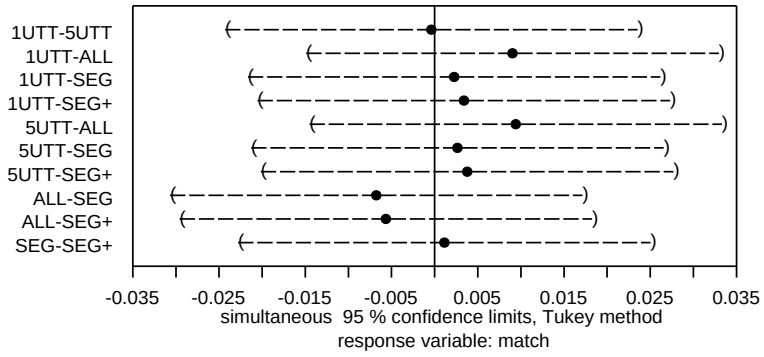

Figure B.3: Multiple Comparisons of Distractor Set for Lex+

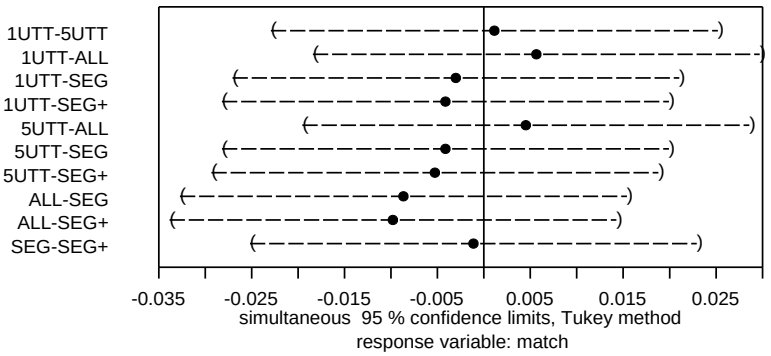

Figure B.4: Multiple Comparisons of Distractor Set for IDAS

B.4 DIFF algorithm

When we varied the distractor set definition for DIFF within **intentional influences** we found an ANOVA of ($F = 11.1, p < 0$). This indicates that there is a significant difference in performance between the distractor set definitions. Looking at the MCA comparison shown in Figure B.5 we found that 1UTT performs better than all the other distractor set definitions.

When we varied the saliency threshold for DIFF within **intentional influences** we found an ANOVA of ($F = .375, p < 77$). This indicates that there is no significant difference in performance when varying the threshold. Looking at the MCA comparison shown in Figure B.6 we found that MEDH has a trend towards better performance.

B.5 The Best Settings

The best internal settings for the identification algorithms were different from those we found when the algorithms were used in isolation:

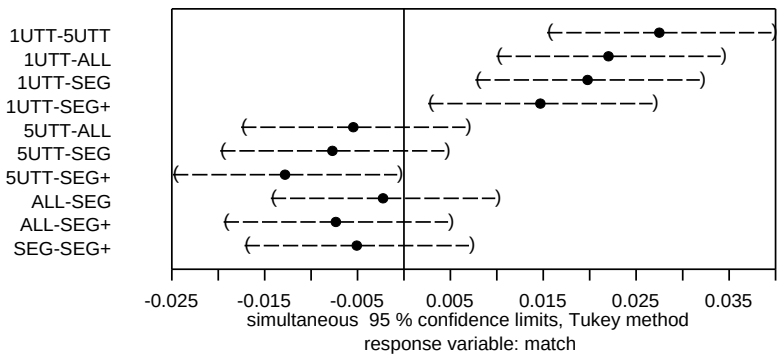

Figure B.5: Multiple Comparisons of Distractor Set for DIFF

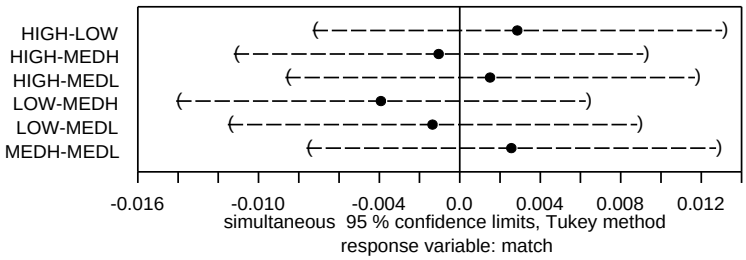

Figure B.6: Multiple Comparisons of Saliency Threshold for DIFF

- GEST+: distractor set=SEG+, template=T(ype)
- LEX+: distractor set=5UTT
- IDAS: distractor set=SEG+
- DIFF: distractor set=1UTT, threshold=MEDH

BIBLIOGRAPHY

BIBLIOGRAPHY

[AK87] Douglas Appelt and Amichai Kronfeld. A computational model of referring. In *Proceedings of IJCAI 87*, 1987.

[And93] John R. Anderson. *Rules of the Mind.* Lawrence Erlbaum, Hillsdale, NJ, 1993.

[App85a] Douglas Appelt. *Planning English Sentences.* Studies in Natural Language Processing. Cambridge University Press, 1985.

[App85b] Douglas E. Appelt. Planning English referring expressions. *Artificial Intelligence*, 26(1):1–33, April 1985.

[App85c] Douglas E. Appelt. Some pragmatic issues in the planning of definite and indefinite noun phrases. In *Proceedings of 23rd ACL*, 1985.

[BPC94] Martha S. Bean and G. Genevieve Patthey-Chavez. Repetition in instuctions discourse: A means for joint cognition. In Barbara Johnstone, editor, *Repetition in Discourse: Interdisciplinary Perspectives, Volume 1*, volume XLVII of *Advances in Discourse Processes*, chapter 14. Ablex, 1994.

[Bre90] Susan E. Brennan. *Seeking and Providing Evidence for Mutual Understanding.* PhD thesis, Stanford University Psychology Dept., 1990. Unpublished Manuscript.

[Byr96] Michael D. Byrne. *A Computational Theory of Working Memory.* PhD thesis, Georgia Institute of Technology, Aug 1996.

[Car92] Jean C. Carletta. *Risk Taking and Recovery in Task-Oriented Dialogue.* PhD thesis, Edinburgh University, 1992.

[Car96] Jean Carletta. Assessing agreement on classification tasks: the kappa statistic. *Computational Lingustics*, 22(2):249–254, 1996.

[CDE+99] Lynne Cahill, Christy Doran, Roger Evans, Chris Mellish, Danile Paiva, Mike Reape, Donia Scott, and Neil Tipper. In search of a reference architecture for NLG systems. In *Proceedings of the European Workshop on Natural Language Generation*, Toulouse France, May 1999.

[Cha80] Wallace L. Chafe. *The Pear Stories: Cognitive, cultural and linguistic aspects of narrative production.* Ablex Publishing Corp., Norwood, NJ, 1980.

[Cla96] Herbert H. Clark. *Using Language.* Cambridge University Press, 1996.

[CM81] Herbert H. Clark and Catherine R. Marshall. Definite reference and mutual knowledge. In Arivind Joshi, Bonnie Webber, and Ivan Sag, editors, *Linguistics Structure and Discourse Setting*, pages 10–63. Cambridge University Press, Cambridge, England, 1981.

[Coh81] Philip R. Cohen. The need for referent identification as a planned action. In *Proceedings of IJCAI 81*, pages 31–36, 1981.

[Coh95] Paul R. Cohen. *Empirical Methods for Artificial Intelligence.* MIT Press, Boston, 1995.

[Cre96] Anita H.M. Cremers. *Reference to Objects : an Empirically Based Study of Task-Oriented Dialogues.* PhD thesis, Technical University of Eindhoven, 1996.

[CS87] Herbert H. Clark and Edward F. Schaefer. Collaborating on contributions to conversations. *Language and Cognitive Processes*, 2:19–41, 1987.

[CS89] Herbert H. Clark and Edward F. Schaefer. Contributing to discourse. *Cognitive Science*, 13:259–294, 1989.

[CSB83] H.H. Clark, R. Schreuder, and S. Buttrick. Common ground and the understanding of demonstrative reference. *Journal of Verbal Learning and Verbal Behavior*, 22:245–258, 1983.

[CWG86] Herbert H. Clark and Deanna Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22:1–39, 1986.

[Dal89] Robert Dale. Cooking up referring expressions. In *Proceedings of the Twenty-Seventh Annual Meeting of the Association for Computational Linguistics*, pages 68–75, 1989.

[Dal92] Robert Dale. *Generating Referring Expressions.* ACL-MIT Series in Natural Language Processing. The MIT Press, 1992.

[Dal95] Robert Dale. Generating one-anaphoric expressions: Where does the decision lie? In *Working Papers of PACLING-II*, pages 49–58, 1995.

[Deu76] W. Deutsch. *Sprachliche Redundanz und Objektidentifikation.* PhD thesis, University of Marburg, 1976.

[Di 98] Barbara Di Eugenio. An action representation formalism to interpret Natural Language instructions. *Computational Intelligence*, 14(1), February 1998. In press.

[DJP98] Barbara Di Eugenio, Pamela W. Jordan, and Liina Pylkkänen. The COCONUT project: Dialogue annotation manual. Technical Report ISP Technical Report 98-1, University of Pittsburgh, December 1998. [On-line] Available: http://www.isp.pitt.edu/~intgen/research-papers.

[DJTM00] Barbara Di Eugenio, Pamela W. Jordan, Richmond H. Thomason, and Johanna D. Moore. The agreement process: An empirical investigation of human-human computer-mediated collaborative dialogues. *To Appear in International Journal of Human-Computer Studies*, 2000.

[DMEPS89] Kathleen Dahlgren, Joyce McDowell, and Jr. Edward P. Stabler. Knowledge representation for commonsense reasoning with text. *Computational Linguistics*, 15(3), 1989.

[Don66] Keith S. Donnellan. Reference and definite description. *Philosophical Review*, 75:281–304, 1966.

[DR95] Robert Dale and Ehud Reiter. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263, Apr–June 1995.

[Dun64] C.W. Dunnett. New table for multiple comparisons with a control. *Biometrics*, 20:482–491, 1964.

[DW96] Barbara Di Eugenio and Bonnie Webber. Pragmatic overloading in natural language instructions. *International Journal of Expert Systems, Special Issue on Knowledge Representation and Reasoning for Natural Language Processing*, 9(1):53–84, March 1996.

[Edm93] Philip Edmonds. A computational model of collaboration on reference in direction-giving dialogues. Technical Report CSRI-289, Department of Computer Science, University of Toronto, 1993. MSc thesis.

[Edm94] Philip Edmonds. Collaboration on reference to objects that are not mutually known. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, pages 1118–1122, 1994.

[Fla95] Giovanni Flammia. N.b.: A graphical user interface for annotating spoken dialogue. In *AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 40–46, Stanford, CA, 1995.

[GA87] S. Garrod and A. Anderson. Saying what you mean in dialogue: A study in conceptual and semantic co-ordination. *Cognition*, 27:181–218, 1987.

[GJW83] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Providing a unified account of definite noun phrases in discourse. In *Proc. 21st Annual Meeting of the ACL, Association of Computational Linguistics*, pages 44–50, 1983.

[Gri75] H.P. Grice. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics III - Speech Acts*, pages 41–58. Academic Press, New York, 1975.

[Gro77] Barbara J. Grosz. The representation and use of focus in dialogue understanding. Technical Report 151, Artificial Intelligence Center, SRI International, 333 Ravenswood Ave, Menlo Park, Ca. 94025, 1977.

[GS86] Barbara J. Grosz and Candace L. Sidner. Attentions, intentions and the structure of discourse. *Computational Linguistics*, 12:175–204, 1986.

[GSGF93] F. Giunchiglia, L. Serafini, E. Giunchiglia, and M. Frixione. Non-omniscient belief as context-based reasoning. In *IJCAI 93*, pages 548–554, Chambery, France, 1993.

[Haw78] J. A. Hawkins. *Definiteness and Indefiniteness.* Humanities Press, Atlantic Highlands, NJ, 1978.

[Hei82] Irene Heim. *The Semantics of Definite and Indefinite Noun Phrases.* PhD thesis, University of Massachusets, Amherst, 1982.

[Hei83] Irene Heim. File change semantics and the theory of definiteness. In R. Bauerle, C. Schwarze, and A von Stechow, editors, *Meaning, Use, and the Interpretation of Language.* Walter de Gruyter, Berlin, 1983.

[HH95] Peter A. Heeman and Graeme Hirst. Collaborating on referring expressions. *Computational Lingusitics*, 21(3), 1995.

[Hor97] Helmut Horacek. An algorithm for generating referential descriptions with flexible interfaces. In *Proceedings of the 35th Annual Meeting of ACL and 8th Conference of EACL*, pages 206–213, 1997.

[Hov88] Eduard H. Hovy. *Generating Natural Language Under Pragmatic Constraints.* Lawrence Erlbaum, Hillsdale, NJ, 1988.

[Hov91] Eduard H. Hovy. Approaches to the planning of coherent text. In Cecile L. Paris, William R. Swartout, and William C. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pages 83–102. Kluwer Academic Publishers, Boston, 1991.

[Hsu96] Jason C. Hsu. *Multiple Comparisons: Theory and Methods.* Chapman and Hall, London, 1996.

[IC87] E. A. Issacs and Herbert H. Clark. References in conversation between experts and novices. *Journal of Experimental Psychology: General*, 116:26–37, 1987.

[JC92] Marcel A. Just and Pat A. Carpenter. A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, 99:123–148, 1992.

[JD97] Pamela W. Jordan and Barbara Di Eugenio. Control and initiative in collaborative problem solving dialogues. In *Computational Models for Mixed Initiative Interaction. Papers from the 1997 AAAI Spring Symposium. Technical Report SS-97-04*, pages 81–84. The AAAI Press, 1997.

[Joh94] Barbara Johnstone. Repetition in discourse: A dialogue. In Barbara Johnstone, editor, *Repetition in Discourse: Interdisciplinary Perspectives, Volume 1*, volume XLVII of *Advances in Discourse Processes*, chapter 1. Ablex, 1994.

[Jos78] Aravind K. Joshi. Some extensions of a system for inference on partial information. In *Pattern Directed Inference Systems*, pages 241–257. Academic Press, 1978.

[Jos96] David Joslin. *Passive and Active Decision Postponement in Plan Generation.* PhD thesis, University of Pittsburgh, Intelligent Systems Program, 1996.

[JW96] Pamela W. Jordan and Marilyn A. Walker. Deciding to remind during collaborative problem solving: Empirical evidence for agent strategies. In *Proceedings of AAAI-96.* AAAI Press, 1996.

[JWW86] Aravind K. Joshi, Bonnie Lynn Webber, and Ralph M. Weischedel. Some aspects of default reasoning in interactive discourse. Technical Report MS-CIS-86-27, University of Pennsylvania, Department of Computer and Information Science, 1986.

[Kam85] Megumi Kameyama. *Zero Anaphora: the Case of Japanese.* PhD thesis, Stanford University, Linguistics Department, 1985.

[Kam93] Hans Kamp. *From Discourse to Logic; Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Kluwer Academic Publishers, Dordrecht Holland, 1993.

[Kar76] Lauri Karttunen. Discourse referents. In J. McCawley, editor, *Syntax and Semantics*, volume 7. Academic Press, 1976.

[Kep91] Geoffrey Keppel. *Design and Analysis: A Researcher's Handbook, 3rd edition.* Prentice Hall, New Jersey, 1991.

[KGBM77] R. M. Krauss, C.M. Garlock, P.D. Bricker, and L.E. McMahon. The role of audible and visible back-channel responses in interpersonal communication. *The Journal of Personality and Social Psychology*, 35(7):523–529, 1977.

[Kri80] Klaus Krippendorff. *Content Analysis: an Introduction to its Methodology.* Beverly Hills: Sage Publications, 1980.

[Kro86] Amichai Kronfeld. Donnellan's distinction and a computational model of reference. In *Proceedings of 24th ACL*, 1986.

[KW66] R. M. Krauss and S. Weinheimer. Concurrent feedback, confirmation, and the encoding of referents in verbal communication. *The Journal of Personality and Social Psychology*, 4:343–346, 1966.

[KW67] R. M. Krauss and S. Weinheimer. Effect of referent similarity and communication mode on verbal encoding. *The Journal of Verbal Learning and Verbal Behavior*, 6:359–363, 1967.

[Lan75] Thomas K. Landauer. Memory without organization: Properties of a model with random storage and undirected retrieval. *Cognitive Psychology*, pages 495–531, 1975.

[Lev89] W. J. M. Levelt. *Speaking: From Intention to Articulation.* MIT Press, 1989.

[Loc93] Karen Lochbaum. A Collaborative Planning Approach to Discourse Understanding. Technical Report TR-20-93, Center for Research in Computing Technology, Harvard University, 1993.

[Loc95] Karen Lochbaum. The use of knowledge preconditions in language processing. In *IJCAI95*, 1995.

[Lot96] Claudio Lottaz. Constraint Solving, Preference Activation and Solution Adaptation in IDIOM. Technical Report 96/204, Artificial Intelligence Laboratory, Swiss Federal Institute of Technology in Lausanne, Switzerland, 1996.

[LS97] Claudio Lottaz and Ian Smith. Collaborative design using constraint solving. From Swiss Workshop on Collaborative and Distributed Systems, Lausanne Switzerland. See http://liawww.epfl.ch/ lottaz/ICCS/Design_and-_CSP/design_and_CSP.html and http://liawww.epfl.ch/ lottaz/ICCS/Collaboration/index.html, May 1997.

[Lup91] Susann Luperfoy. *Discourse Pegs: A Computational Analysis of Context-Dependent Referring Expressions.* PhD thesis, Department of Linguistics, University of Texas at Austin, 1991.

[Lyo95] Kevin W. Lyons. Collaborative design for assembly of complex electromechanical products. Presentation abstract for NCMS Manufacturing Technical Conference. Available: http://elib.cme.nist.gov/made/presentations-/ncms.html, 1995.

[Man86] R. Mangold. *Sensorische Faktoren beim Verstehen uberspezifizierter Objekt-Bennennungen.* Peter Lang: Frankfurt, 1986.

[Mat98] MathSoft, Inc., Seattle, Washington. *S-Plus 5 for Unix Guide to Statistics*, September 1998.

[MB95] John McCarthy and Saša Buvač. Formalizing context (expanded notes). Available from http://www-formal.stanford.edu/buvac., 1995.

[McC86] Kathleen McCoy. The ROMPER system: Responding to object-related misconceptions using perspective. In *Proceedings of the 24th ACL*, 1986.

[McK85] Kathleen R. McKeown. *Text Generation. Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge University Press, 1985.

[MM95] Megan Moser and Johanna D. Moore. Investigating cue placement and selection in tutorial discourse. In *Proceedings of 33rd Annual Meeting of the Association for Computational Linguistics*, pages 130–135, 1995.

[MP89] Johanna D. Moore and Cécile L. Paris. Planning text for advisory dialogues. In *Proc. 27th Annual Meeting of the Association of Computational Linguistics*, Vancouver, 1989. ACL.

[MP92] Johanna D. Moore and Martha E. Pollack. A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics*, 18(4), 1992.

[MP93] Johanna D. Moore and Cécile L. Paris. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4), 1993.

[MT87a] W.C. Mann and S.A. Thompson. Rhetorical Structure Theory: A Framework for the Analysis of Texts. Technical Report RS-87-190, USC/Information Sciences Institute, 1987.

[MT87b] Christian Matthiessen and Sandra A. Thompson. The Structure of Discourse and Subordination. Technical Report ISI/RS-97-183, ISI, USC, 1987.

[MWM85] K.R. McKeown, M. Wish, and K. Matthews. Tailoring explanations for the user. In *Proceedings of the 9th International Conference on Artificial Intelligence*, pages 794–798, 1985.

[OC91] Sharon L. Oviatt and Philip R. Cohen. Discourse structure and performance: Efficiency in interactive and non-interactive spoken modalities. *Computer Speech and Language*, pages 297–326, 1991.

[OWW93] Brid O'Conaill, Steve Whittaker, and Sylvia Wilbur. Conversations over video conferences: An analysis of the spoken aspects of video mediated communication. *To Appear in Human Computer Interaction*, 1993.

[Pas94] Rebecca J. Passonneau. Protocol for coding discourse referential noun phrases and their antecedents. Technical report, Columbia University, 1994.

[Pas95] Rebecca J. Passonneau. Integrating Gricean and attentional constraints. In *Proceedings of IJCAI 95*, 1995.

[Pas96] Rebecca J. Passonneau. Using centering to relax Gricean informational constraints on discourse anaphoric noun phrases. *Language and Speech*, 39(2-3):229–264, 1996.

[Poe93] Massimo Poesio. A situation-theoretic formalization of definite description interpretation in plan elaboration dialogues. In P. Aczel, D. Israel, Y. Katgiri, and S. Peters, editors, *Situation Theory and its Applications*, volume 3, pages 339–374. 1993.

[Pol91] Martha E. Pollack. Overloading intentions for efficient practical reasoning. *Noûs*, 25:513 – 536, 1991.

[Pol92] Martha E. Pollack. The uses of plans. *Artificial Intelligence*, 57(1):43–69, 1992.

[Pri81] Ellen F. Prince. Toward a taxonomy of given-new information. In *Radical Pragmatics*, pages 223–255. Academic Press, 1981.

[PS84] Livia Polanyi and Remko Scha. A syntactic approach to discourse semantics. In *COLING84*, 1984.

[Qu97] Yan Qu. A Constraint-Based Model for Cooperative Response Generation in Information Systems. PhD Proposal, Computational Linguistics Program, Carnegie Mellon University, 1997.

[RDM99] C. P. Rosé, B. Di Eugenio, and J. D. Moore. A dialogue based tutoring system for basic electricity and electronics. In *Proceedings of AI in Education*, 1999.

[Rei85] Rachel Reichman. *Getting Computers to Talk Like You and Me.* MIT Press, Cambridge, MA, 1985.

[Rei90a] Ehud Reiter. Generating Appropriate Natural Language Object Descriptions. Technical Report TR-10-90, Department of Computer Science, Harvard University, 1990. PhD thesis.

[Rei90b] Ehud Reiter. Generating descriptions that exploit a user's domain knowledge. In R. Dale, C. Mellish, and M. Zock, editors, *Current Research in Natural Language Generation.* Academic Press, London, 1990.

[SC88] Sidney Siegel and N. John Castellan, Jr. *Nonparametric Statistics for the Behavioral Sciences*. McGraw Hill, 1988.

[SC89] Michael F. Schober and Herbert H. Clark. Understanding by addressees and overhearers. *Cognitive Psychology*, 21:211–232, 1989.

[Sea69] John Searle. *Speech Acts*. Cambridge Univesity Press, Cambridge, England, 1969.

[SM98] L. Sherry and K.M. Myers. The dynamics of collaborative design. *IEEE Transactions on Professional Communication*, 41(2):123–139, 1998.

[Son82] S. Sonnenschein. The effects of redundant communications on listeners: When more is less. *Child Development*, 53:717–729, 1982.

[Son84] S. Sonnenschein. The effect of redundant communications on listeners: Why different types may have different effects. *Journal of Psycholingistic Research*, 13:147–166, 1984.

[Stu85] Gregory Stump. *The Semantic Variability of Absolute Constructions*. D. Reidel Publishing Company, 1985.

[SW98] Matthew Stone and Bonnie Webber. Textual economy through close coupling of syntax and semantics. In *Proceedings of 1998 International Workshop on Natural Language Generation*, Niagra-on-the-Lake, Canada, 1998.

[Ter85] J. M. B. Terken. *Use and Function of Accentuation: Some Experiments*. PhD thesis, Institute for Perception Research, Eindhoven, The Netherlands, 1985.

[Wal92] Marilyn A. Walker. Redundancy in collaborative dialogue. In *Fourteenth International Conference on Computational Linguistics*, pages 345–351, 1992.

[Wal93] Marilyn A. Walker. *Informational Redundancy and Resource Bounds in Dialogue*. PhD thesis, University of Pennsylvania, December 1993.

[Wal94] Marilyn A. Walker. Experimentally evaluating communicative strategies: The effect of the task. In *AAAI94*, 1994.

[Wal96a] Marilyn A. Walker. The effect of resource limits and task complexity on collaborative planning in dialogue. *Artificial Intelligence*, 1996. to appear.

[Wal96b] Marilyn A. Walker. Inferring acceptance and rejection in dialogue by default rules of inference. *Language and Speech*, 39(2), 1996.

[Wal96c] Marilyn A. Walker. Limited attention and discourse structure. *Computational Linguistics*, 22(1), 1996.

[WB92] Bonnie Lynn Webber and Breck Baldwin. Accommodating context change. In *ACL92, Proceedings of the 30th Meeting of the Association for Computational Linguistics*, pages 96–103, 1992.

[WBO99] Janyce M. Wiebe, Rebecca F. Bruce, and Thomas P. O'Hara. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the Thirty-Seventh Annual Meeting of the Association of Computational Linguistics*, pages 246–253, 1999.

[Web78] Bonnie Lynn Webber. *A Formal Approach to Discourse Anaphora.* PhD thesis, Harvard University, 1978. Garland Press.

[WGR93] Steve Whittaker, Erik Geelhoed, and Elizabeth Robinson. Shared workspaces: How do they work and when are they useful? *IJMMS*, 39:813–842, 1993.

[Whi95] Steve Whittaker. Rethinking video as a technology for interpersonal communications: theory and design implications. *International Journal of Human-Computer Studies*, 42:501–529, 1995.

[WJP97] Marilyn Walker, Arivind Joshi, and Ellen Prince. Centering in naturally occurring discourse: An overview. In Marilyn A. Walker, Arivind K. Joshi, and Ellen Prince, editors, *Centering Theory in Discourse*, pages 1–28. Oxford University Press, Oxford, 1997.

[WLKA97] Marilyn A. Walker, Diane J. Litman, Candace Kamm, and Alicia Abella. Paradise: A framework for evaluating spoken dialogue agents. In *35th Meeting of ACL and 8th Conference of the EACL*, pages 271–280, 1997.

[WS88] Steve Whittaker and Phil Stenton. Cues and control in expert client dialogues. In *Proc. 26th Annual Meeting of the ACL, Association of Computational Linguistics*, pages 123–130, 1988.

[YM94] R. Michael Young and Johanna D. Moore. DPOCL: A principled approach to discourse planning. In *Seventh International Workshop on Natural Language Generation*, pages 13–20, Kennebunkport, Maine, 1994.

[YM97] Ching-Long Yeh and Chris Mellish. An empirical study on the generation of anaphora in chinese. *Computational Linguistics*, 23(1):169–190, 1997.