Interactive generation of effective discourse in situated context

A planning-based approach

Konstantina Garoufi

Dissertation eingereicht
bei der Humanwissenschaftlichen Fakultät
der Universität Potsdam

2013

Betreuer: Prof. Dr. Alexander Koller
Gutachter: Prof. Dr. Manfred Stede
Prof. Dr. Emiel Krahmer
Datum der mündlichen Prüfung: 13. Dezember 2013

Published online at the
Institutional Repository of the University of Potsdam:
URL http://opus.kobv.de/ubp/volltexte/2013/6910/
URN urn:nbn:de:kobv:517-opus-69108
http://nbn-resolving.de/urn:nbn:de:kobv:517-opus-69108

Ich erkläre hiermit, dass die Arbeit selbständig und ohne unzulässige Hilfe Dritter verfasst wurde und bei der Abfassung nur die in der Dissertation angegebenen Hilfsmittel benutzt sowie alle wörtlich oder inhaltlich übernommenen Stellen als solche gekennzeichnet wurden.

Potsdam, 2013

Konstantina Garoufi

Abstract

As our modern-built structures are becoming increasingly complex, carrying out basic tasks such as identifying points or objects of interest in our surroundings can consume considerable time and cognitive resources. In this thesis, we present a computational approach to converting contextual information about a person's physical environment into natural language, with the aim of helping this person identify given task-related entities in their environment. Using efficient methods from automated planning—the field of artificial intelligence concerned with finding courses of action that can achieve a goal—, we generate discourse that interactively guides a hearer through completing their task. Our approach addresses the challenges of controlling, adapting to, and monitoring the situated context.

To this end, we develop a natural language generation system that plans how to manipulate the non-linguistic context of a scene in order to make it more favorable for references to task-related objects. This strategy distributes a hearer's cognitive load of interpreting a reference over multiple utterances rather than one long referring expression. Further, to optimize the system's linguistic choices in a given context, we learn how to distinguish speaker behavior according to its helpfulness to hearers in a certain situation, and we model the behavior of human speakers that has been proven helpful. The resulting system combines symbolic with statistical reasoning, and tackles the problem of making non-trivial referential choices in rich context. Finally, we complement our approach with a mechanism for preventing potential misunderstandings after a reference has been generated. Employing remote eye-tracking technology, we monitor the hearer's gaze and find that it provides a reliable index of online referential understanding, even in dynamically changing scenes. We thus present a system that exploits hearer gaze to generate rapid feedback on a per-utterance basis, further enhancing its effectiveness.

Though we evaluate our approach in virtual environments, the efficiency of our planning-based model suggests that this work could be a step towards effective conversational human-computer interaction situated in the real world.

Acknowledgments

How can I generate the words to express my gratitude to so many people who helped so much? Knowing that this discourse will remain ineffective, I will try my best to acknowledge the valuable contributions of others to my work.

I feel extremely lucky to have been advised by Alexander Koller, whom I knew I wanted to work with, ever since I first met him as a lecturer of semantics a while ago. Over the years, Alexander has spent countless hours advising me on just about every aspect of scientific life—identifying research questions and answers with clarity, implementing ideas and troubleshooting technical problems, presenting those ideas in a more accessible way. I could fill pages listing Alexander's contributions to this work, which would just not have been possible without him. But since one of the things he has advised me on is to adhere to the rule of Three, let me simply say: Thank you, Alexander, for your inspiring enthusiasm about "efficient algorithms in computational linguistics", for your critical comments that always made my work better, and for your contagious laugh.

It has been a pleasure to work together with a number of other researchers, who have contributed significantly to this thesis. Maria Staudte and Matthew Crocker, thank you for guiding me through the exciting world of psycholinguistics, for providing "rapid and reliable feedback" through varying time zones, and for making every meeting easygoing and productive at the same time. To my other collaborators, Andrew Gargett, Kristina Striegnitz, Alexandre Denis and Mariët Theune, thank you for generously contributing your time and competence that enabled much of this research, especially during the intense periods of collecting the GIVE-2 Corpus and running the GIVE-2.5 Challenge. Many thanks, also, to my slightly less systematic collaborators Luciana Benotti and Ivan Titov, who with their thoughtful comments often helped me to better understand my work.

This work was carried out within a wonderful technological framework. It may never have been possible if it weren't for Jörg Hoffmann providing support

for his FF, offering tips on how to optimize my use of this outstanding planning system, and ultimately getting his hands dirty to tweak it so that it works beautifully in my domains. To everyone involved in the development of the GIVE and CRISP software—especially Daniel Bauer—, thank you for contributing to these useful tools. And thanks to the hundreds of computer users who went on a virtual treasure hunt to make my data-collection efforts successful.

To my reviewers Manfred Stede and Emiel Krahmer, thank you for your interest and for so kindly agreeing to become involved; I have benefited from discussions with both of you. I have also benefited from the feedback of numerous reviewers and participants at conferences, workshops, and colloquiums, where I presented portions of this work, and I thank them all. The diligent reviewers and editors—especially Albert Gatt and Claire Gardent—of the journals where parts of this thesis will appear have helped me improve my writings substantially, and I appreciate that the publishers let me retain the right to include them here. I feel particularly obliged to the vibrant Special Interest Group on Discourse and Dialogue, which has honored my work with its conference's best paper award.

My awesome former and current colleagues at the Saarland University and the University of Potsdam have helped me not only by means of scientific discussions (thank you, Timo Baumann, Okko Buß, Christian Chiarcos, Tatjana Scheffler, Sebastian Varges, and so many others) but also in practical ways. I particularly owe to my office mates Nikos Engonopoulos and Martín Villalba—as well as our "special guests" Thomas Hanneforth, Argyro Katsika, Stavroula Sotiropoulou, and Siggi Wrobel—for making my hours in the office (and outside) much more fun. I am thankful to the Cluster of Excellence on Multimodal Computing and Interaction and to the Collaborative Research Center on Information Structure for financial support that enabled me to focus on research for much of my working time. For the rest of the time, I thank my students and especially those in my seminar on planning-based models of communication, who provided me with an opportunity to think more deeply about these topics. Special thanks to Kira Eberle, who chose planning-based generation of discourse as the topic of her Master's thesis and engaged in a thought-provoking endeavor to extend some of this work.

It has been a great privilege to talk with several people who were so friendly as to share their thoughts and ideas. Thank you to the invited speakers at my institutes John Kelleher, Stefan Kopp, Ron Petrick, Verena Rieser, David Schlangen, Matthew Stone, and many others. Special thanks to Hendrik Buschmeier, Nina Dethlefs, and Klaus-Peter Engelbrecht, who accepted my invitations to come to Potsdam and allowed me to indulge in stimulating discussions. Many thanks

to Srini Janarthanam, Aasish Pappu, Ivandré Paraboni, Niels Schütte, and Laura Stoia for eagerly emailing me insightful details about their work.

I wish to extend a big thank-you to the people behind two groundbreaking research and education services. First, the Massive Open Online Course community made it awfully easy for me to broaden my views in my areas of interest—I was inspired by Sebastian Thrun and Peter Norvig's "AI class", I learned a lot in Gerhard Wickler and Austin Tate's "AI planning" as well as Kristin Sainani's "Writing in the sciences" classes, and I was reminded to keep thinking independently by Walter Sinnott-Armstrong and Ram Neta's "Think again" class. Second, my awareness of scientific work relevant to my own across different areas and disciplines would have been much smaller, were it not for the brilliant Google Scholar Updates regularly serving me up with the most interesting papers right at my desk.

I might have never explored any of these fascinating topics if a few people hadn't urged me to pursue my interests or supported me on the way. Costas Dimitracopoulos, Eleni Kalyvianaki, Costas Koutras, Joan and Yiannis Moschovakis at the graduate program "Logic, Algorithms, and Computation" in the University of Athens: I am deeply grateful to you. To my parents, thank you for helping me build up long-lasting stocks to make it through my journey. To my brother, to the rest of the family, and to my friends, thank you for being there, despite geographical distances. For helping me move to Saarland to embark on this trip, for moving with me to Potsdam to help me continue, and for sticking with me till the end (which is only another beginning), I most heartily thank Stefan. You gave me orientation and much needed perspective whenever I was getting lost. For all that, and for being the most effective generator of love I could imagine, thank you.

Contents

Abstract **v**

Acknowledgments **vii**

List of figures **xix**

List of tables **xxii**

1 Introduction **1**

1.1 Research problem 3

1.2 Challenges 4

1.2.1 Controlling the situated context 4

1.2.2 Adapting to the situated context 6

1.2.3 Monitoring the situated context 7

1.3 Related work 8

1.3.1 Controlling the situated context 9

1.3.2 Adapting to the situated context 9

1.3.3 Monitoring the situated context 10

1.4 Our methods 11

1.4.1 The GIVE experimental setting 11

1.4.2 Automated planning for natural language generation . . . 13

1.5 Thesis overview 15
1.5.1 Controlling the situated context with efficient planning . . 16
1.5.2 Adapting to the situated context to optimize effectiveness *during* planning 17
1.5.3 Monitoring the situated context to optimize effectiveness *after* planning 18
1.6 Main contributions 19
1.7 Organization of the thesis 20
1.7.1 Previously published material 20
1.7.2 Outline 23

2 Planning-based models of natural language generation 25
2.1 Introduction 25
2.2 Background 27
2.2.1 Natural language generation 27
2.2.2 Automated planning 28
2.3 Planning speech acts 30
2.4 Planning speech and physical acts 31
2.5 Planning words 33
2.6 Discussion 35
2.7 Conclusion 36

3 Automated planning for situated natural language generation 39
3.1 Introduction 39
3.2 Related work 42
3.3 Sentence generation as planning 43
3.3.1 TAG sentence generation 43
3.3.2 TAG generation as planning 44

3.3.3 Decoding the plan 45
3.4 Context manipulation 46
3.4.1 Situated CRISP 47
3.4.2 An example 48
3.5 Generating context-dependent adjectives 49
3.5.1 Context-dependence of adjectives in SCRISP 50
3.5.2 Adjective word order 51
3.6 Evaluation 52
3.6.1 The SCRISP system 53
3.6.2 Comparison with Baseline A 54
3.6.3 Comparison with Baseline B 55
3.7 Conclusion 56

4 Generation of effective referring expressions in situated context 57
4.1 Introduction 58
4.2 Referential effectiveness: Computational models and empirical insights 59
4.2.1 The effectiveness of humanlike references 60
4.2.2 The effectiveness of fixed preferences 62
4.3 Planning referring expressions 64
4.3.1 Converting language generation problems into planning problems 64
4.3.2 Solving planning problems to generate language 65
4.4 A statistical account of referential effectiveness 66
4.4.1 Situated reference in the GIVE-2 corpus 67
4.4.2 Measuring effectiveness 68
4.4.3 Modeling the situated context of referential scenes 69

4.4.4 A maximum entropy model of effectiveness in context . . 71
4.5 Optimizing effectiveness using metric planning 72
4.5.1 Assigning costs to attributes 73
4.5.2 Working around planner limitations 74
4.5.3 Generating referring expressions with mSCRISP 75
4.6 Automatic evaluation 76
4.6.1 Methods 76
4.6.2 Results 78
4.7 Human task performance evaluation 80
4.7.1 Methods 80
4.7.2 Results 82
4.8 Discussion 83
4.8.1 Improving the model 84
4.8.2 Implications for computational research 85
4.8.3 Implications for empirical research 86
4.9 Conclusion 87

5 Exploiting listener gaze to improve situated communication in dynamic virtual environments 89

5.1 Introduction 90
5.2 Related work 93
5.3 Methods 95
5.3.1 The 3D environments 95
5.3.2 Recording object inspections 96
5.3.3 The natural language generation systems 98
5.3.4 Participants and procedure 102
5.3.5 Data collection and analysis 103

5.4 Results . . . 106
5.4.1 Inspection of referents . . . 106
5.4.2 Visual processing of absolute and relative adjectives . . . 108
5.4.3 Referential understanding . . . 109
5.5 Discussion . . . 115
5.5.1 Key findings . . . 115
5.5.2 Future directions . . . 117
5.6 Conclusion . . . 119

6 Conclusion **121**
6.1 Summary . . . 121
6.2 Outlook . . . 124
6.2.1 Controlling the situated context . . . 124
6.2.2 Adapting to the situated context . . . 124
6.2.3 Monitoring the situated context . . . 125
6.2.4 Scaling up . . . 126

Bibliography **127**

A Grammar specification examples **149**
A.1 The lexicon of Fig. 3.4 . . . 149
A.2 The planning operators of Fig. 3.5 . . . 150
A.3 The lexicon of Fig. 3.6 . . . 151
A.4 The planning operators of Fig. 3.6 . . . 152
A.5 The lexicon of Fig. 4.4 . . . 153
A.6 The planning operators of Fig. 4.4 . . . 154

List of figures

1.1 Searching for an object of interest in a visually cluttered scene. . . 2

1.2 A scene from the GIVE setting (Koller et al., 2010b; Striegnitz et al., 2011), which provides a testbed for our approach. 12

1.3 A cleaning robot in the "vacuum world" (Russell and Norvig, 2009). The robot can transit between states by moving right (R) or left (L), and by sucking up dirt (S). With the goal of cleaning the two dirty rooms starting from the left one, the robot can use a planner to find a sequence of actions that may lead to a goal-satisfying state—e.g., <S, R, S>. 14

1.4 Conceptual model for the interactive generation of situated discourse, as proposed in this thesis. The model follows a continual planning approach, which interleaves planning, plan execution, and execution monitoring. 21

3.1 (a) An example grammar; (b) a derivation of "John pushes the red button" using (a). 41

3.2 CRISP planning operators for the elementary trees in Fig. 3.1(a). . 45

3.3 An example map for instruction giving. 46

3.4 An example SCRISP lexicon. 47

3.5 SCRISP planning operators for the lexicon in Fig. 3.4. 48

3.6 SCRISP operators for context-dependent and context-independent adjectives. 50

3.7 The IF's view of the scene in Fig. 3.3, as rendered by the GIVE client. 53

4.1 A simplified example of a CRISP lexicon and the derivation of the referring expression "the red button" describing b_1. 65

4.2 Simplified CRISP planning operators for the lexicon of Fig. 4.1, as in Garoufi and Koller (2010) (see Chapter 3). Predicates subst express that a syntax node is open for substitution, referent connect syntax nodes to the semantic individuals to which they refer, and canadjoin indicate the possibility of a tree adjoining the given syntax node. 66

4.3 Map of a virtual world from the GIVE-2 corpus. 67

4.4 Simplified mSCRISP planning operators for an attribute of type absolute. 76

4.5 Example of a reference situated in the context of a GIVE evaluation scene, as generated by mSCRISP. 81

5.1 A first-person view of a virtual 3D environment, as seen by users during the interactions. 96

5.2 A map of the environment in Fig. 5.1; note the user in the upper right room. 97

5.3 The course of a user's interaction with the eye-tracking-based system, following the instruction "Push the right button to the right of the green button" (see Table 5.1). The white circles around the rightmost button represent gaze information, as recorded by the system. 101

5.4 A faceLAB eye-tracking system remotely monitored participants' eye movements during the interactions. 101

5.5 A series of snapshots spanning a recorded referential scene with the eye-tracking-based generation system. 105

5.6 Average inspection time (% of time window) spent on target and non-anchor distractor buttons. Grey numbers represent the number of scene fragments falling into each time window. Differences between target and distractor inspection times are statistically significant at $^{***}p(\text{MCMC}) < .001$. 108

5.7 Average inspection time (% of time window) spent on distractor buttons, divided according to the type of adjectival pre-modifier used in the noun phrase (absolute or relative). Grey numbers represent the number of scene fragments falling into each time window. Differences in inspection times during processing of relative adjectives as compared to absolute adjectives are statistically significant at $^{**}p(\textsc{mcmc}) < .01$, $^{\circ}p(\textsc{mcmc}) < .1$. 110

5.8 Average number of 'H' keystrokes per interaction, by system. . . . 111

List of tables

3.1 Example system instructions generated in the same scene. REs for the target are typeset in boldface. 54

3.2 Evaluation results. Differences to SCRISP are significant at $*p < .05$, $**p < .005$ (Pearson's chi-square test for system success rates; unpaired two-sample t-test for the rest). 55

4.1 Attribute type annotations and their relative frequency (i.e., proportion of annotated references that contain an attribute of the given type) in the English edition of the GIVE-2 corpus. In this work, we focus on the six most frequent types. 68

4.2 Context variables of referential scenes. 70

4.3 Example of weights $v_j(s)$ in a scene s and corresponding cost assignments for each attribute type a_j. 75

4.4 Referring expressions produced by a human instruction giver, our model mSCRISP and the two baselines MaxEnt and EqualCosts in the bottom-left room of Fig. 4.3. 77

4.5 Average probabilities of high successfulness. Differences to mSCRISP are significant at $**p < .01$, $***p < .001$ (paired t-tests). 79

4.6 Average DICE coefficients across datasets. Differences to mSCRISP are significant at $*p < .05$, $***p < .001$ (paired t-tests). 79

4.7 Average resolution success and successfulness results in the shared task. Differences to mSCRISP are significant at $***p < .001$ (Pearson's χ^2 test for resolution success rates; unpaired two-sample t-tests for the rest). 82

4.8 Average error rate results as in Striegnitz et al. (2011), putting mSCRISP to comparison against EqualCosts and the six other systems participating in the shared task. Two systems do not share the same letter if the difference between them is significant ($p < .05$; ANOVA and post-hoc Tukey tests). 83

5.1 Example interactions between a participating user (U) and each of the three systems (S). All interactions were recorded during the systems' attempts to refer to the rightmost blue button shown in Fig. 5.1. The course of the interaction with the eye-tracking-based system (up to the onset of positive feedback) is illustrated in Fig. 5.3. 102

5.2 Mean referential success rates, feedback onset times and trial durations, broken down by presence and type of feedback. Differences to the eye-tracking system are significant at ***$p < .001$, **$p < .01$, *$p < .05$, $^{\circ}p < .1$. The number of referential scenes falling under each category is provided in the last column. 112

5.3 Mean values of additional task performance metrics. Differences to the eye-tracking-based system are significant at *$p < .05$. . . . 114

Chapter 1

Introduction

Imagine that you are inside a large shopping center as in Fig. 1.1. After a long Saturday morning of running errands, you are faced with the task of finding the wall-mounted silver-colored LED lamp you were looking for as fast as possible. The room is large and cluttered, and the lamp is distractingly placed among various other lamps and decorative objects, making it difficult to tell it apart. You still need to prepare today's lunch, and you are running the risk of spending too much time looking for the right item, or not finding it at all. As you are searching through the scene, for a brief moment you indulge in a vision of an intelligent computing system that could give you simple instructions to help you complete your task:

(1) a. "Walk three steps forward and then turn right."
(you walk and turn)

b. "OK. You're looking for the upper silver-colored lamp in front of you."
(you are being distracted by another silver-colored lamp in front of you, which uses halogen)

c. "No, not that one!"
(your eyes move upwards to the other silver-colored lamp)
"Yes, that one!"
(you find what you were looking for, successfully completing your task)

The vision of mobile conversational assistants has recently been shared by researchers in artificial intelligence, and with good reason. As our modern-built

Figure 1.1: Searching for an object of interest in a visually cluttered scene.

structures are becoming increasingly complex, carrying out basic activities such as identifying objects or points of interest in our surroundings can consume considerable time and cognitive resources. A computer system that interactively generates instructions in natural language to guide a user quickly and easily through their task—be it shopping at a mall, transiting through a busy metro station, or exploring the large collections of a museum in limited time—could positively impact our life. Remarkable strides in technology over the last few years have facilitated the advent of mobile and pervasive systems, yet several challenging questions remain to be addressed before we can experience such interactions in our daily lives. Beyond the technical difficulties in providing a system with essential contextual information about the user's environment, location, and visual attention, automatically converting these data into usefully verbalized guidance poses a pressing research problem. This is the problem we are addressing in this thesis.

In the present chapter, we examine the problem more closely and analyze the challenges involved, before briefly surveying related research. We then describe our methods, provide an overview of our approach, and summarize the main con-

tributions of the thesis. We conclude the chapter by outlining the organization of this manuscript.

1.1 Research problem

The automatic production of sentences in natural language by a computer system is known in computational linguistics as *natural language generation* (Reiter and Dale, 2000). Such a system typically receives as its input a *communicative goal* specifying the purpose of the sentences, a *knowledge source* (or *knowledge base*) with information about domain entities relevant to that purpose, a *discourse history* keeping track of what has been previously generated, and (sometimes) a *user model* with details about its user; its output is text. In this work, we are interested in text output that goes "beyond the sentence boundary" (Stede, 2011), spanning sequences of sentences that work together towards achieving one or more communicative goals. We call such multi-sentence text a *discourse*.

Discourses can have different purposes, such as to inform about the weather, to explain the functions of a technical device, or to persuade someone of a certain cause. Discourses as in (1), which are aimed at guiding someone through accomplishing a given task, are called *procedural* (Longacre, 1983). Typically, the communicative goals of a system engaged in such a discourse are sparked off by underlying *non-communicative* (or *non-linguistic*) *goals* related to the user's task (Bunt, 1994). The task we consider here—finding a specific location or object in the user's surroundings—mainly triggers two kinds of communicative goals: instructing the user on how to go from one location to another through *navigational* (or *route*) *instructions*, and identifying objects or locations to the user through *referring expressions*. Across a range of navigational settings, referring to landmarks along the way has been found to be more helpful to users than presenting them with purely prescriptive navigational instructions (e.g., Tom and Denis (2003); Dräger and Koller (2012); Mast et al. (2012)). Therefore, in this work we are particularly interested in *referential* communicative goals.

Providing in-situ assistance to the user requires a system to generate language *interactively*, adapting its discourse to the user's actions as they occur. Such interactions are also known as *problem-solving* (Polifroni et al., 1992), *task-oriented* (Traum and Hinkelman, 1992), *task-based* or *goal-directed* (Xu and Rudnicky, 2000), or *practical* (Allen et al., 2001). Interactive language generation is a key capability of *spoken dialog systems*, which are aimed at two-way spoken commu-

nication with a user. Such systems typically augment natural language generation with specialized modules for text-to-speech synthesis, speech recognition, natural language understanding, and overall dialog management; they may also integrate a component for prosody assignment into the language generation module (Walker et al., 2002). In this work, we will focus on the text generation process of interactive systems.

A discourse that unfolds within the context of a shared physical environment, such as the one depicted in Fig. 1.1, is called *situated*. Because situated language is produced "from a particular point of view within a physical context" (Byron, 2003), its form and content (as well as how it will be interpreted) is not influenced only by the linguistic context of the previous discourse, but can also be influenced by multiple aspects of the *non-linguistic context*—e.g., which events have previously taken place (*interaction-history context*), which objects and entities are visually available (*visual context*) or are being looked at (*gaze context*), and where they are located in space (*spatial context*). In a *dynamic* environment as in Fig. 1.1, where the user moves and turns, looks at objects, and acts upon them, this context changes continuously and rapidly.

Thus, in this thesis we will be addressing the problem of interactive generation of discourse, with the goal of identifying task-related entities to the user within a dynamic situated setting. Any approach to this problem must fulfill two requirements: First, the generated discourse must be *effective* in presenting information or instructions to help the user complete the task successfully and as effortlessly as possible. As a second requirement, because the system must provide appropriate guidance in real time, its language generation process must be computationally *efficient*.

1.2 Challenges

The dynamics of our communicative setting impose three fundamental challenges in meeting the above requirements and tackling our research problem. We turn to each of these challenges next.

1.2.1 Controlling the situated context

Let us begin by re-examining the first two system utterances in discourse (1):

(1a) SYSTEM: "Walk three steps forward and then turn right."
USER: *(walks and turns)*

(1b) SYSTEM: "OK. You're looking for the upper silver-colored lamp in front of you."

In this example, the system's communicative goal is to identify to the user one particular object in the scene of Fig. 1.1. The utterance in (1b), which contains the expression "the upper silver-colored lamp in front of you" referring to that object, serves this goal directly. However, the preceding utterance (1a), which is a navigational instruction, does not. Instead, its communicative goal is to instruct the user to move in a specific manner in the scene; its non-communicative goal is to change their position, orientation, and, ultimately, their focus of visual attention. In turn, the user's reaction to that utterance influences the non-linguistic context for the generation (and thus the interpretation) of the subsequent utterance in (1b). Since the user moves to a location from where they can see the object in front of them, the system is able to present a simpler referring expression than what might have been necessary if the user had not moved:

(2) "OK. You're looking for the upper silver-colored lamp on the right-hand side three steps down the aisle."

Such actions that are performed to "uncover information that is hidden or hard to compute mentally" (Kirsh and Maglio, 1994) are generally known as *epistemic actions*, and are distinguished from *pragmatic actions*, which are performed to bring one closer to the goal. The importance of this type of context-changing operations for the generation of referring expressions is observed by Dale and Reiter (1995), who argue that it may be useful to include *attention-directing information* to "bring the intended referent into the hearer's focus of attention" along with *discrimination information* to uniquely distinguish the referent. Though Dale and Reiter's foundational work does not address interactive generation, speakers in situated task-based human-human interaction have been observed to systematically distribute attention-directing expressions over multiple utterances (Stoia et al., 2006a; Schütte et al., 2010). In such settings, speakers frequently produce navigational instructions to make their intended referents visually salient to the hearer (as in (1a)), before producing the references themselves (as in (1b)). As Stoia et al. (2006a) argue, this strategy can lower the cognitive load of both speakers and hearers, thus overall improving the chances of successful communication.

To make the context more favorable for the satisfaction of the communicative goal, situated generation systems also need to master this interplay between language, action, and perception. This challenge requires systems to strategically choose language that can—through the actions it elicits—manipulate the non-linguistic context in ways that will facilitate their future discourse.

1.2.2 Adapting to the situated context

Regardless of whether a system has improved the context conditions before generating an utterance, the interpretation of that utterance will ultimately depend on the linguistic choices that the system makes within that context. To illustrate this, let us consider some alternative referring expressions that the system could choose to describe the referent in the context of Fig. 1.1:

"OK. You're looking for ..."

(3) a. "... the upper silver-colored lamp."

b. "... the upper left lamp in front of you."

c. "... the silver-colored LED lamp in front of you."

Let us assume that all these referring expressions are *distinguishing*, i.e., that they single out the target lamp from all *distractor* objects in the scene. Are they all equally effective in identifying the referent to the user successfully and speedily?

Alternative (3a) omits the logically *redundant* spatial relation "in front of you", which might be taken to be in line with one of the fundamental principles of cooperative communication, Grice's *maxim of quantity* "Do not make your contribution more informative than is required" (Grice, 1975). However, such seemingly unnecessary attributes in referring expressions have been shown to speed up identification time in certain (though not all) situations, especially when they allow the hearer to create a mental image of the referent or limit their search process (e.g., Arts et al. (2011); Paraboni and van Deemter (to appear)).

Alternative (3b) prefers the viewer-centered property "left" over the color, which seems to have the *discriminatory power* to rule out just as many distractors in the given scene. Yet there is empirical evidence that human speakers have a strong preference for color, even when it has less discriminatory power then

other attributes (Gatt et al., 2013). Moreover, Koolen et al. (2013) suggest that the use of color in scenes with high variation might in fact be beneficial for human hearers (while potentially distracting in scenes with low variation).

Finally, alternative (3c) includes the material "LED", which constitutes one of the lamp's basic properties; this property remains invariable even when the user changes perspective. Material has been found to be among the attributes most commonly used by human speakers when referring to everyday objects (Mitchell et al., 2013b). However, this attribute may not be perceptually available to the user in this particular scene, and therefore still not helpful (Paraboni et al., 2007).

Though understanding the exact influences of the situated context remains an active area of experimental research, it does become evident that effective linguistic choices are not fixed but highly dependent on those influences. A second challenge for situated generation systems is, therefore, to measure relevant aspects of the context and learn to choose their utterances in a way that optimizes them for that context.

1.2.3 Monitoring the situated context

Finally, even when a system has tailored its utterance to the situated context, this utterance may still fail in its communicative goal. This appears to be the case after the system's utterance in (1b):

(1b) SYSTEM: "OK. You're looking for the upper silver-colored lamp in front of you."
USER: *(the user is being distracted by another silver-colored lamp in front of them, which uses halogen)*

This risk of not being correctly understood is inherent in every utterance. On the one hand, an unforeseen event might suddenly occur and render the speaker's utterance inaudible or unrecognizable to the hearer (e.g., a child might start crying loudly). On the other hand, the hearer might not pay attention to what they heard (e.g., because their child required their full attention) or might simply not interpret it correctly (e.g., because they were tired). As Hirst et al. (1994) remark, such failures in constructing a correct interpretation can fall into two types: While in a *non-understanding* the hearer is aware of their trouble and may signal it, in a *misunderstanding* the hearer mistakenly believes that their interpretation is correct. Misunderstandings can thus be particularly threatening to communicative

success, as hearers in this case may not explicitly demonstrate that they are having trouble.

To address this problem, human speakers typically engage in a process of *grounding*, in which they work together with the addressee to mutually ensure that they have been understood "to a criterion sufficient for current purposes" (Clark and Wilkes-Gibbs, 1986; Clark and Schaefer, 1989). During this process, speakers *monitor* hearers for (positive or negative) evidence of understanding by attending to the hearers' verbal and facial acts—including their gaze—, their body, their workspace, as well as the overall shared scene (Clark and Krych, 2004); in other words, they monitor the situated context. Based on their observations, human speakers may then follow up with positive or negative *feedback* in ways similar to our envisioned system's utterances in (1c):

(1c) SYSTEM: "No, not that one!"
USER: *(the user's eyes move upwards to the other silver-colored lamp)*
SYSTEM: "Yes, that one!"
USER: *(the user finds what they were looking for, successfully completing their task)*

Because the risk of communicative failure may increase in human-computer interaction, especially when a system operates under limited or noisy information (Boye et al., 2012), being able to carry out this process seems necessary for computer systems, too (Traum, 1994).

Thus, another challenge for generation systems is to actively monitor the situated context for evidence of the user's understanding, and determine whether their communicative goal has been achieved. To support an effective interaction, this monitoring must be performed in real time, providing the system with an opportunity to respond to potential misunderstandings as early as possible.

1.3 Related work

Most research in natural language generation has so far focused on non-interactive, non-situated settings. In this section, we briefly examine how approaches to interactive or situated generation have addressed the three challenges we identified.

1.3.1 Controlling the situated context

As Dale and Reiter (1995) remark, it can be difficult for a system to generate appropriate attention-directing information in a computationally efficient way. A few situated reference generation systems can convey such information verbally (e.g., Appelt (1985a); Zender et al. (2009)) or e.g. with pointing gestures (van der Sluis and Krahmer, 2007), but these systems do not split the information into multiple installments so as to reduce the user's cognitive load. Certain approaches to interactive generation choose or present navigational instructions with some consideration of keeping the user's cognitive load low (e.g., Kray et al. (2003); Striegnitz and Majda (2009)), but they do not optimize the context conditions for their future utterances. Denis (2010) presents a situated system that incrementally modifies the linguistic context, using the *givenness hierarchy* (Gundel et al., 1993) to generate references based on the referent's cognitive status. However, this system does not strategically modify the non-linguistic context.

Stoia et al. (2006a) are the first to address the interleaving of navigational and discrimination information in order to control the situated context. The authors present a machine-learning approach that trains classifiers to signal when the context conditions seem appropriate for generating a referring expression. This method, however, cannot support a decision about which particular navigational instructions to generate, so as to make the subsequent referring expression simple. More recently, Dethlefs et al. (2011) present a reinforcement-learning approach to discourse generation that aims to optimize a combined measure of discourse length, communicative success, and linguistic consistency. Though human judges rated the resulting utterances favorably when receiving them in the context of static graphical scenes, we are not aware of an evaluation of this approach in an interactive task-based setting.

1.3.2 Adapting to the situated context

In recent years, computational approaches to referring expression generation have increasingly addressed the challenge of adapting to the situated context. Some approaches have applied machine learning to human-produced data with a representation of the context, with the purpose of learning how to vary their referring expressions according to features of the context (e.g., Jordan and Walker (2005); Stoia et al. (2006b); Spanger et al. (2009)). Stoia et al. (2006b), in particular, who train a decision tree learner to make decisions such as whether to include a modi-

fier or not, share with us a focus on generation that is situated in dynamic physical scenes.

This research, however, primarily attempts to replicate the referring expressions produced by humans, under the assumption that human-produced references are also, for the large part, effective (Viethen, 2011). Given that empirical findings are mixed as to the extent to which human-produced references are optimally helpful to hearers (e.g., Wardlow Lane and Ferreira (2008)) and that a shared-task evaluation found no correlations between humanlikeness and referential clarity (Gatt et al., 2009), this approach does not necessarily optimize task effectiveness. Additionally, as Stent (2011) argues, "humanlikeness may be unnecessary or maladaptive" in some interactive settings, for instance when the hearer is under increased cognitive load. Though some work has been concerned with optimizing effectiveness directly (e.g., Paraboni et al. (2007); Janarthanam and Lemon (2010)), we are not aware of any such work that has addressed linguistic choices of a broad scope in rich situated context.

1.3.3 Monitoring the situated context

Finally, interactive communicative systems have traditionally focused on the users' utterances as the primary source of evidence in this monitoring process (e.g., Walker et al. (2000); Bohus and Rudnicky (2002); Skantze and Schlangen (2009); DeVault et al. (2011)). While verbal cues can arguably provide rich information (e.g., Malisz et al. (2012)), relying on them has a number of drawbacks: First, such cues may be unavailable, as users may not be able or willing to engage in such dialog with the system. Second, the interpretation of these cues, once available, may be unreliable, since state-of-the-art speech recognition and natural language understanding components are notoriously error-prone. Third, waiting for the availability and semantic analysis of utterances can be time-consuming and enable a response only with delay, which may reach the user too late to prevent them from taking a wrong action.

As an alternative approach, Racca et al. (2011) monitor the user's non-verbal behavior, adapting Traum's (1999) computational model of grounding to dynamic situated context. However, the evidence they collect is limited to the user's moves and general field of view. Nakano et al. (2003) pay particular attention to the user's gaze—a ubiquitous and direct source—as complimentary evidence of understanding. Their approach, though, only monitors the basic direction of the user's gaze and has not been developed in dynamically changing physical context. Despite

recent advancements in remote eye-tracking technology and the increasing use of wearable eye-trackers in research and commercial applications (e.g., Foulsham and Kingstone (2012); Macdonald and Tatler (2013); Horning et al. (2013)), we are not aware of earlier approaches that have monitored fine-grained gaze cues in a complex setting to infer, on a per-utterance basis, a user's state of understanding.

1.4 Our methods

Having gained an overview of related work, we are now ready to describe our own methods for addressing the above challenges. In this section, we introduce our research setting and provide background to our language generation model.

1.4.1 The GIVE experimental setting

We use as a testbed the situated instruction-giving setting of the Challenge on Generating Instructions in Virtual Environments (GIVE; Koller et al. (2010b); Striegnitz et al. (2011)), which has served as a shared task for the evaluation of natural language generation systems three times since 2008. Because the software infrastructure of GIVE allows users to interact with systems conveniently over a computer network, the task has so far attracted the participation of 17 system-development teams and thousands of users.

Users and systems in GIVE interact in the context of a 3D virtual environment, such as the one shown in Fig. 1.2. The role of the system is to put together appropriate utterances in order to guide the user through finding a hidden treasure. Because the user has no previous knowledge of the environment, they rely on the system's instructions to complete the task. Successful task completion involves identifying a series of different-colored buttons that are attached to the walls in various locations and arranged in various ways. In the face of trouble, users can signal to the system their lack of understanding, by using the 'H' key on their keyboard to request help. Further, the task requires users to approach and press the buttons to which they resolved the system's references, thus providing conclusive evidence of the communicative success or failure of the generated utterances. This makes it possible to assess the effectiveness of different generation strategies directly.

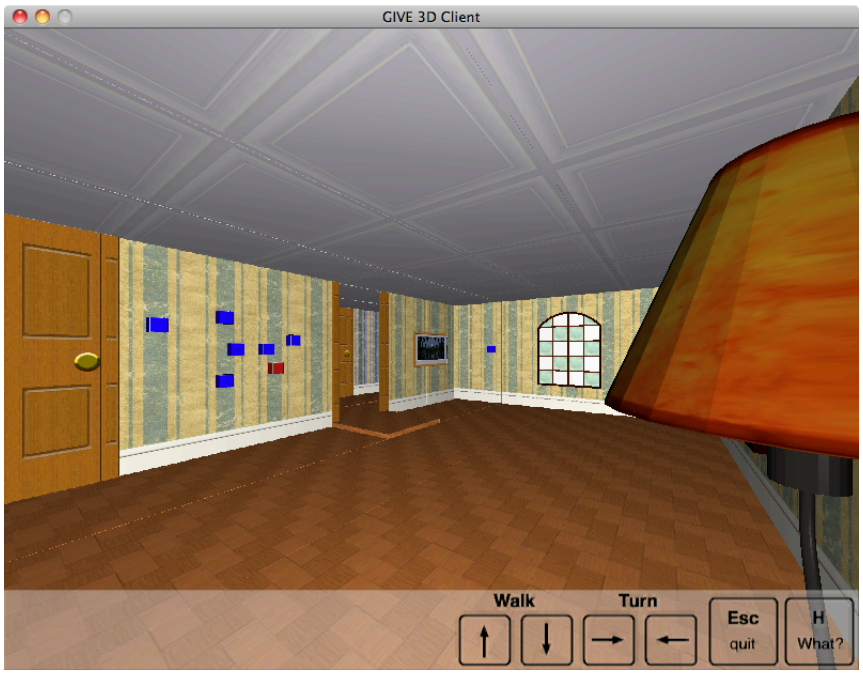


Figure 1.2: A scene from the GIVE setting (Koller et al., 2010b; Striegnitz et al., 2011), which provides a testbed for our approach.

During the interactions, systems are kept up-to-date with information about the relevant objects and their properties, the location of the user and their field of view, as well as the non-communicative goals related to the task. The setup thus simulates the information that a mobile computing system operating in the real world might maintain, assuming (idealized) access to a map and inventory of the environment, location-tracking and object-recognition technology, and knowledge of the user's goals. Though the difficulties in obtaining such knowledge in daily-life settings are not to be underestimated, this idealization allows communication within rich situated context, while making the influences of different aspects of the context measurable. For our research purposes, therefore, this setting retains many of the characteristics of real physical scenes as in Fig. 1.1, while still being sufficiently controlled.

Throughout this work, we will use the GIVE virtual environments as a platform both for empirical research and for system evaluation, as we shall see over the course of the thesis.

1.4.2 Automated planning for natural language generation

Our starting point for developing our approach is the observation that goal-directed language production, such as the one required for our task, constitutes a kind of *action*: If the circumstances are right, certain effects may ensue. As a result, specific goals of social, cognitive, or physical nature can sometimes be achieved by following appropriate courses of verbal action.

To illustrate this, let us assume that agent A has a *physical* goal that agent B moves to another location. Then agent A may attempt to satisfy that goal by speaking an utterance like the one spoken by the system in (1a):

(1a) "Walk three steps forward and then turn right."

Under the circumstances that agent B correctly understands the utterance, is rational and cooperative, and can afford performing the requested action, this *communicative act* will trigger a number of changes, called by Austin (1962) *perlocutionary effects*. First, agent B will believe that agent A wants them to move, and will come to adopt the intention to move—this changes their mental state. Second, to execute this intention, B will move to the specified location—this changes the state of the physical environment. Third, by moving to the requested location, B satisfies A's communicative goal, ultimately satisfying A's initial physical goal.

In situated task-based interaction, where participants interleave communication and physical action to complete the given task, communicative acts commonly have effects of both linguistic and physical nature, as in the above example. To generate discourse with the effects required for task success, a system then needs to be able to reason about both.

Fortunately, the problem of projecting the impact of actions and synthesizing them into an organized collection that will achieve the specified goal is what the over-four-decades-old field of *automated planning* (or *planning*) (Ghallab et al., 2004) specializes in. Given information about the *initial state* of an environment, the possible ways of making *transitions* from one state to another, and the *planning goal*, a planning algorithm can look for appropriate *actions* that may change the initial state in a way that satisfies the goal. Such an algorithm, known as a *planner*, can help, for instance, the robot of Fig. 1.3 to devise a *plan* of actions that, once executed, can achieve its cleaning goal. This fundamental form of reasoning, which is central to both artificial and human intelligence (see, e.g., Meyer et al. (2013)), can also be employed in natural language generation: As Cohen

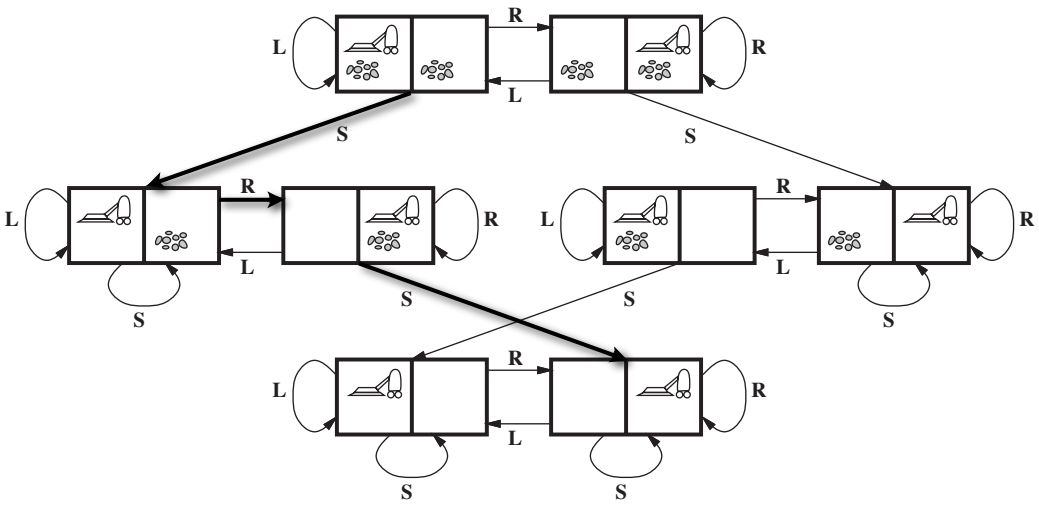


Figure 1.3: A cleaning robot in the "vacuum world" (Russell and Norvig, 2009). The robot can transit between states by moving right (R) or left (L), and by sucking up dirt (S). With the goal of cleaning the two dirty rooms starting from the left one, the robot can use a planner to find a sequence of actions that may lead to a goal-satisfying state—e.g., <S, R, S>.

and Perrault (1979) first showed, it is possible to generate language by formulating a communicative goal as the goal of a planning problem, and using automated planning techniques to compute plans of action that can solve that problem. Automated planning, the method that can help an autonomous robot work out how to navigate from one location to another in order to successfully complete its mission (e.g., Hofner and Schmidt (1995)), can also help a computer system reason about what to say in order to successfully communicate.

Given the compelling intuition that language and action can be treated (and planned for) uniformly, numerous approaches have used planning-based methods to generate language since Cohen and Perrault's (1979) work. A common characteristic of many such approaches, however, is their complexity, which may be prohibiting for real-time communication. Indeed, planning is, in the general case, a problem of high computational complexity. Yet, being driven by regular benchmark evaluations in the context of the International Planning Competitions (IPC; Coles et al. (2012)) starting in 1998, the state of the art has greatly advanced since the early approaches to language generation. Especially for the restricted problem of *classical planning* (Fikes and Nilsson, 1971), for which accurate domain-independent heuristics have been developed, efficient off-the-shelf planning tools have become widely available.

To benefit from such advancements, Koller and Stone (2007) re-implemented Stone et al.'s (2003) SPUD generator as a planning-based system, called CRISP, which is designed to be compatible with modern planners. The system formulates the natural language generation problem as a planning problem encoded in the Planning Domain Description Language (PDDL; McDermott (2000)), the language used by the IPC and understood by any participating planning system. In particular, it converts each entry of a *lexicalized grammar* into a *planning operator* that specifies how its use will contribute to the derivation of a sentence, uses a planner to organize these operators into a plan of action that achieves the communicative goal, and translates that plan back into a sentence. As a result, CRISP can generate full sentences by tapping into the capabilities of off-the-shelf classical planners such as the IPC-winning FF (Hoffmann and Nebel, 2001). In doing so, it utilizes Lexicalized Tree Adjoining Grammar (LTAG; Joshi and Schabes (1997)) as its grammar formalism, which has been shown to be particularly fitting for generation due to its tight coupling of syntax and semantics (e.g., McDonald and Pustejovsky (1985); Joshi (1987); Stone and Doran (1997); Stone and Webber (1998)). Unlike the traditionally used *pipeline* architecture, which separates the generation of each piece of text into distinct consecutive stages (Reiter and Dale, 2000), the CRISP system thus follows an *integrated* approach that allows decisions across stages to interact; this approach has repeatedly been argued to yield output of superior quality (e.g., Danlos (1984); Marciniak and Strube (2005); Dethlefs and Cuayáhuitl (2011); Lampouras and Androutsopoulos (2013)).

The CRISP system has not been previously used for the generation of discourse, is not interactive, and is not sensitive to aspects of the situated context. Additionally, its output does not attempt to optimize effectiveness (or any other qualitative metric) and has not been subjected to task-based evaluation. However, because of its unique combination of planning capabilities, efficiency, and expressiveness, we set out to build on this system for developing our approach.

1.5 Thesis overview

To extend CRISP to a system that can plan sequences of utterances that will work together towards achieving a given goal, we must be able to determine the perlocutionary effects of each single utterance. As we saw in the previous sections, however, the effects of utterances are typically uncertain. A system performing the communicative act in (1a), for instance, cannot be fully certain, in advance, that the user will be able to achieve the physical goal or even correctly understand

the communicative goal of the act—let alone understand it with ease. This implies that communicative settings are *non-deterministic* (since a communicative act may have different possible outcomes) and only *partially observable* (since the mental state of the user may remain unknown).

In such a setting we cannot model the uncertainty using classical planning, because this type of planning requires full observability and determinism. Some formalisms—in particular *conformant*, *contingent*, and *probabilistic* planning—reason about the uncertainty by computing conditional plans or state-action mappings for all possible contingencies "such that the agent can react adequately when faced with them" (Brenner and Nebel, 2009). However, the tools available for these forms of planning are generally not as efficient and robust as those for classical planning. On the other hand, approaches that translate problems featuring uncertainty into classical problems in richer domains, and solve them using efficient classical planners, have been found to outperform corresponding conformant (Palacios and Geffner, 2009) and contingent (Albore et al., 2009) planners. Most strikingly, an approach that determinizes probabilistic problems simply by ignoring the probabilities, generates plans using FF, and re-plans when things do not to go according to plan, has repeatedly outperformed probabilistic planners at the IPC in terms of success rate and planning time (Yoon et al., 2007).

To plan discourses efficiently, we therefore decide to follow a similar approach.

1.5.1 Controlling the situated context with efficient planning

We model communicative acts by assuming for each act an optimistic yet reasonable outcome: that it will be successful.[1] That is, we assume, for the purposes of planning, that all likely perlocutionary effects of utterances will indeed come true as intended. Under this assumption, we extend the CRISP planning operators with the non-linguistic effects that uttering a particular word is expected to have; in particular, those in the physical environment. This deterministic modeling makes it possible to use a classical planner based on FF (Koller and Hoffmann, 2010) in order to predict the situated context in which a later part of the utterance will be generated. To generate a plan that may achieve a given referential goal, the planner

[1] Recent research uses the term *assumption-based* or *commonsense* planning to describe a form of planning that shares with us the principle of making reasonable assumptions (Davis-Mendelow et al., 2013). As in our case, this approach is designed to benefit from efficient classical tools when resolving the uncertainty before planning is impossible.

then searches for contexts in which the referent is visible to the user; among the different discourse segments that change the initial context in this way, it chooses one for which the overall discourse length remains small. This allows the system to compute discourses that, if necessary, contain attention-directing information to make the referent visually salient, while at the same time distributing the cognitive load of interpreting the reference over multiple utterances. This context-manipulation capability addresses the challenge of Section 1.2.1. As an additional benefit, this approach models and correctly generates context-dependent spatial adjectives such as "left" and "right", whose interpretation is influenced not only by the linguistic but also by the non-linguistic context.

Though Koller and Hoffmann (2010) classify the CRISP domain "in the most difficult class of Hoffmann's (2005) planning domain taxonomy", we show with a human task-performance evaluation in GIVE that our approach performs well even under the constraints of real-time generation. We present this work in Chapter 3.

1.5.2 Adapting to the situated context to optimize effectiveness *during* planning

Though the above approach generates relatively simple and succinct referring expressions, these expressions are not necessarily optimal with respect to their effectiveness in the given context. To address this, we use a human-human interaction corpus in the GIVE setting (Gargett et al., 2010) that we annotated with the purpose of analyzing referential choices in different contexts. On this corpus, we train a maximum entropy model of the helpfulness of each attribute of a referring expression, given a set of variables that formalize the situated context. In contrast to traditional corpus-based approaches, the machine learner does not attempt to model the observed human speaker behavior invariably. Instead, it learns to distinguish speaker behavior according to its helpfulness to the hearer, and to selectively model behavior that was proven (by the hearer's reaction) to be helpful. By making this distinction anew in the context of each individual reference, the model tailors its output to the situated context, addressing the challenge of Section 1.2.2. We integrate the statistical model into CRISP under a *metric* planning formalism (Fox and Long, 2003) that associates each attribute type of a referent with an estimation of how preferable it is in the given context, and uses these preferences to optimize its referential effectiveness during planning.

In an intrinsic evaluation, we find the system's references to resemble those of

effective human speakers more closely than references of either a purely planning-based or a purely statistical baseline system. To assess its effectiveness with human hearers, we then further implemented our approach as the reference generation module of a GIVE system and participated in the third installment of the shared task (Striegnitz et al., 2011). While in this approach we make use of a more expressive (but still deterministic) planning formalism, we again achieve real-time performance using an off-the-shelf planner (Hoffmann, 2002). Though not all differences are statistically significant, the system's references were resolved correctly more often than those of seven other systems participating in the task. We present this work in Chapter 4.

1.5.3 Monitoring the situated context to optimize effectiveness *after* planning

Our optimistic approach to estimating the perlocutionary effects of utterances is able to find plans for non-trivial generation problems in real time. However, as the ensuing effects may sometimes differ from what has been predicted, we must be alert to failures in reaching the expected states as the system delivers its utterances, executing a given plan. In the fields of robotics and autonomous control, this problem is commonly addressed using *execution monitoring*, which Pettersson (2005) describes as "a continuous real-time task of determining the conditions of a physical system, by recording information, recognizing and indicating anomalies in the behavior". To achieve adequate execution monitoring in our setting, we consider the non-linguistic aspects of the context, and especially the user's gaze, which has been shown in eye-tracking studies to be rapidly directed towards understood referents (Tanenhaus et al., 1995; Allopenna et al., 1998). Though these studies have been limited to static visual settings, we hypothesize that monitoring a user's gaze can be a reliable means of optimizing the effectiveness of our communicative acts after they have been planned.

To this end, we design an execution-monitoring mechanism that addresses the challenge of Section 1.2.3 and tracks the user's gaze in the virtual environments of GIVE in order to assess whether they have correctly understood a generated referring expression. Starting immediately after the offset of the system's spoken utterance, this mechanism is *proactive* in that it is triggered before the user has had time to respond to that utterance with a physical action. The mechanism operates on top of the system's basic *reactive* execution monitoring, which monitors the user's physical actions (in our setting, button presses) after they have

occurred. We show that the user's gaze can indeed provide a reliable real-time index of their understanding, even in complex and dynamic environments, and on a per-utterance basis. Using this information to provide rapid feedback to the user improves overall task performance in comparison with two baseline systems that either do not engage in proactive execution monitoring, or do not exploit the user's gaze for their monitoring. We present this work in Chapter 5.

1.6 Main contributions

In short, we see the primary contributions of this work as follows:

1. We present an LTAG-based syntax-semantics interface for situated language, where non-linguistic information is naturally integrated with information of linguistic nature. We gain two main advantages from this modeling. First, we can generate context-dependent referring expressions by keeping track of both linguistically and non-linguistically introduced distractors during a unified generation process. Second, we develop the first, to our knowledge, full-fledged generation system that can deliberately manipulate the non-linguistic context of a communicative scene in order to make it more favorable for the generation of referring expressions.

2. We show how effective referential choices in situated context can be learned by assessing human-produced references in a task-based corpus for their effectiveness. Unlike traditional approaches, this model does not mimic human choices blindly—it only does so when there is indication that these choices are effective. We then show how the learned model of referential effectiveness can be integrated into a planning-based generation system. The resulting system, which combines symbolic and statistical reasoning, goes beyond the state of the art by tackling the problem of making non-trivial linguistic choices in rich situated context.

3. We demonstrate that the hearer's gaze provides a reliable index of online referential understanding even in complex and dynamic situated environments. At the same time, we present the first—to our knowledge—language generation system that monitors fine-grained gaze cues and uses them to generate feedback on a per-utterance basis. We show that exploiting hearer gaze in

this way enables the generation of appropriate feedback rapidly, which results in considerably improved task performance as revealed by a range of metrics.

Taken together, these results indicate that the challenges of language generation in situated context can be addressed using efficient methods from automated planning. By casting a non-deterministic planning problem as a deterministic one, and complementing deterministic planning with rapid and reliable execution monitoring, we have optimized the effectiveness of the generated discourse while retaining real-time performance. Though we have implemented and evaluated the systems of Chapters 3–5 individually rather than as a combined system, we sketch an integrated conceptual model of our approach in Fig. 1.4. This general approach is known as *continual planning*—an ongoing process in which planning, executing, and monitoring are interleaved (desJardins et al., 1999).

1.7 Organization of the thesis

We finish this chapter with a description of how the contents of the thesis have been organized.

1.7.1 Previously published material

Parts of the thesis have been published, accepted for publication, or are currently under review for publication, as follows:

- Konstantina Garoufi. Planning-based models of natural language generation. To appear in *Language and Linguistics Compass*. (Chapter 2)
- Konstantina Garoufi and Alexander Koller. Automated planning for situated natural language generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010. (Chapter 3)
- Konstantina Garoufi and Alexander Koller. Generation of effective referring expressions in situated context. To appear in *Language and Cognitive Processes*. (Chapter 4)

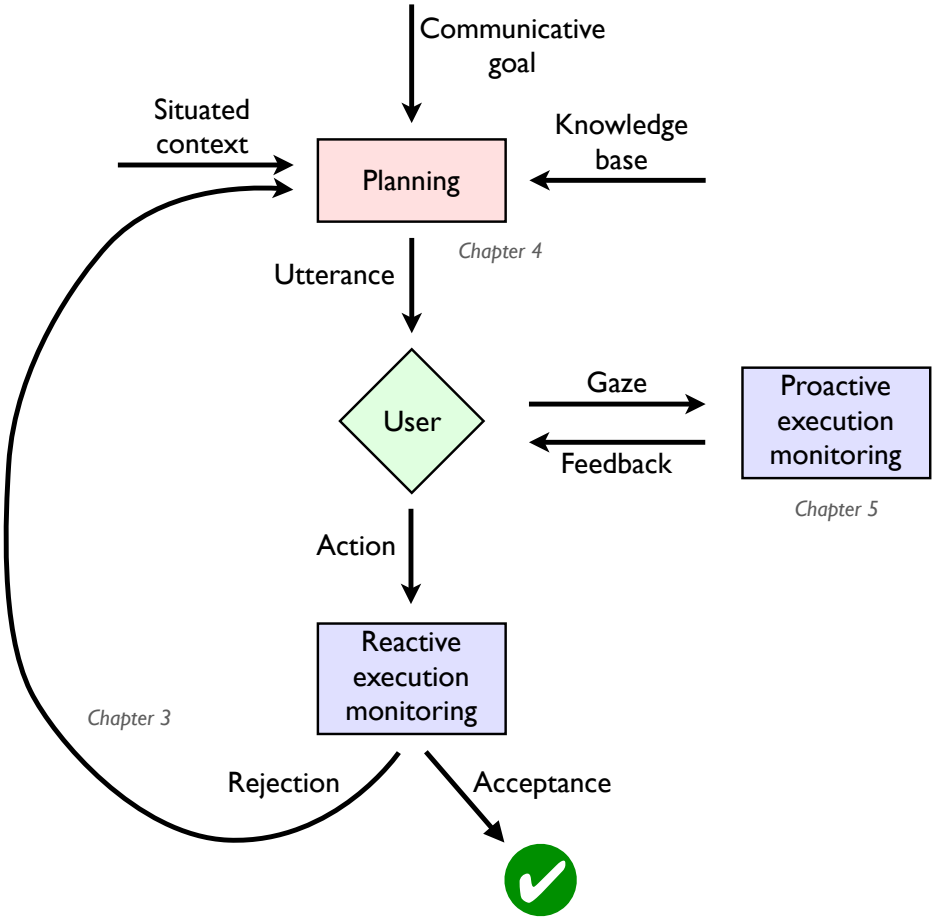


Figure 1.4: Conceptual model for the interactive generation of situated discourse, as proposed in this thesis. The model follows a continual planning approach, which interleaves planning, plan execution, and execution monitoring.

- Konstantina Garoufi, Maria Staudte, Alexander Koller, and Matthew Crocker. Exploiting listener gaze to improve situated communication in dynamic virtual environments. Under review for journal publication. (Chapter 5)

Preliminary work or partial findings have further been published as follows:

- Konstantina Garoufi and Alexander Koller. Controlling the spatio-visual context in situated natural language generation. In *Abstracts of the International Conference on Space in Language*, Pisa, Italy, 2009. (Chapter 3)

- Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. The GIVE-2 Corpus of Giving Instructions in Virtual Environments. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valletta, Malta, 2010. (Chapter 4)

- Alexander Koller, Andrew Gargett, and Konstantina Garoufi. A scalable model of planning perlocutionary acts. In P. Łupkowski and M. Purver, editors, *Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010*, pages 9–16. Polish Society for Cognitive Science, 2010. (Chapter 3)

- Konstantina Garoufi and Alexander Koller. Combining symbolic and corpus-based approaches for the generation of successful referring expressions. In *Proceedings of the 13th European Workshop on Natural Language Generation*, Nancy, France, 2011. (Chapter 4)

- Kristina Striegnitz, Alexandre Denis, Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Mariët Theune. Report on the Second Second NLG Challenge on Generating Instructions in Virtual Environments (GIVE-2.5). In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, Nancy, France, 2011. (Chapter 4)

- Konstantina Garoufi and Alexander Koller. The Potsdam NLG systems at the GIVE-2.5 Challenge. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, Nancy, France, 2011. (Chapter 4)

- Konstantina Garoufi. Position paper at the YRRSDS 2012. In *Proceedings of the 8th Annual Young Researchers' Roundtable on Spoken Dialogue Systems*, Seoul, South Korea, 2012. (Chapter 1)

- Alexander Koller, Konstantina Garoufi, Maria Staudte, and Matthew Crocker. Enhancing referential success by tracking hearer gaze. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Seoul, South Korea, 2012. (Chapter 5)
- Maria Staudte, Alexander Koller, Konstantina Garoufi, and Matthew Crocker. Using listener gaze to augment speech generation in a virtual 3D environment. In *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, Sapporo, Japan, 2012. (Chapter 5)
- Maria Staudte, Matthew Crocker, Alexander Koller, and Konstantina Garoufi. Grounding spoken instructions using listener gaze in dynamic virtual environments. In *Abstracts of the 5th Workshop on Embodied and Situated Language Processing*, Newcastle, UK, 2012. (Chapter 5)

1.7.2 Outline

In the rest of the thesis, we first survey the state of the art in planning-based models of language generation in Chapter 2. Beginning with an examination of the basic notions of natural language generation and planning, we show how these two fields of artificial intelligence have been drawn together. From the early work of Cohen and Perrault (1979) to ongoing research, we present a number of approaches that generate language by modeling communicative acts—at varying degrees of linguistic analysis—as actions in a planning problem. We consider classical and non-classical planning approaches, including recent probabilistic planning models, and discuss their strengths and weaknesses. We conclude the chapter by identifying a few possible ways in which this line of work could be advanced.

We then present our own planning-based approach to discourse generation that can control the situated context in Chapter 3. We extend this approach with a statistical model of effective referring expressions that adapt to the situated context in Chapter 4. Further, we complement our approach with eye-tracking-based mechanisms for monitoring the situated context in Chapter 5. We summarize our results and present ideas for future work in Chapter 6. Finally, we provide specification examples for our grammars and corresponding planning operators in Appendix A, supplementing Chapters 3 and 4.

Being self-contained pieces, Chapters 2–5 can be read out of order and fully independently from each other. Cross-references between them have been added where appropriate.

Chapter 2

Planning-based models of natural language generation[1]

Producing language is a kind of action—if the circumstances are right, certain effects may ensue, and specific physical, cognitive or social goals can sometimes be achieved by following appropriate courses of verbal action. The problem of synthesizing an organized collection of actions that leads to goal achievement can often be solved with automated planning methods. It is thus natural that such methods have found application to the automatic production of understandable text in natural language, i.e., to natural language generation. In this chapter, we survey a number of earlier and ongoing computational approaches to natural language that generate utterances by modeling speech acts or words as particular types of actions in a planning problem. After discussing strengths and weaknesses of the different models, we outline some possible directions for future work that could further advance this field.

2.1 Introduction

When people want to convey a belief, an emotion or an intention, change someone else's mental state, or even get them to physically do something, they can often accomplish that with language. For example, a museum attendant who is trying to help a visitor find the toilets might simply utter a string of words such as "Left

[1]This chapter is based on: Konstantina Garoufi. Planning-based models of natural language generation. To appear in *Language and Linguistics Compass.*

door at the end of the hallway in front of you" or "Second door on your left once you go straight down the hallway behind you". Depending on their current location in the room, one of these utterances can be correctly understood by the visitor and enable them to find the desired location, thus serving the museum attendant's communicative goal. That is, producing language is a kind of action: Under the right circumstances, a certain outcome may ensue; and the intended outcome could be achieved by taking the right course of action.

The problem of projecting the impact of actions and synthesizing them into an organized collection (e.g., a sequence) that will achieve the specified goal is what the over four decades old field of automated planning specializes in. Just as automated planning methods can help a robot work out how to navigate from one location to another, they can also help a computer system reason about what to say. By formulating the system's communicative goals as goals of planning problems, appropriate plans of action that solve these problems can be computed. Because effects of both linguistic (e.g., adding the description of a given location to the discourse history) and physical (e.g., enabling an addressee to arrive at that location) nature could result from an utterance, planning may sometimes involve a mixture of linguistic and non-linguistic elements.

This fundamental form of reasoning, which is central to both human and artificial intelligence, has been employed in computational models of natural language in different ways. Early work has shown how speakers' acts of producing certain types of utterances can be modeled in terms of planning actions, and how such acts can be automatically generated. Later work enriched this kind of planning in two principal ways: with physical acts of some sort, so as to capture the overall behavior of the speaker and other agents; and with a grammar, so as to construct, word by word, full sentences that obey the grammar rules. A further line of research has employed planning for natural language understanding instead of generation, seeking to infer a speaker's plans (and thus, their communicative goals) by observing their actions.

Plan of the chapter. In this chapter, we focus on planning-based models of generation. We will first provide some background in natural language generation and automated planning, and then survey the main lines of earlier and ongoing research that bring these two fields together. We will conclude by discussing strengths as well as weaknesses of different approaches and outlining possible directions for future work.

2.2 Background

To better understand how natural language generation and automated planning are drawn together, let us first examine some of the basic notions of each of these two subfields of artificial intelligence.

2.2.1 Natural language generation

Natural language generation deals with the task of developing computer systems that can put together meaningful natural-language utterances in order to meet specific communicative goals (Reiter and Dale, 2000). Such utterances may take the form of words, sentences, or discourses that span several sentences, according to the requirements of the task. As an example, a natural language generation system with the goal of communicating to the user the results of a given database query may attempt to satisfy this goal by summarizing, comparing, or describing (all or some of) the results (Rieser and Lemon, 2009). Such a system may also be part of a larger *spoken dialog system* that is designed to engage in two-way interactive communication with a user (Jurafsky and Martin, 2000). In this case, the system's communicative goals may be provided by a dedicated component called *dialog manager*, which, in turn, is interfaced with a *task manager* that has knowledge about the underlying (possibly non-communicative) goals of the interaction. Regardless of how their communicative goals are set, natural language generation systems need to choose meaningful utterances that express those goals.

The production of a meaningful utterance is what Austin (1962) names a *speech act*. In his seminal work "How to do things with words", Austin analyzes language production at three distinct levels. At the most basic level, putting words together in a legitimate way to form an utterance constitutes a *locutionary act*. Such an act—e.g., articulating the English words 'go', 'down', 'the' and 'hallway' in the right order and with sufficient clarity—allows a hearer to understand what meaning the speaker wants to convey. At the second level, the intended meaning of an utterance brings forth an *illocutionary act* of the speaker, e.g. an act of directing. Finally, an utterance will often not simply convey something to the hearer; it will also change their mental state or even future actions, constituting a *perlocutionary act*. For example, the utterance "go down the hallway" may cause the hearer to believe that those are the right directions to a particular destination and start following them, thus triggering off a number of effects.

Deciding what kind of speech act to perform is part of a more comprehensive process that a computer system follows to generate language. Reiter and Dale (2000) identify three main stages that natural language generation systems typically go through. Initially, *document planning* addresses the problem of determining what information to communicate (this is known as *content determination*), and how to arrange it into a discourse (*document structuring*). The results of document planning are next processed in the *microplanning* stage, which is responsible for making more fine-grained decisions such as how to aggregate sentences and which specific words to use. One important task at this stage is the generation of referring expressions, which is the task of creating descriptions—i.e., *referring expressions*—for referents in the domain. The final stage of *surface realization* translates the specifications made by microplanning into actual sentences that follow the rules of grammar.

Incidentally, the document planning and microplanning parts of natural language generation should not be confused with automated planning. In fact, automated planning is a distinct field of artificial intelligence, and is the topic we shall turn to next.

2.2.2 Automated planning

Automated planning—or, simply, *planning*—is the process of synthesizing an organized collection of actions whose execution will achieve a specified goal (Ghallab et al., 2004). Though planning specifications vary, in its most basic form a planning problem involves an initial state, a state transition system and a goal, as follows:

- The *initial state* of the world prescribes (in a formal logic-based language) which propositions are true at the moment of planning.
- The *state transition system* describes how the world can evolve as a result of *actions* (or *planning operators*), with which we can make transitions from one state to another. Actions consist of *preconditions*, determining which propositions must be true in a given state so that the given action can be executed, and *effects*, specifying how the truth conditions of those propositions will change after the execution.
- Finally, the *goal* is a specified state (or set of states) that we would like the world eventually to reach.

Solving the planning problem requires coming up with an appropriate specification of actions (in the simplest case, a sequential list) whose execution will lead us from the initial state to a goal-satisfying state. Such a solution is called a *plan*.

As a simple example, let's assume that a cleaning robot in a housekeeping domain is able to clean and move from one room to another by executing instances of the following actions:

clean($room1$):
PRECOND: $\mathsf{in}(room1)$
EFFECT: $\mathsf{cleaned}(room1)$

move($room1, room2$):
PRECOND: $\mathsf{in}(room1),\ \mathsf{accessible}(room1, room2)$
EFFECT: $\neg\mathsf{in}(room1),\ \mathsf{in}(room2)$

Let's further assume that such a robot is initially in a state which includes $\{\mathsf{in}(hallway), \mathsf{accessible}(hallway, bathroom)\}$, and it has a goal which includes $\{\mathsf{cleaned}(bathroom)\}$. The robot could achieve this goal if it could clean the bathroom. However, a precondition for cleaning the bathroom is that the robot is in there. Since that room happens to be accessible from its current room (the hallway), the robot could first move from the hallway to the bathroom and, once there, clean the bathroom. Supposing that no wheels get stuck or anything else unanticipated occurs, this two-step sequential plan <**move**($hallway, bathroom$), **clean**($bathroom$)> can be expected to achieve the robot's goal, thus solving the planning problem.

A computer system that solves planning problems is called a *planner*. Though planning technology has greatly advanced in the past years, planning is, in the general case, a problem of high computational complexity. To simplify the problem, *classical planning*, also known as STRIPS planning (Fikes and Nilsson, 1971), makes a number of restrictive assumptions—e.g., that actions are deterministic, that no *exogenous* events (i.e., events other than the encoded actions) can change the planning state, and that states are fully observable. Because such restrictions have allowed accurate domain-independent heuristics to be developed, numerous efficient off-the-shelf tools for classical planning have become available. Since 1998, planning tools have regularly participated in benchmark evaluations in the

context of the International Planning Competition[2] (e.g., Coles et al. (2012)), which has over the years evolved to encompass an uncertainty and a learning track next to classical planning. Both classical and more expressive planning formalisms have been explored in natural language generation, as we shall see in the next sections.

2.3 Planning speech acts

Since producing utterances is much like performing speech acts (Section 2.2.1), and automated planning can be used to figure out which sequences of acts will achieve a given goal (Section 2.2.2), the question arises whether planning methods can be applied to the automatic generation of natural language. Cohen and Perrault (1979) were among the first to explore this question systematically, arguing that the same processes used to construct plans of physical actions could also be used to construct communicative plans of speech acts. In their influential work, Cohen and Perrault showed how techniques from classical planning could be employed to the generation of speech acts for the satisfaction of a speaker's communicative goals. The authors focus on requesting and providing information in a cooperative task-based dialog setting, modeling, for example, a request as the following action:

request$(speaker, hearer, act)$:
PRECOND: $\mathsf{cando}(hearer, act)$,
$\mathsf{believe}\Big(speaker, \mathsf{want}\big(speaker, \mathsf{request}(speaker, hearer, act)\big)\Big)$
EFFECT: $\mathsf{believe}\Big(hearer, \mathsf{believe}\big(speaker, \mathsf{want}(speaker, act)\big)\Big)$

This definition states that if a hearer is able to do an act (cando), and a speaker believes themselves to be wanting to request that the hearer does that act, then the speaker may formulate this request. As an effect of the request, the hearer will then come to believe that the speaker believes themselves to be wanting the act.

Modeling speech acts as planning actions based on the (human or artificial) agents' mental states has been the topic of a considerable amount of later work (e.g., Hovy (1991); Maybury (1992); Moore and Paris (1993)). Cohen and Levesque (1990), in particular, refine the semantics of speech acts using a modal

[2] http://ipc.icaps-conference.org

temporal logic. This expressive formalism, which is based on philosophical foundations laid out by Bratman (1987), is intended to formalize the principles of rational action. Because this line of work involves deep reasoning about the beliefs, desires and intentions (BDI) of agents participating in dialog, it is known as the BDI-based framework of communicative planning.

One aspect of acting that the BDI approach typically does not address is uncertainty, which prevails in many (if not most) real-world interactions. For example, the effects of an action cannot always be fully foreseen; a moving act may fail to bring a robot to another location because its wheels might get stuck on an obstacle on the way, and an act of giving directions may fall short of changing an addressee's beliefs because a loud truck might happen to pass by and render the directions inaudible. To address such problems of non-determinism, different non-classical planning approaches have been explored. In a logic-based framework, Steedman and Petrick (2007) generate speech acts using Petrick and Bacchus's (2002) *contingent* planning system PKS, which aims at constructing conditional plans to cover all possible contingencies. Proposing a *probabilistic* planning approach, on the other hand, Rieser and Lemon (2009) formulate the speech act generation problem as a Markov Decision Process (MDP), a sequential decision problem for which stochastic reasoning about the best course of action (in this case, a state-to-action mapping called *policy*) can be performed.

Finally, it is worth mentioning that planning-based methods (in particular, plan inference and recognition) in the BDI paradigm have also been applied to solving problems of natural language interpretation in collaborative settings (e.g., Allen and Perrault (1980); Litman and Allen (1987); Grosz and Sidner (1990); Chu-Carroll and Carberry (1996)). More recently, Benotti and Blackburn (2011) use the classical planner FF (Hoffmann and Nebel, 2001) to construct plans from which conversational implicatures can be inferred.

2.4 Planning speech and physical acts

The synergy between communicative and physical act planning becomes most obvious in *situated* natural language generation, where language unfolds in the context of a physical environment that the communicating agents share. In such an environment, non-linguistic aspects of context like the agents' position in space and their visual fields can have a direct impact on the type and form of language they choose to produce; spoken language can in turn have its own impact on the agents'

future physical actions. Situated language planning thus becomes part of a more general architecture for an agent's behavior planning that integrates both speech and physical acts. Depending on their levels of embodiment, such agents must be able to switch seamlessly among the planning of speech acts (which may involve aspects—e.g., perlocutionary effects—of a non-linguistic nature), the planning of physical acts, the execution of these plans, and the observation of their environment.

Brenner and Kruijff-Korbayová (2008) approach this problem with a non-classical planning algorithm for *continual multiagent planning* (Brenner and Nebel, 2009). In this approach, agents do not only execute plans but they also monitor whether their current plans are still valid, and revise any parts that are no longer executable. Speech acts, e.g., requesting information, arise in this setting naturally as a result of collaborative problem solving behavior. Continual multiagent planning has also been employed in robotic systems that coordinate different forms of planning (e.g. perception, motion and communication) to engage in purposeful behavior in large-scale space (Hawes et al., 2009).

In recent work, Petrick and Foster (2012) apply Petrick and Bacchus's (2002) PKS planning system to a scenario of social interaction with a robot, in which social goals such as politeness must be satisfied in tandem with the task-based ones. As an example, the speech act of asking a customer to place their drink order at a bar is modeled as follows (in simplified notation):

ask-drink$(agent)$:
PRECOND: $\mathsf{inTrans}(agent)$, $\neg\mathsf{ordered}(agent)$, $\neg\mathsf{badASR}(agent)$, $\neg\mathsf{otherAttnReq}$
EFFECT: $\mathsf{ordered}(agent)$, $\mathsf{K_v}\big(\mathsf{request}(agent)\big)$

This states that, as a precondition for taking a drink order, the ordering agent is interacting with the robot (inTrans), has not already ordered, is being understood (¬badASR), and no other agents are seeking the robot's attention at the same time (¬otherAttnReq). As an effect, the planning state gets updated with the fact that the agent has ordered their drink, and with the specific kind of drink that they have requested ($\mathsf{K_v}(\mathsf{request})$). Since this information is unavailable at the moment of planning and can only become known at runtime, after the agent has placed their order, this action involves sensing in addition to the physical and verbal aspects. Along similar lines, Briggs and Scheutz (2013) extend Cohen and Perrault's (1979) approach to the generation and understanding of indirect

speech acts (in addition to direct speech acts and physical acts) in accordance with politeness norms and the agents' social roles.

Finally, planning-based methods have also been successful in generating narrative texts for multiagent story plots. Though the elements of good plot quality are arguably hard to lay down and formalize, some recent computational approaches to storytelling have employed planning technology to generate coherent plots consisting of physical and verbal acts. For example, Riedl and Young (2010) develop a special-purpose planner that plans the behavior of different story characters based on causal relationships between actions (for plot coherence) and intentionality of actions (for character believability). Also Brenner and Nebel's (2009) continual multiagent planner has been used to create story plots in which the characters' mental states can change dynamically as the story unfolds (Brenner, 2010).

2.5 Planning words

The planning-based approaches discussed so far may differ in the range of speech acts they consider and how they model them, but they mostly have one thing in common: They focus on the basic contents of the messages that need to be communicated and not on their precise form; in particular, they only employ planning to tackle initial stages of the generation process such as the content determination task (Section 2.2.1). Later stages, such as referring expression generation and surface realization, which spell out exactly how the specified contents are to be formulated in terms of full sentences, have so far been handed over to separate modules (e.g., template-based realizers). The works we discuss in this section refine the planning acts to include semantic and syntactic word-level details that enable full-fledged generation of sentences.

To this end, Appelt (1985a) develops a hierarchical planner that decomposes the referring expression generation problem in an abstraction hierarchy, gradually refining abstract goals and actions into sequences of primitive actions that the planning agent knows how to perform. One level of abstraction in this hierarchy is that of so-called surface speech acts, which, using a simple context-free grammar, can translate an illocutionary act into a linguistically realized utterance. This way, referring expression generation becomes an interleaved process that does not distinguish between deciding what to say and how to say it. Heeman and Hirst (1995) extend this approach to generate referring expressions in a collaborative

setting using more fine-grained surface speech acts. Though they do not handle the final mapping of those acts to realized utterances, the planning actions of Heeman and Hirst do encode detailed linguistic information that specifies how a valid definite description can be constructed from a head noun and the appropriate modifiers.

This idea of modeling the full sentence generation problem as a planning problem by including syntactic information in the planning actions has been taken one step further by Koller and Stone (2007). Koller and Stone utilize a lexicalized tree-adjoining grammar (Joshi and Schabes, 1997) to specify how each individual word contributes to the semantics and syntax of an utterance, and convert the grammar's lexical entries to planning actions. For instance, the word 'likes', as e.g. in the context of the sentence "Mary likes the white rabbit", corresponds to a planning action of the (simplified) form:

likes$(u, u_1, u_2, x, x_1, x_2)$:
PRECOND: $\mathsf{like}(x, x_1, x_2),\ \mathsf{subst}(\mathrm{S}, u),\ \mathsf{referent}(u, x)$
EFFECT: $\neg\mathsf{subst}(\mathrm{S}, u),\ \mathsf{subst}(\mathrm{NP}, u_1),\ \mathsf{subst}(\mathrm{NP}, u_2),\ \mathsf{referent}(u_1, x_1),$
$\mathsf{referent}(u_2, x_2),\ \forall y.\ \big(y \neq x_1 \rightarrow \mathsf{distractor}(u_1, y)\big),$
$\forall y.\ \big(y \neq x_2 \rightarrow \mathsf{distractor}(u_2, y)\big)$

This action contains a mixture of low-level semantic and syntactic information: To utter the verb 'likes' legitimately, its semantic content (like) must be satisfied, and the derivation of the utterance must be able to accommodate a sentence (S) about this liking event (expressed by the subst and referent predicates). After uttering 'likes', the syntax node u for that sentence has been filled, but also new noun phrase (NP) syntax nodes u_1 and u_2 for the grammatical agent and patient of the event, respectively, have appeared. These nodes correspond to domain entities for which referring expressions need to be generated. Because 'likes' alone does not specify which these entities are, all other known domain entities are recorded as *distractors*, i.e., entities that the actual referents need to be distinguished from (by uttering more words). Koller and Stone are able to solve the resulting planning problems efficiently using Hoffmann and Nebel's (2001) off-the-shelf classical planner FF. A series of recent works has extended this model to address situated language settings (Garoufi and Koller (2010); see Chapter 3), and to integrate corpus-based measures of humanlikeness (Bauer and Koller, 2010) or effectiveness (Garoufi and Koller (2011a); see Chapter 4) as metric planning constraints (Fox and Long, 2003).

Certain word-level decisions at varying degrees of linguistic analysis—though generally not as fine-grained as the above line of work—have recently also been addressed using different forms of probabilistic planning (Janarthanam and Lemon, 2009; Dethlefs and Cuayáhuitl, 2010; Smith and Lieberman, 2013).

2.6 Discussion

In the previous sections, we surveyed a growing body of research that has explored the use of automated-planning methods in natural language generation. While a wide range of planning forms have been explored, the main idea behind most of these approaches is the one of human communication as a goal-oriented process: People typically produce utterances because they want to achieve specific goals. As Hovy (1993) argues, analyzing such goals in terms of individual communicative subgoals that can be planned for may not always be possible, e.g., when the goal is to make a joke or to write a poem. Yet this does seem feasible for many other types of goals—e.g., to present information or to give instructions for the completion of certain tasks—, and such goals are what computers are most clearly useful for. This view is perhaps supported by the fact that shared tasks in the natural language generation community have increasingly been turning towards extrinsic evaluation scenarios, in which the systems' utterances are generated to serve tangible goals as required by the particular task at hand (e.g., Gatt et al. (2009); Striegnitz et al. (2011); Janarthanam et al. (2012)).

Which form of planning appears most promising? Different forms come with different trade-offs. Classical planning is well-studied and can offer efficient tools, but several researchers (e.g., Brenner and Nebel (2009)) argue that it can be too limited for real-world problems, in which the planning state can change unexpectedly due to exogenous events and may only be partially observable by the agent. Moreover, though domain-independent planning systems have been argued to meet the functional requirements of tasks such as document structuring (Young and Moore, 1994), some classical domain-independent planners were recently found to be too slow for certain language generation problems (Koller and Petrick, 2011). However, more flexible symbolic approaches, such as contingent planning, which constructs conditional plans in the face of uncertain initial states and action effects, are less efficient. Probabilistic planning approaches, on the other hand, are more complex, require substantial amounts of data for learning, and have only recently started being applied to sophisticated problems in natural language generation. In some cases, determinizing a planning problem by

ignoring the probabilities of outcomes, and simply re-planning when things are observed not to go according to plan, has been shown to outperform probabilistic approaches (Yoon et al., 2007); this approach may be particularly promising in interactive communicative settings.[3]

Different perspectives are taken also on the depth and granularity of the planning actions. The deep reasoning about agents' mental states that the BDI model performs has been regarded as "clearly necessary for building conversational agents that can interact" (Jurafsky, 2004), but has also been criticized for lack of scalability (Koller et al., 2010a). At the same time, the high number of choices involved in the generation process has led to the common use of a pipeline architecture that separates microplanning from document planning and surface realization (Section 2.2.1). However, successful approaches to tackling these tasks jointly, through a uniform process, have made compelling arguments for the interdependent nature of the stages in the pipeline (e.g., Stone et al. (2003)). The deterministic planning approaches of Section 2.5 are built on such architectures, but no known work has employed expressive deterministic formalisms such as hierarchical task-network planning (e.g., Nau et al. (2003)) to handle the complexity of generating full sentences while still leveraging the dependencies among language-producing acts.

2.7 Conclusion

To conclude, natural language generation and automated planning are connected through a strong intuition of treating language and action uniformly, and a four-decade long tradition of interdisciplinary research. Despite these connections, the problem of how planning can most adequately model the language generation process is not yet well understood. The complex nature of communicative planning, which involves numerous sources of uncertainty about the quality of a given plan, makes the problem challenging; still, a few promising directions have recently emerged. The use of statistical techniques—be it in the form of probabilistic planning as introduced in Section 2.3, preference learning for metric planning as in Section 2.5 (see Chapter 4), or some other form—can help express and reason about uncertainties in a natural way. Yet symbolic methods can be more advantageous when obvious logical requirements exist that must satisfied (e.g., distinguishing a referent from all distractors in referring expression generation).

[3] In fact, this is the general approach we propose in this thesis (see overview in Chapter 1).

As the strengths and weaknesses of different planning approaches become better understood, it is possible that combinations of symbolic and statistical methods that can complement each other will further advance the field.

Chapter 3

Automated planning for situated natural language generation[1]

We present a natural language generation approach which models, exploits, and manipulates the non-linguistic context in situated communication, using techniques from AI planning. We show how to generate instructions which deliberately guide the hearer to a location that is convenient for the generation of simple referring expressions, and how to generate referring expressions with context-dependent adjectives. We implement and evaluate our approach in the framework of the Challenge on Generating Instructions in Virtual Environments, finding that it performs well even under the constraints of real-time generation.

3.1 Introduction

The problem of situated natural language generation (NLG)—i.e., of generating natural language in the context of a physical (real or virtual) environment—has received increasing attention in the past few years. On the one hand, this is because it is the foundation of various emerging applications, including human-robot interaction and mobile navigation systems, and is the focus of a current evaluation effort, the Challenges on Generating Instructions in Virtual Environments (GIVE; Koller et al. (2010b)). On the other hand, situated generation comes with inter-

[1]This chapter is based on: Konstantina Garoufi and Alexander Koller. Automated planning for situated natural language generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010.

esting theoretical challenges: Compared to the traditional generation of text given the linguistic context, the interpretation of expressions in situated communication is sensitive to the non-linguistic context, and this context can change as easily as the user can move around in the environment.

One interesting aspect of situated communication from an NLG perspective is that this non-linguistic context can be manipulated by the speaker. Consider the following segment of discourse between an instruction giver (IG) and an instruction follower (IF), which is adapted from the SCARE corpus (Stoia et al., 2008):

(4) IG: "Walk forward and then turn right."
IF: *(walks and turns)*
IG: "OK. Now hit the button in the middle."

In this example, the IG plans to refer to an object (here, a button); and in order to do so, gives a navigation instruction to guide the IF to a convenient location at which she can then use a simple referring expression (RE). That is, there is an interaction between navigation instructions (intended to manipulate the non-linguistic context in a certain way) and referring expressions (which exploit the non-linguistic context). Although such subdialogs are common in SCARE, we are not aware of any previous research that can generate them in a computationally feasible manner.

This chapter presents an approach to generation which is able to model the effect of an utterance on the non-linguistic context, and to intentionally generate utterances such as the above as part of a process of referring to objects. Our approach builds upon the CRISP generation system (Koller and Stone, 2007), which translates generation problems into planning problems and solves these with an AI planner. We extend the CRISP planning operators with the perlocutionary effects that uttering a particular word has on the physical environment if it is understood correctly; more specifically, on the position and orientation of the hearer. This allows the planner to predict the non-linguistic context in which a later part of the utterance will be interpreted, and therefore to search for contexts that allow the use of simple REs. As a result, the work of referring to an object gets distributed over multiple utterances of low cognitive load rather than a single complex noun phrase.

A second contribution of this work is the generation of REs involving context-dependent adjectives: A button can be described as "the left blue button" even if

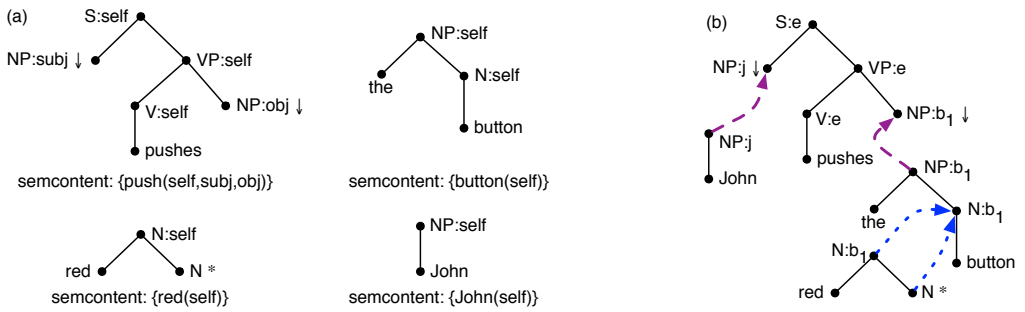


Figure 3.1: (a) An example grammar; (b) a derivation of "John pushes the red button" using (a).

there is a red button to its left. We model adjectives whose interpretation depends on the nominal phrases they modify, as well as on the non-linguistic context, by keeping track of the distractors that remain after uttering a series of modifiers. Thus, unlike most other RE generation approaches, we are not restricted to building an RE by simply intersecting lexically specified sets representing the extensions of different attributes, but can correctly generate expressions whose meaning depends on the context in a number of ways. In this way we are able to refer to objects earlier and more flexibly.

We implement and evaluate our approach in the context of a GIVE NLG system, by using the GIVE-1 software infrastructure and a GIVE-1 evaluation world. This shows that our system generates an instruction-giving discourse as in (4) in about a second. It outperforms a mostly non-situated baseline significantly, and compares well against a second baseline based on one of the top-performing systems of the GIVE-1 Challenge. Next to the practical usefulness this evaluation establishes, we argue that our approach to jointly modeling the grammatical and physical effects of a communicative action can also inform new models of the pragmatics of speech acts.

Plan of the chapter. We discuss related work in Section 3.2, and review the CRISP system, on which our work is based, in Section 3.3. We then show in Section 3.4 how we extend CRISP to generate navigation-and-reference discourses as in (4), and add context-dependent adjectives in Section 3.5. We evaluate our system in Section 3.6; Section 3.7 concludes and points to future work.

3.2 Related work

The research reported here can be seen in the wider context of approaches to generating referring expressions. Since the foundational work of Dale and Reiter (1995), there has been a considerable amount of literature on this topic. Our work departs from the mainstream in two ways. First, it exploits the situated communicative setting to deliberately modify the context in which an RE is generated. Second, unlike most other RE generation systems, we allow the contribution of a modifier to an RE to depend both on the context and on the rest of the RE.

We are aware of only one earlier[2] study on generation of REs with focus on interleaving navigation and referring (Stoia et al., 2006a). In this machine learning approach, Stoia et al. train classifiers that signal when the context conditions (e.g. visibility of target and distractors) are appropriate for the generation of an RE. This method can be then used as part of a content selection component of an NLG system. Such a component, however, can only inform a system on whether to choose navigation over RE generation at a given point of the discourse, and is not able to help it decide what kind of navigational instructions to generate so that subsequent REs become simple.

To our knowledge, the only previous research on generating REs with context-dependent modifiers is van Deemter's (2006) algorithm for generating vague adjectives. Unlike van Deemter, we integrate the RE generation process tightly with the syntactic realization, which allows us to generate REs with more than one context-dependent modifier and model the effect of their linear order on the meaning of the phrase. In modeling the context, we focus on the non-linguistic context and the influence of each of the RE's words; this is in contrast to previous research on context-sensitive generation of REs, which mainly focused on the discourse context (Krahmer and Theune, 2002). Our interpretation of context-dependent modifiers picks up ideas by Kamp and Partee (1995) and implements them in a practical system, while our method of ordering modifiers is linguistically informed by the class-based paradigm (e.g., Mitchell (2009)).

On the other hand, our work also stands in a tradition of NLG research that is based on AI planning. Early approaches (Perrault and Allen, 1980; Appelt, 1985b) provided compelling intuitions for this connection, but were not computationally viable. The research we report here can be seen as combining Appelt's idea of using planning for sentence-level NLG with a computationally benign variant of Perrault and Allen's approach of modeling the intended perlocutionary effects of

[2] See Section 1.3.1 for a discussion that includes later published work.

a speech act as the effects of a planning operator. Our work is linked to a growing body of very recent work that applies modern planning research to various problems in NLG (Steedman and Petrick, 2007; Brenner and Kruijff-Korbayová, 2008; Benotti, 2009). It is directly based on Koller and Stone's (2007) reimplementation of the SPUD generator (Stone et al., 2003) with planning. As far as we know, ours is the first system in the SPUD tradition that explicitly models the context change effects of an utterance.

While nothing in our work directly hinges on this, we implemented our approach in the context of an NLG system for the GIVE Challenge (Koller et al., 2010b), that is, as an instruction giving system for virtual worlds. This makes our system comparable with other approaches to instruction giving implemented in the GIVE framework.

3.3 Sentence generation as planning

Our work is based on the CRISP system (Koller and Stone, 2007), which encodes sentence generation with tree-adjoining grammars (TAG; Joshi and Schabes (1997)) as an AI planning problem and solves that using efficient planners. It then decodes the resulting plan into a TAG derivation, from which it can read off a sentence. In this section, we briefly recall how this works. For space reasons, we will present primarily examples instead of definitions.

3.3.1 TAG sentence generation

The CRISP generation problem (like that of SPUD (Stone et al., 2003)) assumes a lexicon of entries consisting of a TAG elementary tree annotated with semantic and pragmatic information. An example is shown in Fig. 3.1(a). In addition to the elementary tree, each lexicon entry specifies its *semantic content* and possibly a *semantic requirement*, which can express certain presuppositions triggered by this entry. The nodes in the tree may be labeled with argument names such as *semantic roles*, which specify the participants in the relation expressed by the lexicon entry; in the example, every entry uses the semantic role self representing the event or individual itself, and the entry for "pushes" furthermore uses subj and obj for the subject and object argument, respectively. We combine here for simplicity the entries for "the" and "button" into "the button".

For generation, we assume as input a knowledge base and a communicative goal in addition to the grammar. The goal is to compute a derivation that expresses the communicative goal in a sentence that is grammatically correct and complete; whose meaning is justified by the knowledge base; and in which all REs can be resolved to unique individuals in the world by the hearer. Let's say, for example, that we have a knowledge base $\{\mathsf{push}(e, \mathsf{j}, b_1), \mathsf{John}(\mathsf{j}), \mathsf{button}(b_1), \mathsf{button}(b_2), \mathsf{red}(b_1)\}$. Then we can combine instances of the trees for "John", "pushes", and "the button" into a grammatically complete derivation. However, because both b_1 and b_2 satisfy the semantic content of "the button", we must adjoin "red" into the derivation to make the RE refer uniquely to b_1. The complete derivation is shown in Fig. 3.1(b); we can read off the output sentence "John pushes the red button" from the leaves of the derived tree we build in this way.

3.3.2 TAG generation as planning

In the CRISP system, Koller and Stone (2007) show how this generation problem can be solved by converting it into a planning problem (Ghallab et al., 2004). The basic idea is to encode the partial derivation in the planning state, and to encode the action of adding each elementary tree in the planning operators. The encoding of our example as a planning problem is shown in Fig. 3.2.

In the example, we start with an initial state which contains the entire knowledge base, plus atoms $\mathsf{subst}(\mathrm{S}, \mathsf{root})$ and $\mathsf{referent}(\mathsf{root}, e)$ expressing that we want to generate a sentence about the event e. We can then apply the (instantiated) action **pushes**$(\mathsf{root}, \mathsf{n}_1, \mathsf{n}_2, \mathsf{n}_3, e, \mathsf{j}, b_1)$, which models the act of substituting the elementary tree for "pushes" into the substitution node root: It can only be applied because root is an unfilled substitution node (precondition $\mathsf{subst}(\mathrm{S}, \mathsf{root})$), and its effect is to remove $\mathsf{subst}(\mathrm{S}, \mathsf{root})$ from the planning state while adding two new atoms $\mathsf{subst}(\mathrm{NP}, \mathsf{n}_1)$ and $\mathsf{subst}(\mathrm{NP}, \mathsf{n}_2)$ for the substitution nodes of the "pushes" tree. The planning state maintains information about which individual each node refers to in the referent atoms. The current and next atoms are needed to select unused names for newly introduced syntax nodes.[3] Finally, the action introduces a number of distractor atoms including $\mathsf{distractor}(\mathsf{n}_2, e)$ and $\mathsf{distractor}(\mathsf{n}_2, b_2)$, expressing that the RE at n_2 can still be misunderstood by the hearer as e or b_2.

In this new state, all subst and distractor atoms for n_1 can be eliminated with the action **John**$(\mathsf{n}_1, \mathsf{j})$. We can also apply the action **the-button**(n_2, b_1) to

[3] This is a different solution to the name-selection problem than in Koller and Stone (2007). It is simpler and improves computational efficiency.

pushes$(u, u_1, u_2, u_n, x, x_1, x_2)$:
PRECOND: $\mathsf{subst}(\mathrm{S}, u), \mathsf{referent}(u, x), \mathsf{push}(x, x_1, x_2), \mathsf{current}(u_1), \mathsf{next}(u_1, u_2), \mathsf{next}(u_2, u_n)$
EFFECT: $\neg\mathsf{subst}(\mathrm{S}, u), \mathsf{subst}(\mathrm{NP}, u_1), \mathsf{subst}(\mathrm{NP}, u_2), \mathsf{referent}(u_1, x_1), \mathsf{referent}(u_2, x_2), \forall y.\mathsf{distractor}(u_1, y), \forall y.\mathsf{distractor}(u_2, y)$

John(u, x):
PRECOND: $\mathsf{subst}(\mathrm{NP}, u), \mathsf{referent}(u, x), \mathsf{John}(x)$
EFFECT: $\neg\mathsf{subst}(\mathrm{NP}, u), \forall y.\big(\neg\mathsf{John}(y) \rightarrow \neg\mathsf{distractor}(u, y)\big)$

the-button(u, x):
PRECOND: $\mathsf{subst}(\mathrm{NP}, u), \mathsf{referent}(u, x), \mathsf{button}(x)$
EFFECT: $\neg\mathsf{subst}(\mathrm{NP}, u), \mathsf{canadjoin}(\mathrm{N}, u), \forall y.\big(\neg\mathsf{button}(y) \rightarrow \neg\mathsf{distractor}(u, y)\big)$

red(u, x):
PRECOND: $\mathsf{canadjoin}(\mathrm{N}, u), \mathsf{referent}(u, x), \mathsf{red}(x)$
EFFECT: $\forall y.\big(\neg\mathsf{red}(y) \rightarrow \neg\mathsf{distractor}(u, y)\big)$

Figure 3.2: CRISP planning operators for the elementary trees in Fig. 3.1(a).

eliminate $\mathsf{subst}(\mathrm{NP}, \mathsf{n}_2)$ and $\mathsf{distractor}(\mathsf{n}_2, e)$, since e is not a button. However $\mathsf{distractor}(\mathsf{n}_2, b_2)$ remains. Now because the action **the-button** also introduced the atom $\mathsf{canadjoin}(\mathrm{N}, \mathsf{n}_2)$, we can remove the final distractor atom by applying **red**(n_2, b_1). This brings us into a goal state, and we are done. Goal states in CRISP planning problems are characterized by axioms such as $\forall A \forall u.\neg\mathsf{subst}(A, u)$ (encoding grammatical completeness) and $\forall u \forall x.\neg\mathsf{distractor}(u, x)$ (requiring unique reference).

3.3.3 Decoding the plan

An AI planner such as FF (Hoffmann and Nebel, 2001) can compute a plan for a planning problem that consists of the planning operators in Fig. 3.2 and a specification of the initial state and the goal. We can then decode this plan into the TAG derivation shown in Fig. 3.1(b). The basic idea of this decoding step is that an action with a precondition $\mathsf{subst}(A, u)$ fills the substitution node u, while an action with a precondition $\mathsf{canadjoin}(A, u)$ adjoins into a node of category A in the elementary tree that was substituted into u. CRISP allows multiple trees to adjoin into the same node. In this case, the decoder executes the adjunctions in the order in which they occur in the plan.

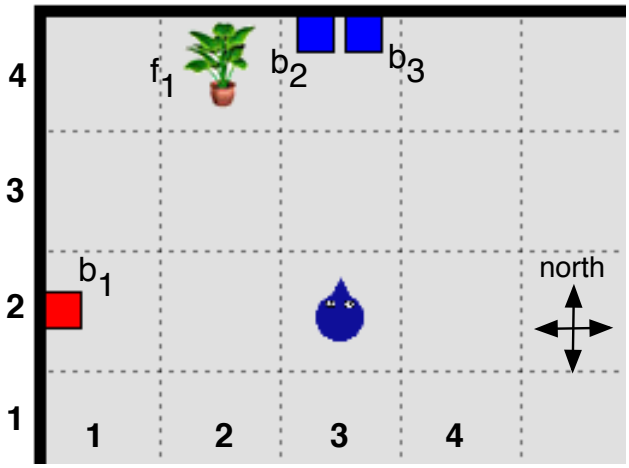


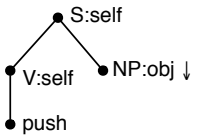
Figure 3.3: An example map for instruction giving.

3.4 Context manipulation

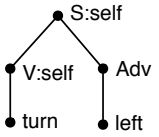
We are now ready to describe our NLG approach, SCRISP ("Situated CRISP"), which extends CRISP to take the non-linguistic context of the generated utterance into account, and deliberately manipulate it to simplify RE generation.

As a simplified version of our introductory instruction giving example (4), consider the map in Fig. 3.3. The instruction follower (IF), who is located on the map at position $\mathsf{pos}_{3,2}$ facing north, sees the scene from the first-person perspective as in Fig. 3.7. Now an instruction giver (IG) could instruct the IF to press the button b_1 in this scene by saying "push the button on the wall to your left". Interpreting this instruction is difficult for the IF because it requires her to either memorize the RE until she has turned to see the button, or to perform a mental rotation task to visualize b_1 internally. Alternatively, the IG can first instruct the IF to "turn left"; once the IF has done this, the IG can then simply say "now push the button in front of you". This lowers the cognitive load on the IF, and presumably improves the rate of correctly interpreted REs.

SCRISP is capable of deliberately generating such context-changing navigation instructions. The key idea of our approach is to extend the CRISP planning operators with preconditions and effects that describe the (simulated) physical environment: A "turn left" action, for example, modifies the IF's orientation in space and changes the set of visible objects; a "push" operator can then pick up this changed set and restrict the distractors of the forthcoming RE it introduces



semreq: visible(p, o, obj)
pragcon: player–pos(p),
player–ori(o)
impeff: push(obj)



pragcon: player–ori(o_1),
next–ori–left(o_1, o_2)
prageff: ¬player–ori(o_1),
player–ori(o_2)
impeff: turnleft

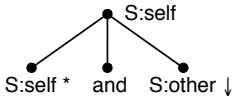


Figure 3.4: An example SCRISP lexicon.

(i.e. "the button") to only objects that are visible in the changed context. We also extend CRISP to generate imperative rather than declarative sentences.

3.4.1 Situated CRISP

We define a lexicon for SCRISP to be a CRISP lexicon in which every lexicon entry may also describe *non-linguistic conditions*, *non-linguistic effects* and *imperative effects*. Each of these is a set of atoms over constants, semantic roles, and possibly some free variables. Non-linguistic conditions specify what must be true in the world so a particular instance of a lexicon entry can be uttered felicitously; non-linguistic effects specify what changes uttering the word brings about in the world; and imperative effects contribute to the IF's "to-do list" (Portner, 2007) by adding the properties they denote.

A small lexicon for our example is shown in Fig. 3.4. This lexicon specifies that saying "push X" puts pushing X on the IF's to-do list, and carries the presupposition that X must be visible from the location where "push X" is uttered; this reflects our simplifying assumption that the IG can only refer to objects that are currently visible. Similarly, "turn left" puts turning left on the IF's agenda. In addition, the lexicon entry for "turn left" specifies that, under the assumption that the IF understands and follows the instruction, they will turn 90 degrees to the left after hearing it. The planning operators are written in a way that assumes that the intended (perlocutionary) effects of an utterance actually come true. This assumption is crucial in connecting the non-linguistic effects of one SCRISP ac-

push$(u, u_1, u_n, x, x_1, p, o)$:
PRECOND: $\mathsf{subst}(\mathsf{S}, u), \mathsf{referent}(u, x), \mathsf{player\text{–}pos}(p),$
$\mathsf{player\text{–}ori}(o), \mathsf{visible}(p, o, x_1), \ldots$
EFFECT: $\neg\mathsf{subst}(\mathsf{S}, u), \mathsf{subst}(\mathsf{NP}, u_1), \mathsf{referent}(u_1, x_1),$
$\forall y.\big(y \neq x_1 \wedge \mathsf{visible}(p, o, y) \rightarrow \mathsf{distractor}(u_1, y)\big),$
$\mathsf{to\text{–}do}(\mathsf{push}(x_1)), \mathsf{canadjoin}(\mathsf{S}, u), \ldots$

turnleft(u, x, o_1, o_2):
PRECOND: $\mathsf{subst}(\mathsf{S}, u), \mathsf{referent}(u, x), \mathsf{player\text{–}ori}(o_1),$
$\mathsf{next\text{–}ori\text{–}left}(o_1, o_2), \ldots$
EFFECT: $\neg\mathsf{subst}(\mathsf{S}, u), \neg\mathsf{player\text{–}ori}(o_1), \mathsf{player\text{–}ori}(o_2),$
$\mathsf{to\text{–}do}(\mathsf{turnleft}), \ldots$

and(u, u_1, u_n, e_1, e_2):
PRECOND: $\mathsf{canadjoin}(\mathsf{S}, u), \mathsf{referent}(u, e_1), \ldots$
EFFECT: $\mathsf{subst}(\mathsf{S}, u_1), \mathsf{referent}(u_1, e_2), \ldots$

Figure 3.5: SCRISP planning operators for the lexicon in Fig. 3.4.

tion to the non-linguistic preconditions of another, and generalizes to a scalable model of planning perlocutionary acts. We discuss this in more detail in Koller et al. (2010a).

We then translate a SCRISP generation problem into a planning problem. In addition to what CRISP does, we translate all non-linguistic conditions into preconditions and all non-linguistic effects into effects of the planning operator, adding any free variables to the operator's parameters. An imperative effect P is translated into an effect $\mathsf{to\text{–}do}(P)$. The operators for the example lexicon of Fig. 3.4 are shown in Fig. 3.5. Finally, we add information about the situated environment to the initial state, and specify the planning goal by adding $\mathsf{to\text{–}do}(P)$ atoms for each atom P that is to be placed on the IF's agenda.

3.4.2 An example

Now let's look at how this generates the appropriate instructions for our example scene of Fig. 3.3. We encode the state of the world as depicted in the map in an initial state which contains, among others, the atoms $\mathsf{player\text{–}pos}(\mathsf{pos}_{3,2})$,

player–ori(north), next–ori–left(north, west), visible($\mathsf{pos}_{3,2}$, west, b_1), etc.[4] We want the IF to press b_1, so we add to–do(push(b_1)) to the goal.

We can start by applying the action **turnleft**(root, e, north, west) to the initial state. Next to the ordinary grammatical effects from CRISP, this action makes player–ori(west) true. The new state does not contain any subst atoms, but we can continue the sentence by adjoining "and", i.e. by applying the action **and**(root, n_1, n_2, e, e_1). This produces a new atom subst(S, e_1), which satisfies one precondition of **push**(n_1, n_2, n_3, e_1, b_1, $\mathsf{pos}_{3,2}$, west). Because **turnleft** changed the player orientation, the visible precondition of **push** is now satisfied too (unlike in the initial state, in which b_1 was not visible). Applying the action **push** now introduces the need to substitute a noun phrase for the object, which we can eliminate with an application of **the-button**(n_2, b_1) as in Section 3.3.2.

Since there are no other visible buttons from $\mathsf{pos}_{3,2}$ facing west, there are no remaining distractor atoms at this point, and a goal state has been reached. Together, this four-step plan decodes into the sentence "turn left and push the button". The final state contains the atoms to–do(push(b_1)) and to–do(turnleft), indicating that an IF that understands and accepts this instruction also accepts these two commitments into their to-do list.

3.5 Generating context-dependent adjectives

Now consider if we wanted to instruct the IF to press b_2 in Fig. 3.3 instead of b_1, say with the instruction "push the left button". This is still challenging, because (like most other approaches to RE generation) CRISP interprets adjectives by simply intersecting all their extensions. In the case of "left", the most reasonable way to do this would be to interpret it as "leftmost among all visible objects"; but this is f_1 in the example, and so there is no distinguishing RE for b_2.

In truth, spatial adjectives like "left" and "upper" depend on the context in two different ways. On the one hand, they are interpreted with respect to the current spatio-visual context, in that what is on the left depends on the current position and orientation of the hearer. On the other hand, they also depend on the meaning of the phrase they modify: "the left button" is not necessarily both a button and further to the left than all other objects, it is only the leftmost object among the

[4] In a more complex situation, it may be infeasible to exhaustively model visibility in this way. This could be fixed by connecting the planner to an external spatial reasoner (Dornhege et al., 2009).

left(u, x):
PRECOND: $\forall y.\neg\big(\mathsf{distractor}(u, y) \wedge \mathsf{left\text{–}of}(y, x)\big)$,
$\mathsf{canadjoin}(\mathsf{N}, u), \mathsf{referent}(u, x)$
EFFECT: $\forall y.\big(\mathsf{left\text{–}of}(x, y) \rightarrow \neg\mathsf{distractor}(u, y)\big)$,
$\mathsf{premod\text{–}index}(u, 2), \ldots$

red(u, x):
PRECOND: $\mathsf{red}(x), \mathsf{canadjoin}(\mathsf{N}, u), \mathsf{referent}(u, x)$,
$\neg\mathsf{premod\text{–}index}(u, 2)$
EFFECT: $\forall y.\big(\neg\mathsf{red}(y) \rightarrow \neg\mathsf{distractor}(u, y)\big)$,
$\mathsf{premod\text{–}index}(u, 1), \ldots$

Figure 3.6: SCRISP operators for context-dependent and context-independent adjectives.

buttons. We will now show how to extend SCRISP so it can generate REs that use such context-dependent adjectives.

3.5.1 Context-dependence of adjectives in SCRISP

As a planning-based approach to NLG, SCRISP is not limited to simply intersecting sets of potential referents that only depend on the attributes that contribute to an RE: Distractors are removed by applying operators which may have context-sensitive conditions depending on the referent and the distractors that are still left.

Our encoding of context-dependent adjectives as planning operators is shown in Fig. 3.6. We only show the operators here for lack of space; they can of course be computed automatically from lexicon entries. In addition to the ordinary CRISP preconditions, the **left** operator has a precondition requiring that no current distractor for the RE u is to the left of x, capturing a presupposition of the adjective. Its effect is that everything that is to the right of x is no longer a distractor for u. Notice that we allow that there may still be distractors after **left** has been applied (above or below x); we only require unique reference in the goal state. (Ignore the premod–index part of the effect for now; we will get to that in a moment.)

Let's say that we are computing a plan for referring to b_2 in the example map of Fig. 3.3, starting with **push**$(\mathsf{root}, \mathsf{n}_1, \mathsf{n}_2, e, b_2, \mathsf{pos}_{3,1}, \mathsf{north})$ and **the-button**(n_1, b_2). The state after these two actions is not a goal state, because it still contains the atom $\mathsf{distractor}(\mathsf{n}_1, b_3)$ (the plant f_1 was removed as a distractor by the action

the-button). Now assume that we have modeled the spatial relations between all objects in the initial state in left–of and above atoms; in particular, we have left–of(b_2, b_3). Then the action instance **left**(n_1, b_2) is applicable in this state, as there is no other object that is still a distractor in this state and that is to the left of b_2. Applying **left** removes distractor(n_1, b_3) from the state. Thus we have reached a goal state; the complete plan decodes to the sentence "push the left button".

This system is sensitive to the order in which operators for context-dependent adjectives are applied. To generate the RE "the upper left button", for instance, we first apply the **left** action and then the **upper** action, and therefore **upper** only needs to remove distractors in the leftmost position. On the other hand, the RE "the left upper button" corresponds to first applying **upper** and then **left**. These action sequences succeed in removing all distractors for different context states, which is consistent with the difference in meaning between the two REs.

Furthermore, notice that the adjective operators themselves do not interact directly with the encoding of the context in atoms like visible and player–pos, just like the noun operators in Section 3.4 didn't. The REs to which the adjectives and nouns contribute are introduced by verb operators; it is these verb operators that inspect the current context and initialize the distractor set for the new RE appropriately. This makes the correctness of the generated sentence independent of the order in which noun and adjective operators occur in the plan. We only need to ensure that the verbs are ordered correctly, and the workload of modeling interactions with the non-linguistic context is limited to a single place in the encoding.

3.5.2 Adjective word order

One final challenge that arises in our system is to generate the adjectives in the correct order, which on top of semantically valid must be linguistically acceptable. In particular, it is known that some types of adjectives are limited with respect to the word order in which they can occur in a noun phrase. For instance, "large foreign financial firms" sounds perfectly acceptable, but "? foreign large financial firms" sounds odd (Shaw and Hatzivassiloglou, 1999). In our setting, some adjective orders are forbidden because only one order produces a correct and distinguishing description of the target referent (cf. "upper left" vs. "left upper" example above). However, there are also other constraints at work: "? the red left button" is rather odd even when it is a semantically correct description, whereas "the left red button" is fine.

To ensure that SCRISP chooses to generate these adjectives correctly, we follow a class-based approach to the premodifier ordering problem (Mitchell, 2009). In our lexicon we assign adjectives denoting spatial relations ("left") to one class and adjectives denoting color ("red") to another; then we require that spatial adjectives must always precede color adjectives. We enforce this by keeping track of the current *premodifier index* of the RE in atoms of the form premod–index. Any newly generated RE node starts off with a premodifier index of zero; adjoining an adjective of a certain class then raises this number to the index for that class. As the operators in Fig. 3.6 illustrate, color adjectives such as "red" have index one and can only be used while the index is not higher; once an adjective from a higher class (such as "left", of a class with index two) is used, the premod–index precondition of the "red" operator will fail. For this reason, we can generate a plan for "the left red button", but not for "? the red left button", as desired.

3.6 Evaluation

To establish the quality of the generated instructions, we implemented SCRISP as part of a generation system in the GIVE-1 framework, and evaluated it against two baselines. GIVE-1 was the First Challenge on Generating Instructions in Virtual Environments, which was completed in 2009 (Koller et al., 2010b). In this challenge, systems must generate real-time instructions that help users perform a task in a treasure-hunt virtual environment such as the one shown in Fig. 3.7.

We conducted our evaluation in World 2 from GIVE-1, which was deliberately designed to be challenging for RE generation. The world consists of one room filled with several objects and buttons, most of which cannot be distinguished by simple descriptions. Moreover, some of those may activate an alarm and cause the player to lose the game. The player's moves and turns are discrete and the NLG system has complete and accurate real-time information about the state of the world. Instructions that each of the three systems under comparison generated in an example scene of the evaluation world are presented in Table 3.1.

The evaluation took place online via the Amazon Mechanical Turk, where we collected 25 games for each system. We focus on four measures of evaluation: success rates for solving the task and resolving the generated REs, average task completion time (in seconds) for successful games, and average distance (in steps) between the IF and the referent at the time when the RE was generated. As in the challenge, the task is considered as solved if the player has correctly

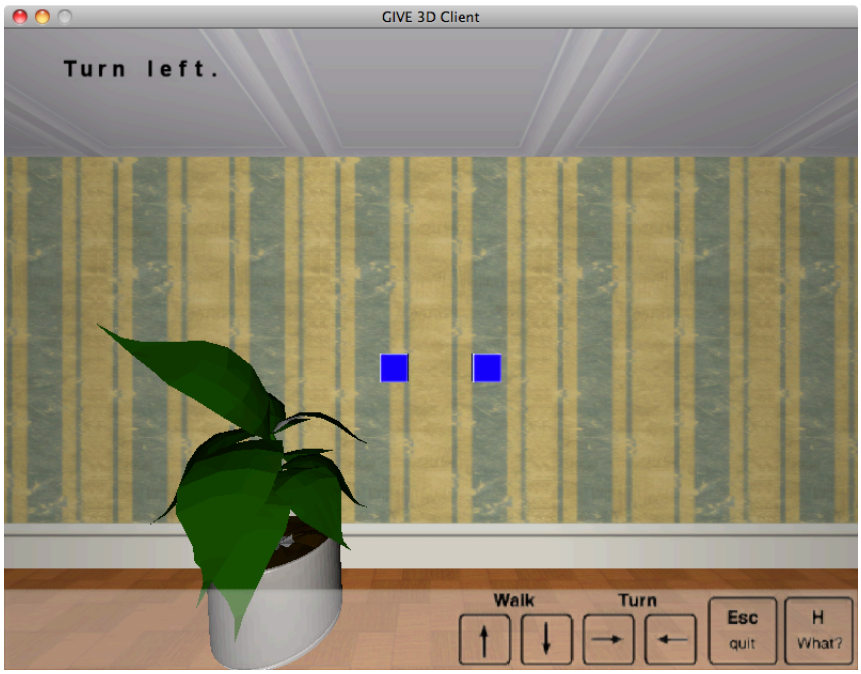


Figure 3.7: The IF's view of the scene in Fig. 3.3, as rendered by the GIVE client.

been led through manipulating all target objects required to discover and collect the treasure; in World 2, the minimum number of such targets is eight. An RE is successfully resolved if it results in the manipulation of the referent, whereas manipulation of an alarm-triggering distractor ends the game unsuccessfully.

3.6.1 The SCRISP system

Our system receives as input a plan for what the IF should do to solve the task, and successively takes object-manipulating actions as the communicative goals for SCRISP. Then, for each of the communicative goals, it generates instructions using SCRISP, segments them into navigation and action parts, and presents these to the user as separate instructions sequentially (see Table 3.1).

For each instruction, SCRISP thus draws from a knowledge base of about 1500 facts and a grammar of about 30 lexicon entries. We use the FF planner (Hoffmann and Nebel, 2001; Koller and Hoffmann, 2010) to solve the planning problems. The maximum planning time for any instruction is 1.03 seconds on a 3.06 GHz Intel Core 2 Duo CPU. So although our planning-based system tackles

System	Instructions
SCRISP	*1. Turn right and move one step.* *2. Push* ***the right red button****.*
Baseline A	*1. Press* ***the right red button on the wall to your right****.*
Baseline B	*1. Turn right.* *2. Walk forward 3 steps.* *3. Turn right.* *4. Walk forward 1 step.* *5. Turn left.* *6. Good! Now press* ***the left button****.*

Table 3.1: Example system instructions generated in the same scene. REs for the target are typeset in boldface.

a very difficult search problem, FF is very good at solving it—fast enough to generate instructions in real time.

3.6.2 Comparison with Baseline A

Baseline A is a very basic system designed to simulate the performance of a classical RE generation module which does not attempt to manipulate the visual context. We hand-coded a correct distinguishing RE for each target button in the world; the only way in which Baseline A reacts to changes of the context is to describe on which wall the button is with respect to the user's current orientation (e.g. "Press the right red button *on the wall to your right*").

As Table 3.2 shows, our system guided 69% of users to complete the task successfully, compared to only 16% for Baseline A (difference is statistically significant at $p < .005$; Pearson's chi-square test). This is primarily because only 49% of the REs generated by Baseline A were successful. This comparison illustrates the importance of REs that minimize the cognitive load on the IF to avoid misunderstandings.

	Success		RE	
	rate	**time**	**success**	**distance**
SCRISP	69%	306	71%	2.49
Baseline A	16%**	230	49%**	1.97*
Baseline B	84%	288	81%*	2.00*

Table 3.2: Evaluation results. Differences to SCRISP are significant at *$p < .05$, **$p < .005$ (Pearson's chi-square test for system success rates; unpaired two-sample t-test for the rest).

3.6.3 Comparison with Baseline B

Baseline B is a corrected and improved version of the "Austin" system (Chen and Karpov, 2009), one of the best-performing systems of the GIVE-1 Challenge. Baseline B, like the original "Austin" system, issues navigation instructions by precomputing the shortest path from the IF's current location to the target, and generates REs using the description logic based algorithm of Areces et al. (2008). Unlike the original system, which inflexibly navigates the user all the way to the target, Baseline B starts off with navigation, and opportunistically instructs the IF to push a button once it has become visible and can be described by a distinguishing RE. We fixed bugs in the original implementation of the RE generation module, so that Baseline B generates only unambiguous REs. The module nonetheless naively treats all adjectives as intersective and is not sensitive to the context of their comparison set. Specifically, a button cannot be referred to as "the *right* red button" if it is not the rightmost of all visible objects—which explains the long chain of navigational instructions the system produced in Table 3.1.

We did not find any significant differences in the success rates or task completion times between this system and SCRISP, but the former achieved a higher RE success rate (see Table 3.2). However, a closer analysis shows that SCRISP was able to generate REs from significantly further away. This means that SCRISP's RE generator solves a harder problem, as it typically has to deal with more visible distractors. Furthermore, because of the increased distance, the system's execution monitoring strategies (e.g. for detection and repair of misunderstandings) become increasingly important, and this was not a focus of this work. In summary, then, we take the results to mean that SCRISP performs quite capably in comparison to a top-ranked GIVE-1 system.

3.7 Conclusion

In this chapter, we have shown how situated instructions can be generated using AI planning. We exploited the planner's ability to model the perlocutionary effects of communicative actions for efficient generation. We showed how this made it possible to generate instructions that manipulate the non-linguistic context in convenient ways, and to generate correct REs with context-dependent adjectives.

We believe that this illustrates the power of a planning-based approach to NLG to flexibly model very different phenomena. An interesting topic for future work, for instance, is to expand our notion of context by taking visual and discourse salience into account when generating REs. In addition, we plan to experiment with assigning costs to planning operators in a metric planning problem (Hoffmann, 2002) in order to model the cognitive cost of an RE (Krahmer et al. (2003)) and compute minimal-cost instruction sequences.[5]

On a more theoretical level, the SCRISP actions model the physical effects of a correctly understood and grounded instruction directly as effects of the planning operator. This is computationally much less complex than classical speech act planning (Perrault and Allen, 1980), in which the intended physical effect comes at the end of a long chain of inferences. But our approach is also very optimistic in estimating the perlocutionary effects of an instruction, and must be complemented by an appropriate model of execution monitoring.[6] What this means for a novel scalable approach to the pragmatics of speech acts (Koller et al., 2010a) is, we believe, an interesting avenue for future research.

[5] We address this in Chapter 4, which focuses on the optimization of REs in situated context.

[6] We propose such a model in Chapter 5.

Chapter 4

Generation of effective referring expressions in situated context[1]

In task-oriented communication, references often need to be effective in their distinctive function, that is, help the hearer identify the referent correctly and as effortlessly as possible. However, it can be challenging for computational or empirical studies to capture referential effectiveness. Empirical findings indicate that human-produced references are not always optimally effective, and that their effectiveness may depend on different aspects of the situational context that can evolve dynamically over the course of an interaction. On this basis, we propose a computational model of effective reference generation which distinguishes speaker behavior according to its helpfulness to the hearer in a certain situation, and explicitly aims at modeling highly helpful speaker behavior rather than speaker behavior invariably. Our model, which extends the planning-based paradigm of sentence generation with a statistical account of effectiveness, can adapt to the situational context by making this distinction newly for each new reference. We find that the generated references resemble those of effective human speakers more closely than references of baseline models, and that they are resolved correctly more often than those of other models participating in a shared-task evaluation with human hearers. Finally, we argue that the model could serve as a methodological framework for computational and empirical research on referential effectiveness.

[1]This chapter is based on: Konstantina Garoufi and Alexander Koller. Generation of effective referring expressions in situated context. To appear in *Language and Cognitive Processes.*

4.1 Introduction

In task-oriented communication, speakers frequently produce distinctive referring expressions. The primary purpose of such expressions is to help the hearer uniquely identify the referent; a distinctive referring expression is *effective* if the hearer resolves it to the intended referent correctly and, ideally, effortlessly. As a consequence, computational models of reference in task-oriented settings typically aim at generating referring expressions that are as effective as possible. However, it can be challenging to capture effectiveness, as this, by definition, involves fine-grained observations about how hearers process referring expressions. Furthermore, whether a given referring expression is effective depends on the surrounding linguistic and non-linguistic properties of the referential scene, i.e., the *situated context*. Because of the number and complexity of the different factors involved, empirical research findings about what types of referring expressions (e.g., overspecified or not) are optimally effective, and under what circumstances, can be hard to generalize.

An increasing number of computational and empirical studies has been concerned with modeling or analyzing referential effectiveness. Computational models have frequently approximated the problem by generating expressions that are as *humanlike* as possible, i.e. that are optimized for resembling those produced by human speakers in similar contexts (see e.g. Viethen (2011) for an overview). However, empirical findings are mixed as to the extent to which human-produced references are easy for hearers to understand (e.g., Keysar et al. (2003)). Another common approximation is to assume that one can generate effective references by considering attributes of referents for selection one by one, as specified by a fixed preference order (Dale and Reiter, 1995). Yet empirical studies provide evidence that referential preferences of speakers and hearers (supposing that hearers prefer easy-to-understand expressions) vary according to dynamically evolving aspects of a referential scene's context (see, e.g., van Deemter et al. (2012)). The problem of understanding the exact influence of different aspects of context on referential preferences, and modeling effective reference production under these influences, has remained unsolved.

In the present work, we aim at addressing this problem. We propose an alternative approach to reference generation, which draws a distinction between less helpful and more helpful human behavior, and is explicitly concerned with modeling the latter. We are able to make this distinction by using an interaction corpus in a 3D environment, which records the hearers' reactions along with the speakers'

referring expressions.[2] On this corpus we train a maximum entropy model of the helpfulness of each attribute of a referring expression, given a certain scene. We then design and implement a reference generation model, mSCRISP, that incorporates the derived statistical estimations into the problem-solving technique of automated planning, to compute, for each referential context, a distinguishing referring expression of optimal estimated effectiveness. We find that mSCRISP manages to serve the needs of hearers well, while generating references that resemble those produced by effective human speakers. The model outperforms baseline models on referential effectiveness, in both automatic and human task-based evaluation. These results allow us to argue that, because mSCRISP is able to optimize effectiveness under selected, explicitly formalized aspects of situated context, it can serve as a methodological framework for computational and empirical research on referential effectiveness.

Plan of the chapter. In the remainder of this chapter, we first review state-of-the-art computational models of reference and discuss them in the light of empirical findings about referential effectiveness. We then introduce the planning-based approach to sentence generation, which enables the generation of semantically valid references, and illustrate how we are able to rank these references according to their effectiveness by obtaining a statistical account of context-dependent attribute preferences. We go on to show how we combine these two types of reasoning to derive our model mSCRISP. Finally, we evaluate the model and discuss possible improvements and implications for future computational as well as empirical research.

4.2 Referential effectiveness: Computational models and empirical insights

The objectives of a computational model of reference are subject to the nature of the generation task at hand. In news or narrative discourse generation, for example, in which descriptive reference plays a major role (Hervas and Finlayson, 2010), it might be important to explore the breadth of human creativity in producing descriptions whose functionality goes beyond activating referents (Maes et al., 2004). In procedural discourse (Longacre, 1983), on the other hand, in which the

[2]Though this corpus is written, for the sake of simplicity we use the terms "speaker" and "hearer" for both spoken and written language settings.

speaker tells the hearer how to accomplish a given task, references primarily have the distinctive function of helping the hearer identify entities involved in the task. To serve this function, models need to generate effective referring expressions; otherwise an expression would be of small use to a hearer, regardless of how natural or fluent it might sound. In this section we examine, in the light of insights gained from empirical research, two main ways in which state-of-the-art computational models have typically approached this problem: optimizing humanlikeness and using fixed attribute preference orders.

4.2.1 The effectiveness of humanlike references

Computational models. State-of-the-art computational approaches to distinctive reference often aspire to generate referring expressions that are as humanlike as possible. For instance, Viethen et al. (2008) tune the parameters of the graph-based algorithm of Krahmer et al. (2003) by computing attribute costs from the TUNA corpus (Gatt et al., 2007) in order to model the redundancy often found in human-produced references. Other approaches apply machine learning to human-produced data with richer representations of the situational context in their domains, with the purpose of varying their output in ways similar to human speakers (e.g., Jordan and Walker (2005); Stoia et al. (2006b)). In general, this line of research primarily attempts to replicate the referring expressions produced by humans, under the assumption that human-produced references are also, for the large part, effective (Viethen, 2011).

Empirical insights. This is by no means an unfounded assumption; several studies have shown that human reference production is often hearer-oriented and specially designed to facilitate the identification process of the hearer. Particularly in interactive dialog settings, speakers and hearers have been observed to systematically collaborate towards establishing mutually acceptable forms of reference with an aim of minimizing their joint effort (Clark and Wilkes-Gibbs, 1986)[3]. References thus often become partner-specific, making identification easy for their particular addressee but not so much so for an overhearer or a new hearer (e.g., Schober and Clark (1989); Brown-Schmidt (2009)). Such audience-design mechanisms have been argued to be strong and early-onsetting (e.g., Brennan

[3] See also an interesting computational treatment of Clark and Wilkes-Gibbs' collaborative reference model by Heeman and Hirst (1995). This earlier computational model addresses reference generation using methods from automated planning, as we also do in this work.

and Hanna (2009)). Even in non-interactive settings, common characteristics of human-produced referring expressions such as overspecification have been found to speed up identification (e.g., Arts et al. (2011)).

Another large body of research, however, provides conflicting evidence. Keysar et al. (2003), for instance, suggest that interlocutors sometimes fail to take the conceptual perspective of their partner into account during procedural interaction. Wardlow Lane and Ferreira (2008) find that speaker-internal cognitive pressures can be so powerful that they may override speaker-external communicative pressures, even when that threatens referential success. In the TUNA shared task on referring expression generation, Gatt et al. (2009) evaluate both human-produced and system-generated expressions using automatic measures of humanlikeness, human judgments of adequacy and fluency, as well as the referential clarity measures of accuracy and identification speed. The results provide compelling evidence that human-produced referring expressions are not necessarily effective: Measures of humanlikeness and referential clarity are not found to correlate in any significant way; in fact, human-produced referring expressions are systematically and significantly outperformed in terms of identification speed by the expressions that some of the systems generate for the task.

Conclusions. We conclude that both the speaker's beliefs about the hearer's ease of comprehension and the speaker's own ease of production can influence human reference production. Factors such as the speaker's cognitive load, the extent to which considerations of the hearer are salient in the interaction, as well as the severity of the consequences of being unclear, are all likely to play an important role in determining how the tension between speaker- and hearer-oriented processes will be resolved (Roßnagel, 2000; Haywood et al., 2005). However, quantifying the exact influence of these and any other relevant factors is still a matter of ongoing experimental work. Optimizing humanlikeness therefore does not necessarily guarantee optimal effectiveness. To overcome this problem, computational models can directly assess human-produced references for their effectiveness and aim at reproducing only the ones among them that are effective. This is the approach we explore in this work.

Some recent computational works share this view and make explicit attempts to tailor models' outputs to hearers' needs. Paraboni et al. (2007), for example, present rule-based models that can deliberately generate redundant expressions in order to make referents in hierarchically structured domains easy to identify. The models can generate e.g. the redundant expression "the library in room 120 in

Cockcroft" instead of the likewise distinguishing but less useful "the library", as a means of helping a hearer locate the library of a university campus for the first time. Similarly, Guhe (2009) presents a model that decides upon the inclusion of color as an attribute of a referring expression according to the probability that the hearer knows the referent's color. Golland et al. (2010) present a model of a "rational speaker", which is based on a maximum entropy learner and generates references optimally with respect to an embedded hearer model. Also reinforcement learning techniques have been used to adapt to (human or simulated) users and optimize task success (e.g., Janarthanam and Lemon (2010); Dethlefs et al. (2011)). Nonetheless, most of these approaches have not yet been tried in tasks in which good content determination choices are less obvious, addressing problems of a broader scope is required, or realistic interactions with human hearers are involved. In this work, we address the problem of effective reference generation in complex situated context, and test the performance of our approach in a shared-task human evaluation.

4.2.2 The effectiveness of fixed preferences

Computational models. Another approach commonly followed by computational models of reference for tackling the attribute selection task is the use of fixed preferences for processing attributes. This approach is motivated by the assumption that human speakers consistently prefer the use of certain forms of referring expressions over others, depending on factors such as the cognitive load of the speakers themselves or their hearers while processing these expressions. Indeed, speakers often prefer for example to include perceptually salient attributes of referents (such as color) in their expressions, even when this results in overspecified utterances (Pechmann, 1989). Such observations have been taken as evidence that each domain of reference may have its own, fixed, attribute preference order, based on which computational models should consider attributes for inclusion in a referring expression (e.g., Dale and Reiter (1995); Kelleher and Kruijff (2006); Gatt et al. (2007); Viethen et al. (2008)).

Empirical insights. However, psycholinguistic studies increasingly suggest that referential choices speakers make are not fixed throughout an interaction. Fukumura et al. (2010), for instance, show that linguistic and non-linguistic context bears upon the choice of a pronoun over a repeated noun phrase when speakers refer back to a referent in a preceding utterance. The influence of the situational

context on referential choice is not restricted to pronouns: Goudbeek and Krahmer (2010) find that attribute choice and modifier ordering are subject to priming and adaptation mechanisms arising in an interaction, while Koolen et al. (2011) further observe that attribute choice in both spoken and written human reference production is affected by features of the communicative setting as well as the referent. Also a corpus study by Viethen and Dale (2008) concludes that factors such as the salience of potential landmarks in a scene can encourage speakers to use e.g. more spatial relations in their referring expressions.

Not only speakers' but also hearers' preferences (supposing that hearers prefer easy-to-understand referring expressions) seem to be sensitive to the situational context. For example, even though overspecification in referring expressions can under certain circumstances—especially when it allows the hearer to create a mental image of the referent or limit their search process—speed up identification (Arts et al., 2011), in other cases it can hinder it. In particular, the study of Engelhardt et al. (2011) shows that unnecessary attributes in referring expressions can actually impair comprehension in simple visual scenes, even when they are realized in a syntactically unambiguous way. This negative effect arises also from the color attribute, which may come as a surprise; color has typically been considered a highly preferable attribute, whose use, even when unnecessary, is favored by speakers and hearers alike (e.g., Dale and Reiter (1995)). In any case, it remains an open issue how such results would generalize in more complex referential scenes with richer visual and other types of context.

Conclusions. All considered, modeling referential preferences as fixed and uniform for all situations in a referential domain may not necessarily lead to optimally effective expressions, as it does not seem to reflect mechanisms of either human-made choices or choices that result in easy-to-understand output. Instead, computational models of reference can increase their effectiveness by making their choices sensitive to dynamic aspects of a scene's context. For instance, as van Deemter et al. (2012) argue, preference for the color attribute in a visual scene should be subject to the degree of its perceptibility, among other factors, which in turn might depend on how far in the given scene referents are located. Further potentially important factors that van Deemter et al. (2012) draw attention to include the discriminatory power and the "extremity" of an attribute in the scene, intentional influences, as well as the dynamically evolving mechanisms of alignment. Although identifying the relevant aspects of linguistic and non-linguistic context that come into play and measuring their effect is certainly not a trivial task in complex scenes, we show with our model how such a task could be approached.

4.3 Planning referring expressions

Our model builds upon CRISP (Koller and Stone, 2007), an approach to natural language generation which handles the full sentence-generation problem as an automated-planning problem. Although we only use this approach to generate individual noun phrases here, these are in fact part of an expressive integrated sentence planning and realization process, which has also been extended to the generation of entire discourses (Garoufi and Koller (2010); see Chapter 3). Generation with CRISP involves the following two main stages: converting a language-generation problem into an automated-planning problem, and providing a solution to the former by solving the latter.

4.3.1 Converting language generation problems into planning problems

As a source of linguistic knowledge about the expressions it generates, CRISP utilizes a *lexicalized tree adjoining grammar* (Joshi and Schabes, 1997). In this formalism, each lexical item is associated with an elementary tree encoding a certain phrasal structure, as in Fig. 4.1. Such trees can be combined with each other by means of substitution and adjunction operations, as specified by the grammar. This way increasingly larger trees are derived, which correspond to sentence constituents and, ultimately, full sentences. To allow for the generation of meaningful expressions, we enrich the lexicon with semantic and pragmatic information in addition to the syntactic information it encodes. CRISP obtains awareness of the domain entities a hearer knows about, their semantic content and the relations holding between them, by tapping into a *knowledge base* that models the referential scene. Given an example knowledge base $\{\mathsf{button}(b_1),\ \mathsf{red}(b_1),\ \mathsf{button}(b_2),\ \mathsf{blue}(b_2),\ \mathsf{left\text{–}of}(b_2, b_1),\ \mathsf{chair}(c_1)\}$, and a communicative goal that requires describing b_1, Fig. 4.1 shows with a simplified version of the lexicon how the grammar derives the expression "the red button" referring to b_1.

In order to arrive at the generation of this expression, CRISP converts the lexicon of Fig. 4.1 and the given communicative goal into an *automated-planning problem* (Ghallab et al., 2004), which is the problem of finding a sequence of operators whose execution will achieve the specified goal. In this conversion, the entries of the lexicon are encoded as individual planning *operators*. The *preconditions* of an operator determine which logical propositions must be true in a given

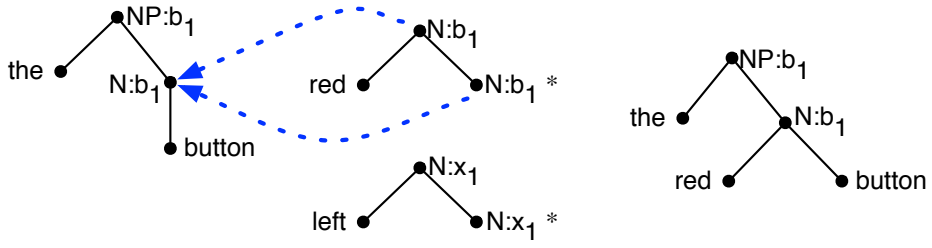


Figure 4.1: A simplified example of a CRISP lexicon and the derivation of the referring expression "the red button" describing b_1.

planning state so that the operator can be executed, while its *effects* specify how the truth conditions of these propositions will change after the execution. The operators integrate linguistic and non-linguistic preconditions and effects, as shown in simplified form in Fig. 4.2; in particular, the operators **red** and **left**, which encode a context-independent and a context-dependent attribute, respectively, include preconditions determining the eligibility of an entity to be described as "red" or "left" at a given state of the derivation (Garoufi and Koller (2010); see Chapter 3). The planning problem adopts the facts of the knowledge base in its *initial state*, and sets as its *goal* the fulfillment of the communicative goal, along with the satisfaction of a set of constraints. Syntactic constraints postulate syntactic completeness, while semantic constraints require that any entity referred to can be distinguished from other entities, called *distractors*, thus making sure that the generated referring expressions are *distinguishing*.

4.3.2 Solving planning problems to generate language

This conversion makes it possible to generate referring expressions by reasoning about how the available lexical entries can be organized into correct and distinguishing descriptions of the referents, as encoded formally in the planning problem. CRISP outsources this task to an off-the-shelf dedicated planning system, which allows it to benefit from the efficiency of modern planning algorithms. Let us examine step by step what reasoning a planning system may follow for the generation of an expression describing b_1, given the example knowledge base we specified and the operators of Fig. 4.2. The system can rule out as distractors for b_1 any entities that are not buttons, by executing the operator **the–button** applied to an available syntax node n_1 and the entity b_1, i.e. the *action* **the–button**(n_1, b_1). This rules out c_1, since it is a chair, but the second button of the knowledge base

red(u, x):
PRECOND: $\mathsf{canadjoin}(\mathrm{N}, u), \mathsf{referent}(u, x), \mathsf{red}(x), \ldots$
EFFECT: $\forall y.\big(\neg\mathsf{red}(y) \rightarrow \neg\mathsf{distractor}(u, y)\big), \ldots$

left(u, x):
PRECOND: $\forall y.\neg\big(\mathsf{distractor}(u, y) \wedge \mathsf{left\text{–}of}(y, x)\big)$,
$\mathsf{canadjoin}(\mathsf{N}, u), \mathsf{referent}(u, x), \ldots$
EFFECT: $\forall y.\big(\mathsf{left\text{–}of}(x, y) \rightarrow \neg\mathsf{distractor}(u, y)\big), \ldots$

the–button(u, x):
PRECOND: $\mathsf{subst}(\mathsf{NP}, u), \mathsf{referent}(u, x), \mathsf{button}(x), \ldots$
EFFECT: $\forall y.\big(\neg\mathsf{button}(y) \rightarrow \neg\mathsf{distractor}(u, y)\big), \neg\mathsf{subst}(\mathsf{NP}, u), \ldots$

Figure 4.2: Simplified CRISP planning operators for the lexicon of Fig. 4.1, as in Garoufi and Koller (2010) (see Chapter 3). Predicates subst express that a syntax node is open for substitution, referent connect syntax nodes to the semantic individuals to which they refer, and canadjoin indicate the possibility of a tree adjoining the given syntax node.

b_2 remains as a distractor. Because it has a goal of eliminating all distractors, the system goes on to check the preconditions of other potential actions. It finds that **left**(n_1, b_1) is not applicable, as the knowledge base stipulates that b_2 is the left one among the two buttons, and this entails that b_1 fails to satisfy the action's preconditions. However, action **red**(n_1, b_1) is applicable, as b_1 is red. Since this action now eliminates b_2 (which is blue) as a distractor, and the noun phrase is syntactically complete, the planner has achieved its goal and can terminate. CRISP finally realizes the computed *plan*, i.e., the goal-reaching sequence of actions found, as the referring expression "the red button".

4.4 A statistical account of referential effectiveness

Though the symbolic reasoning of CRISP guarantees the generation of semantically valid and distinguishing referring expressions (given a correct and complete model of the referential scene), these expressions are not necessarily optimal with respect to their effectiveness. In this section, we explain how we obtain a statistical account of effective referential choices in situated context. We start off with a human-human interaction corpus, in which we analyze hearers' reactions after being presented with referring expressions. This enables us to distinguish

between more effective and less effective references. We also model contextual properties of the referential scenes in order to capture the effects of different aspects of context on referential effectiveness. The resulting annotations allow us to train a maximum entropy learner that can predict, for new referential contexts, whether a reference will likely be effective or not.

4.4.1 Situated reference in the GIVE-2 corpus

We use the GIVE-2 corpus of Giving Instructions in Virtual Environments[4] (Gargett et al., 2010). In this corpus, a human instruction giver (IG) types text to guide a human instruction follower (IF) through a virtual 3D world, with the goal of completing a treasure-hunting task. The worlds comprise entities of different types (e.g., movable objects such as furniture pieces and immovable features of rooms such as doorways), including buttons on walls, which IFs can approach and press. Fig. 4.3 presents a top-down view of one of the three corpus worlds.

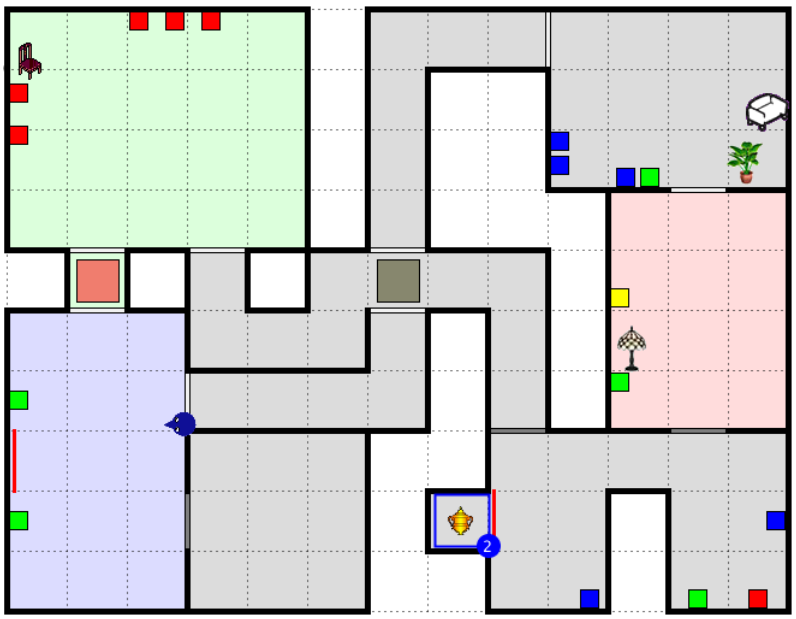


Figure 4.3: Map of a virtual world from the GIVE-2 corpus.

[4]The GIVE-2 corpus is freely available and viewable online at: `http://www.give-challenge.org/research/page.php?id=give-2-corpus`.

During the interactions, IGs refer to a series of *target* buttons, which are buttons that IFs need to press in order to progress in the task. In Gargett et al. (2010) we have created an annotation scheme for these referring expressions, which classifies them according to the basic types of attributes of which they are made up, as shown in Table 4.1. Applying this scheme to the full English edition of the corpus, we find that human IGs describe target buttons most frequently in terms of their color (which is the only absolute attribute used), type, and spatial relations with respect to the hearer, landmarks or button distractors in the scene. In this work, we focus on the six most frequent attribute types, as illustrated in the upper six entries of Table 4.1. Of the 714 referring expressions annotated, 598 only use attributes of these types.

	Attribute type	**Freq. (%)**
1.	**Absolute** (color; e.g. "red", "blue")	79.83
2.	**Taxonomic** (object type; e.g. "button", "box")	59.80
3.	**Viewer-centered** (e.g. "on the right", "the left one")	19.33
4.	**Micro-level landmark intrinsic** (spatial relation with respect to a movable landmark; e.g. "by the chair", "next to the couch")	17.37
5.	**Macro-level landmark intrinsic** (spatial relation with respect to an immovable landmark; e.g. "close to door", "on other side")	8.54
6.	**Distractor intrinsic** (e.g. "across from yellow button", "to the left of the blue button")	7.00
7.	**History of interaction** (e.g. "same", "from before")	5.60
8.	**Visual focus** (e.g. "that", "in your view")	5.32
9.	**Elimination** (e.g. "other", "wrong")	4.48
10.	**Relative** (e.g. "first", "middle")	4.34

Table 4.1: Attribute type annotations and their relative frequency (i.e., proportion of annotated references that contain an attribute of the given type) in the English edition of the GIVE-2 corpus. In this work, we focus on the six most frequent types.

4.4.2 Measuring effectiveness

In this task-based setting, we can assess whether a referring expression to a target button has served its purpose by examining whether it led the IF to press the intended referent. A manual annotation, based on this examination, reveals that 92% of all expressions referring to target buttons in the corpus allow the IF to cor-

rectly identify the referent (regardless of how long it takes them). We can achieve a more even split of the data by assuming that an IF who understands the expression easily will walk towards the correct referent quickly and directly; in other words, the average speed at which they approach the referent is an indication of effectiveness. We thus define the measure of *successfulness* $succ(r)$ of a referring expression r, which is intended to model computationally the linguistic property of referential effectiveness, as follows:

$$succ(r) = \begin{cases} 0 & \text{if } r \text{ was not correctly resolved} \\ \dfrac{\Delta S}{\Delta T} & \text{otherwise,} \end{cases} \tag{4.1}$$

where ΔS is the distance in the GIVE world (including turning distance) between the target button and the IF's location at the time when they encounter the expression r, and ΔT is the time elapsed between encountering r and pressing the referent. We can now split the referring expressions into a class of high successfulness and one of low successfulness, as follows:

$$succ^*(r) = \begin{cases} 0 & \text{if } succ(r) \leq \tilde{S} \\ 1 & \text{otherwise,} \end{cases} \tag{4.2}$$

where $\tilde{S}$ is the median of all values that $succ(r)$ takes for all referring expressions r in the data. This *binarized successfulness* abstracts away from the exact numeric value of an expression's successfulness, which is not important for our purpose, and allows us to create a balanced dataset with two classes of equal size. We examine this modeling choice further in the discussion section of the chapter.

4.4.3 Modeling the situated context of referential scenes

Referents (and distractors) in our corpus are situated in particular spatio-visual configurations and are associated with certain properties (e.g., discourse history). We call such sets of objects, with any properties they have, *referential scenes*. The surrounding linguistic and non-linguistic properties of referential scenes characterize their context. We formalize this notion of situated context via a collection of *context variables*, which represent individual measurable properties of scenes. Though variables can be defined to capture other possibly important influences of context on referential choices, such as the codability of the referent's attributes in a scene (Viethen et al., 2012), we focus on the hearer's conceptual accessibility

of the referent here (e.g., Fukumura et al. (2010); Arts et al. (2011)). Table 4.2 presents the basic set of ten variables we define on these grounds.

	Variable	Definition	Values
	OBJECT RELATIONS		
1.	*RoomSameTypeDisNum*	the number of distractors of the same type as the referent in the room	numeric
2.	*MicroLandmarkInRoom*	whether there are any micro-level (i.e., movable) landmarks in the referent's room	$\{0, 1\}$
3.	*MacroLandmarkNearby*	whether there are any macro-level (i.e., immovable) landmarks near the referent	$\{0, 1\}$
	SPATIO-VISUAL		
4.	*Distance*	the distance (in GIVE space units; including turning distance) between the IF and the referent	numeric
5.	*Angle*	the angle (in radians) between the center of the IF's field of view and the referent	numeric
	REFERENT'S DISTINCTIVENESS		
6.	*ColorUnique*	whether the referent's color is unique (i.e., not shared by other objects) in the world	$\{0, 1\}$
7.	*LandmarkTypeUnique*	whether a landmark of unique type in the world exists in the referent's room	$\{0, 1\}$
	INTERACTION HISTORY		
8.	*Round*	the number of times the referent has become a target to press throughout a session	numeric
9.	*ReferenceAttempt*	the number of times the referent has been referred to in the same round	numeric
10.	*SeenDeltaTime*	the time elapsed (in seconds) since the referent was last seen by the IF	numeric

Table 4.2: Context variables of referential scenes.

We compute the values of these variables from the corpus automatically, except for the $Round$ and $ReferenceAttempt$ variables, which we annotated manually. Variables in the OBJECT RELATIONS and REFERENT'S DISTINCTIVENESS groups view the referent in relation to other objects, and could be defined in terms of different scopes of comparison: For the $ColorUnique$ variable, for instance, one could ask whether the referent has unique color among the objects (a) near it, (b) in the room, or (c) anywhere in the virtual world. We choose as scope for these variables the one that yields best results during the training of our statistical model, as indicated in Table 4.2. Notice that the $Angle$ variable subsumes whether a referent is visible, which likely exerts strong influence on referential choices (Stoia et al., 2006b).

4.4.4 A maximum entropy model of effectiveness in context

Now we combine the information we collected about human referential choices, their relative successfulness, and the context in which they were made, in order to train a maximum entropy model that can estimate the successfulness of any referring expression in any context. We assume that in a given context, all attributes of the same type as classified in Table 4.1 are equally effective for a hearer. Based on this assumption (which we examine in the discussion), we model a referring expression r as a set of attribute types, and let $a_j(r) = 1$ if r contains an attribute of the j-th attribute type of Table 4.1 ($j = 1, \ldots, 6$). Otherwise, $a_j(r) = 0$. For a referential scene s, we let $c_i(s)$ take the value of the i-th context variable of Table 4.2 in this scene ($i = 1, \ldots, 10$). We then combine attribute types and context variables into features of the form:

$$\phi_{ij}(r, s) = c_i(s) \cdot a_j(r). \tag{4.3}$$

These features allow us to cast the problem as a simple binary classification task, in which our goal is to estimate the conditional probability of a referring expression r presented in a scene s being highly successful, given a joint representation of attribute types and context:

$$P\Big(succ^*(r) = 1 \;\Big|\; \big\{\phi_{ij}(r, s)\big\}_{i,j}\Big). \tag{4.4}$$

We train a maximum entropy model to learn this distribution; this will later allow us to convert the model's parameters into parameters for planning in order to steer CRISP's attribute choices towards high successfulness. Maximum entropy models for binary classification tasks (high/low successfulness) are equivalent to logistic regression, as implemented e.g. in the Weka data mining workbench (Hall et al., 2009), which we use here. The model estimates the above probability as

$$\hat{P}\Big(succ^*(r) = 1 \;\Big|\; \big\{\phi_{ij}(r,s)\big\}_{i,j}\Big) = \frac{1}{e^{\sum_{i,j}\big(w_{ij}\,\cdot\,\phi_{ij}(r,s)\big) + w_0} + 1}, \tag{4.5}$$

for coefficients w_{ij} and intercept w_0. By letting

$$v_j(s) = \sum_i \big(w_{ij} \cdot c_i(s)\big), \tag{4.6}$$

we derive:

$$\hat{P}\big(succ^*(r) = 1 \;\big|\; r, s\big) = \frac{1}{e^{\sum_j \big(v_j(s)\,\cdot\,a_j(r)\big) + w_0} + 1}. \tag{4.7}$$

This way, we obtain a *weight* $v_j(s)$ for each attribute type $a_j(r)$ of a reference r in a scene s. In our data, we observe that every context variable of Table 4.2 affects the weight of at least one attribute type.

4.5 Optimizing effectiveness using metric planning

The weight of an attribute type as defined in (4.6) provides an estimation of that attribute's contribution to the successfulness of a certain reference in a scene, according to (4.7). In this section, we explain how we combine these statistical estimations with CRISP's symbolic reasoning about referential correctness. We first describe how we employ the formalism of metric planning to assign individual costs to CRISP's planning operators, associating each attribute type with an estimation of how preferable it is in the given context. We then illustrate how we work around planner limitations to derive our model, mSCRISP.

4.5.1 Assigning costs to attributes

We employ *metric planning* (Fox and Long, 2003), which is a form of automated planning that can handle numeric reasoning. This allows us to assign to each planning operator a numeric *cost*, such that the use of the operator in a plan will add its cost to the total cost of the plan. We further introduce a *plan metric*, which specifies that a planner should try to find a plan of minimal total cost; as it plans referring expressions, the system thus evaluates them according to their quality (as determined by their total cost), and looks for optimal-quality ones. Though off-the-shelf planners may not guarantee that they actually find an optimal plan for efficiency reasons, in practice the plans that our planner Metric-FF (Hoffmann, 2002) finds are close to optimal (see evaluation results). This way we can reduce the problem of computing an effective referring expression to that of planning under an appropriate cost metric.

Following the CRISP model, we represent each attribute value that we might want to include in a referring expression as a single planning operator of a planning problem, as in Fig. 4.2. The key problem we must now solve is to determine what cost to assign to each of these operators, so that the most preferable attribute choices receive the lowest costs. We can approach this by inspecting how the individual attribute weights $v_j(s)$ contribute to the value of the probability in (4.7). If for some j, $v_j(s)$ is a negative value in scene s, then $\hat{P}\big(succ(r) = 1 \mid r, s\big)$ is higher for a reference r such that $a_j(r) = 1$, rather than if $a_j(r) = 0$. That is, choosing to include the attribute a_j in this case will increase the probability that the resulting reference will be highly successful. If $v_j(s)$ is positive, then the effect is reversed: Choosing a_j will lower the probability of high successfulness. This means that, given a scene s, we can determine the optimal choice of attributes for a reference r by the attributes' weights in s, as follows:

$$a_j(r) = \begin{cases} 1 & \text{if } v_j(s) \leq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.8}$$

The effect that choosing a_j has on the probability grows with the absolute value of $v_j(s)$. It therefore seems natural to use $v_j(s)$ as the cost of operators for attributes of type a_j.

4.5.2 Working around planner limitations

We must address one final technical complication: Most off-the-shelf metric planners do not accept negative costs (because otherwise an action could be executed again and again in order to lower the total cost), but $v_j(s)$ may be a negative value in a scene s. Such negative-weight attributes improve the successfulness estimate of an expression even if they are not necessary to distinguish the referent, and we would like the generation model to include them in its (redundant) referring expressions.

We work around this problem by introducing, for each attribute type a_j, a new operator **non-a_j**. This operator does not correspond to a lexical entry and lacks any preconditions or effects pertaining to syntax or semantics. Its presence in a plan represents a deliberate choice not to include any attribute of type a_j in a referring expression. To enforce that the planner will consider every available attribute while making its choices, we further introduce formulas $\mathsf{needtodecide}(a_j, u)$ for each attribute a_j and syntax node u that holds a referring expression. These formulas convey that the planner needs to decide whether or not to include a_j in an expression. We ensure this by setting an additional planning goal that no needtodecide formulas remain at the end of the planning process. Finally, we insert $\neg$needtodecide effects in such a way that removing these formulas is possible only by executing operators for attributes of type a_j or the operator **non-a_j**. This means that, to arrive at a valid referring expression, the planner must decide, for every attribute, whether to include it in the expression or not.

The planner makes these decisions based on the cost that each outcome incurs. To favor good choices as dictated by (4.8), we assign the cost

$$\mathsf{cost}(a_j) = \max\bigl\{0, v_j(s)\bigr\} \tag{4.9}$$

to each operator that represents an attribute of type a_j, and the cost

$$\mathsf{cost}(\text{non-}a_j) = \max\bigl\{0, -v_j(s)\bigr\} \tag{4.10}$$

to the operator **non-a_j**. Notice that the cost of an attribute type depends on the referential scene s (as seen through the context variables). We present an example cost assignment for our six attribute types in Table 4.3.

j	a_j	$v_j(s)$	$cost(a_j)$	$cost(\text{non-}a_j)$
1.	**Absolute**	0.03	0.03	0.00
2.	**Taxonomic**	1.02	1.02	0.00
3.	**Viewer-centered**	−29.40	0.00	29.40
4.	**Micro-level landmark intrinsic**	3.41	3.41	0.00
5.	**Macro-level landmark intrinsic**	11.84	11.84	0.00
6.	**Distractor intrinsic**	−23.00	0.00	23.00

Table 4.3: Example of weights $v_j(s)$ in a scene s and corresponding cost assignments for each attribute type a_j.

We thus obtain a metric planning problem in which all operator costs are positive or zero and whose minimal-cost plans correspond to maximal-probability referring expressions. Because the original planning problem (as constructed by CRISP) already enforces that a referring expression must be distinguishing, this amounts to finding the referring expression of lowest cost among the distinguishing ones.

4.5.3 Generating referring expressions with mSCRISP

As an example of how the resulting model mSCRISP operates, consider the planning operators for the attribute value "red" and for **non-absolute**, shown in Figure 4.4. These replace the operator for **red** shown in Fig. 4.2; the other operators from Fig. 4.2 change analogously.

Let's suppose we have a knowledge base that contains $\{\mathsf{button}(b),\ \mathsf{red}(b)\}$, stating that b is a red button. The initial state of the planning problem might contain the formulas $\mathsf{subst}(\mathsf{NP}, n_1)$ and $\mathsf{referent}(n_1, b)$, indicating that we want to generate a noun phrase on syntax node n_1 that refers to b. Because n_1 holds a reference, there will also be formulas $\mathsf{needtodecide}(\mathsf{taxonomic}, n_1)$ and $\mathsf{needtodecide}(\mathsf{absolute}, n_1)$. (We ignore all other attributes for this example.) Now suppose that the planner executes the action **the-button**(n_1, b), deciding to include a taxonomic attribute and incurring its cost. This removes $\mathsf{needtodecide}(\mathsf{taxonomic}, n_1)$ from the planning state. At this point, the planner must consider executing either the action **red**(n_1, b), incurring the cost for an absolute attribute, or the action **non-absolute**(n_1), with the cost of not choosing an absolute attribute. One of the two actions must be executed eventually, as it is impossible to arrive at a final state before all needtodecide formulas have been

red(u, x):
PRECOND: canadjoin(N, u), referent(u, x), ...
EFFECT: ¬needtodecide(absolute, u), ...
COST: cost(absolute)

non-absolute(u):
PRECOND: needtodecide(absolute, u)
EFFECT: ¬needtodecide(absolute, u)
COST: cost(non-absolute)

Figure 4.4: Simplified mSCRISP planning operators for an attribute of type absolute.

removed. In case b is the only button in the domain, the choice between the two actions depends on which of cost(absolute) and cost(non-absolute) is greater. If, however, a distractor exists and it is not red, the planner may be forced to apply **red** in order to distinguish b from that distractor—regardless of the relative costs. mSCRISP does not compute the cheapest combination of arbitrary attributes, but the cheapest combination among those that result in a distinguishing referring expression.

4.6 Automatic evaluation

To assess the adequacy of our approach, we evaluate mSCRISP with respect to intrinsic and extrinsic measures. In this section, we present an automatic evaluation study against a purely statistical and a purely symbolic baseline model in referential scenes of a GIVE-2 corpus world (shown in Fig. 4.3). We find that our model generates more highly successful references than the purely symbolic baseline, according to the estimations of (4.7), and that its references are more similar to highly successful human-produced ones than those of either of the baselines.

4.6.1 Methods

The models. We design two baseline reference generation models to compare mSCRISP against. The *MaxEnt* baseline builds a reference in a scene s by selecting all attributes of type a_j for which $v_j(s) \leq 0$, as prescribed by (4.8). That is, this baseline makes its choices exclusively based on the successfulness estimations of the maximum entropy model of (4.7), without combining those with

IG	Referring expression
Human	*the green button on the left*
MaxEnt	*the button to the left of the picture*
EqualCosts	*the left button, to the left of the right button*
mSCRISP	*the button to the left of the picture*

Table 4.4: Referring expressions produced by a human instruction giver, our model mSCRISP and the two baselines MaxEnt and EqualCosts in the bottom-left room of Fig. 4.3.

reasoning about the semantics of the resulting references like mSCRISP does. This is therefore a purely statistical model, which does not verify the applicability or discriminatory power of the attribute types it selects, and thus makes no correctness or uniqueness guarantees. The *EqualCosts* baseline, on the other hand, is a version of our mSCRISP model in which all attribute costs are equal. That is, unlike mSCRISP and the MaxEnt baseline, this baseline does not choose attributes by considering their contribution to successfulness according to (4.8). It is a purely symbolic model which computes correct and distinguishing referring expressions, but does this without any guidance about their expected successfulness. Finally, because we conduct this evaluation in referential scenes of a GIVE-2 corpus world, we also have human-produced references (*Human*) to compare the models' choices against.

Table 4.4 presents example referring expressions that these three different IGs produce for one of the buttons in the bottom-left room of Fig. 4.3. As the IF is entering the room, they see from left to right a green button, a picture, and another green button. All referring expressions in this example are distinguishing. However, the human-produced expression, which favors the use of an absolute ("green") and a viewer-centered ("on the left") attribute over one pointing to the micro-level landmark ("to the left of the picture"), was not particularly effective in the scene: After encountering it, the IF spent time scanning the room further to the left before finally approaching the referent. The MaxEnt baseline and mSCRISP generate a different expression, using a landmark, which they estimate to be more effective. By contrast, EqualCosts's referring expression is correct but more complex.

Procedure. We train the maximum entropy model of (4.7) on a dataset consisting of referring expressions in the virtual worlds 1 and 2 of the GIVE-2 corpus.

We then perform automatic evaluations on a test set consisting of expressions in world 3 (Fig. 4.3). Specifically, we use mSCRISP and the two baselines to generate expressions for the referents in the test corpus, and estimate the probability that the generated references fall into the high successfulness class. We construct the knowledge bases of the planning-based generation models mSCRISP and EqualCosts to include the objects that are visible by the IF within the target referent's room, and we restrict ourselves to those scenes in which the target is among these objects. Finally, to determine to what extent mSCRISP aligns effectiveness and humanlikeness, we look at the similarity of the generated references to those originally produced by the human IGs in the corpus. We model this similarity by the *Dice coefficient* metric (Dice, 1945; Gatt et al., 2007), which, examining references on the level of attribute selection (rather than lexicalization or surface realization), is in line with the focus of this work.

In both the training and the test set, we include only referential scenes in which (a) the referent is in the same room as the IF (so that it is visible by the IF or near them; this is meant to reduce the interference of navigation instructions), and (b) the referring expression only contains the attribute types shown in Table 4.1. This amounts to 358 referential scenes in the training set and 174 scenes in the test set.

4.6.2 Results

Accuracy of successfulness estimations. The *accuracy* of the maximum entropy classifier, i.e. the proportion of references in the given scenes whose binarized successfulness is estimated correctly according to (4.7), differs between the training and test set. On the training data, the accuracy is 75.1%; on the test data, it is 62.1%. This compares favorably to a majority classifier, which would achieve 50% accuracy on the training dataset (since it is balanced); that is, the maximum entropy model does learn to predict successfulness to some degree. The difference in accuracy indicates that the training and test data are varied enough for a fair evaluation. In addition, the drop suggests that more training data might further improve mSCRISP's overall performance.

Probability of being highly successful. Table 4.5 presents, for each model, the average probability (4.7) that the references it produces fall under the high successfulness class. We find that the MaxEnt baseline significantly outperforms all other models. This is not surprising, as the metric of evaluation here is exactly what this baseline is designed to optimize for. However, MaxEnt picks the differ-

IG	Prob. of high successfulness
Human	0.467***
MaxEnt	**0.984****
EqualCosts	0.649***
mSCRISP	0.957

Table 4.5: Average probabilities of high successfulness. Differences to mSCRISP are significant at $^{**}p < .01$, $^{***}p < .001$ (paired t-tests).

ent attributes independently, ignoring whether the resulting expression is semantically informative; correctness and uniqueness of a referring expression are not captured by the statistical model. Of the reference generation models which warrant that the generated expression refers uniquely, mSCRISP performs the best.

Humanlikeness. Table 4.6 presents average Dice coefficient results, both for all references in the test set and for those of high and low human-achieved successfulness separately.

	DICE		
IG	low succ.	high succ.	all
MaxEnt	0.320***	0.449*	0.371***
EqualCosts	**0.512**	0.475	**0.497**
mSCRISP	0.457	**0.519**	0.482
# references	78	51	129

Table 4.6: Average DICE coefficients across datasets. Differences to mSCRISP are significant at $^{*}p < .05$, $^{***}p < .001$ (paired t-tests).

This test reveals that the expressions computed by the MaxEnt baseline are less humanlike than those computed by either of the planning-based generation models. This can be explained by the fact that, in contrast to MaxEnt, the planning-based models generate their expressions on the basis of a set of referential correctness and uniqueness principles, which are, at least to some extent, shared by humans. Though the difference is not statistically significant, mSCRISP reaches a higher degree of humanlikeness than EqualCosts on references of high successfulness; this is reversed in the low-successfulness dataset. The distinction is relevant

because mSCRISP does not attempt to mimic human IG choices under all circumstances; it only does so when it believes that these choices are highly effective. If this is not the case, it makes different choices—those that a more effective human IG might make.

4.7 Human task performance evaluation

The automatic evaluation results rely on the estimations of a statistical model, which may not be fully representative of the effectiveness of references in scenes with human hearers. To assess the performance of our model in the context of real interactions, we participated in the Challenge on Generating Instructions in Virtual Environments[5] (GIVE-2.5; Striegnitz et al. (2011)). Systems participating in this shared task engaged in the role of generating written instructions to guide human IFs through a virtual treasure hunt. The virtual worlds were designed to be similar in nature to the GIVE-2 corpus worlds that were available for training, but also to provide reference generation models with challenges of varying complexity. We next present results of this human evaluation, focusing on the model's referential performance. We find that mSCRISP's references are resolved correctly more often and faster than those of our symbolic baseline, and lead to fewer errors on behalf of hearers than those of any other model participating in the shared task.

4.7.1 Methods

The models. We implemented mSCRISP and the purely symbolic baseline EqualCosts (which is the only one of our two baselines that always generates correct and distinguishing referring expressions) as parts of GIVE natural language generation systems. Both systems operate by first generating a *first-attempt* reference for a given target button as soon as the IF is in the target's room and can see the target. Subsequently, they generate *follow-up* references at regular intervals until the IF responds by either pressing some button or navigating away from the target. Fig. 4.5 shows an example of a reference situated in one of the GIVE evaluation scenes, as generated by mSCRISP. In this scene, EqualCosts would generate the different expression "the right one to the right of the green button".

[5]Further information on the GIVE Challenge as well as evaluation results are available at: `http://www.give-challenge.org/research`.

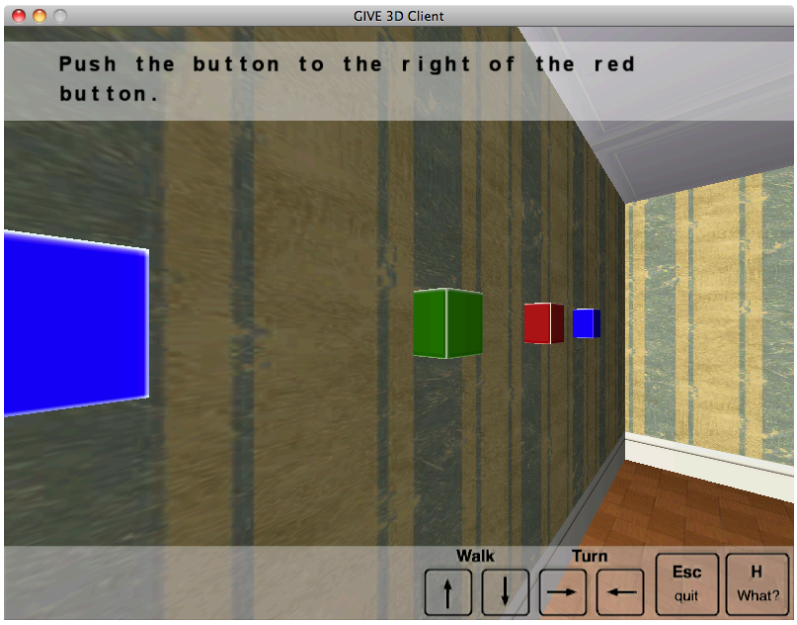


Figure 4.5: Example of a reference situated in the context of a GIVE evaluation scene, as generated by mSCRISP.

In addition to our own models, six other reference generation models participated in the GIVE-2.5 Challenge: Systems *A*, *C*, *L* and *T* generate references following hand-crafted approaches, while systems *B* and *CL* both base their referential choices on human production; CL selects individual references from a dedicated human-human interaction corpus, and B constructs references based mostly on a decision tree learnt from the GIVE-2 corpus. This latter system represents a supervised-learning approach applied to the same corpus as we use, which, however, optimizes references for humanlikeness rather than effectiveness.

Procedure. We first compare mSCRISP against EqualCosts with respect to two metrics of referential success: *resolution success*, which represents the rate of expressions whose intended referents have been correctly identified by the IF (regardless of how fast), and successfulness, as defined in (4.1). Then, we compare the model's performance with the other models that were entered into the shared task evaluation. Though resolution success and successfulness rates are not immediately available for comparison, Striegnitz et al. (2011) report on similar measures that reflect to what degree IFs could identify the systems' intended referents. In particular, the *error rate* of a system denotes the average number of incorrect button presses over the total number of actions performed in a single interaction.

We must draw this comparison with caution, since the different approaches of systems to execution monitoring and repair may also bear on the prevention of errors. However, the mSCRISP system only uses simple execution monitoring techniques (see Garoufi and Koller (2011b)); and in any case, one may expect that if referring expressions are effective, misunderstandings are less likely to arise in the first place.

Both mSCRISP and EqualCosts generate follow-up references at regular intervals, which may differ from first-attempt ones. Such references are important for the GIVE task, yet the fact that they are generated regardless of whether the IF is on the right track or not poses a problem on assessing referential success. We therefore base our analysis of resolution success and successfulness only on first-attempt references. To control for the effect of rephrasing, we separately examine the subset of references for which all follow-up references were *non-rephrasing*, i.e. exactly the same as the original. The results are derived from a total of 536 human-system interactions, which Striegnitz et al. (2011) collected over the Internet.

4.7.2 Results

Comparison against our own baseline. Table 4.7 presents average resolution success and successfulness rates for mSCRISP and the purely symbolic baseline.

	Resolution success		**Successfulness**	
IG	all	non-rephrased	all	non-rephrased
EqualCosts	86%***	86%	0.32	0.38***
mSCRISP	**95%**	**89%**	**0.33**	**0.52**

Table 4.7: Average resolution success and successfulness results in the shared task. Differences to mSCRISP are significant at $^{***}p < .001$ (Pearson's χ^2 test for resolution success rates; unpaired two-sample t-tests for the rest).

In terms of resolution success, we find that mSCRISP significantly outperforms the baseline. Though the results are measured on different datasets and are thus not directly comparable, mSCRISP's success rate of 95% surpasses the 92% success rate of human IGs in the GIVE-2 corpus. The system's performance remains

better than the baseline's, though not significantly so, in the non-rephrased reference dataset. Turning to the metric of successfulness, the two systems do not differ significantly when all first-attempt references are considered. However, rephrasing may affect an IF's response, since processing new expressions can take additional time. Examining the portion of non-rephrased first-attempt references, we find that the system does generate expressions that human IFs are able to resolve significantly faster.

Comparison against other reference generation models. Striegnitz et al. (2011) report upon error rates for all GIVE-2.5 participating systems as in Table 4.8.

	A	**B**	**C**	**CL**	**L**	**T**	**EqualCosts**	**mSCRISP**
	21%	49%	10%	11%	12%	19%	15%	9%
Error			A	A	A	A	A	A
rate	B		B	B	B	B	B	
		C						

Table 4.8: Average error rate results as in Striegnitz et al. (2011), putting mSCRISP to comparison against EqualCosts and the six other systems participating in the shared task. Two systems do not share the same letter if the difference between them is significant ($p < .05$; ANOVA and post-hoc Tukey tests).

We observe that, although pairwise Tukey's tests do not find all differences to be statistically significant, mSCRISP outperforms the other systems with respect to this final measure of referential performance. Further comparative results from Striegnitz et al. (2011) rank mSCRISP among the top systems on most objective and subjective evaluation measures, including overall duration and task success.

4.8 Discussion

The evaluation shows that mSCRISP can serve the needs of hearers well, while generating references that resemble those produced by effective human speakers. The model derives its ability to refer effectively both from its planning component, which ensures that references are correct and distinguishing, and from its corpus-based learning. In both components, we made a number of modeling choices

and assumptions. In this section, we discuss these choices and how the model could be improved by exploring alternatives for these. Finally, although mSCRISP is primarily a computational approach to reference generation, it may support empirical studies into human reference processing. We end the discussion by examining some implications of this work for future computational and empirical research.

4.8.1 Improving the model

Our definition of successfulness in (4.1) as a metric that models referential effectiveness considers the time window from the presentation of a referring expression until the hearer's action. This definition captures two important aspects of referential understanding: *interpretation*, which relates to determining the meaning of an utterance, and *resolution*, which involves identifying the referent, once a referring expression has been interpreted (Paraboni et al., 2007). On the other hand, this time window also includes the process of (simulated) physical interaction with the referent. The cognitive load of this subtask likely affects the hearer's reaction speed and may also need to be factored in. Apart from that, processing a scene has an intrinsic cognitive load due to the inherent characteristics of that specific scene, which an improved metric of effectiveness would ideally account for. Another limitation is that it may be harder to detect referential understanding in domains without physical interaction. In certain situated domains, this may be achieved using other observable cues, such as eye movements (Staudte et al. (2012b); see Chapter 5). In any case, the precise quantification of referential effectiveness is a matter for further research.

The model chooses attributes assuming that in a given context, all attributes of the same type as classified in Table 4.1 are equally effective for a hearer. This grouping allows us to make good use of our limited training data, but is an oversimplification: For instance, color and shape are both absolute attributes, but their inclusion in a referring expression can result in significantly different identification times for hearers (Arts et al., 2011). Similarly, color may be more preferable in describing an elephant that is pink than in describing a gray one (van Deemter et al., 2012). Since the model assigns an individual cost to each lexical entry, it could be refined in such a way that every attribute (or even every value of an attribute) receives by the maximum-entropy learner its own cost. Given this kind of refinement, the mechanism of context variables may go a long way toward accounting for such context-dependent preferences (see, e.g., the $ColorUnique$ vari-

able of Table 4.2). The set of context variables we selected here was intended as a starting point and could be extended to capture additional aspects of the linguistic and non-linguistic context.

Finally, an error analysis shows that the most problematic expressions the model generates are referring expressions with recursive structure such as "the button to the left of the right button", "the button below the upper button", and variants of these. These constructions arise primarily because our account of effectiveness focuses on attributes of the main referring expression and does not consider any embedded ones; e.g. it does not address the problem of optimizing the noun phrase complement "the right button" in "[the button to the left of [the right button]]". It turns out that the baseline EqualCosts was more prone to this kind of expressions than mSCRISP, which at first glance could be hypothesized as a possible reason for the lower referential success rates of that model. However, examining the portion of non-rephrased expressions of each model that do not display this particular structure, we still find that mSCRISP's references were significantly more effective. Despite this fact, extending the model to optimize embedded references could further improve its performance.

4.8.2 Implications for computational research

This work stands among other recent approaches that design referring expressions to suit the hearers' needs. Because of the complexity of this problem, models so far have mostly focused on simpler tasks, e.g. whether or not to redundantly use the room-number and building-name attribute in a reference to a place that the hearer is not familiar with (Paraboni et al., 2007). By contrast, our approach makes it possible to optimize effectiveness in richer communicative settings, where a model has a higher number of non-trivial choices to make. Capturing effectiveness with a maximum entropy classifier is natural, in that maximum entropy models try to make minimal assumptions about the probability distribution beyond the empirical observations. Similar models have also been used to capture the process of an agent making a choice between discrete alternatives in other domains (Train, 2009). While we used it for a binary classification task here, it would be interesting for future computational work to employ this statistical model for a more fine-grained characterization of effectiveness.

The model is sensitive to diverse aspects of the situated context and can dynamically adjust its output to match the degree of saliency of different objects. Because it has access to information about the interaction history, it can also adapt

its output interactively. As an example, the $ReferenceAttempt$ context variable (Table 4.2) might be able to steer mSCRISP towards choosing a highly overspecified new reference, if it records that the model has unsuccessfully attempted to refer to a given referent before. Because it integrates sentence planning and realization, the model is not limited to content determination but can do full-fledged generation of references as parts of sentences. This could help generation models overcome the limitations of producing one-shot references in a null context and move towards references in which the surrounding linguistic (and non-linguistic) context also plays a role. As our approach is not domain-specific, it could transfer to other domains. Exploring grammar design and cost assignment for different domains would be interesting directions for future computational work.

4.8.3 Implications for empirical research

Our model functions in situated context, where spatio-visual and other non-linguistic context bears on referential preferences. Because of the interactive communicative setting we consider, it is possible to study reference as part of a longer interaction rather than as an isolated process. Empirical studies can thus be conducted in a more complex and realistic domain, which may be useful in generalizing the observations made in simple visual scenes (e.g., Engelhardt et al. (2011)). At the same time, such studies can often benefit from explicit formalization of the mechanisms involved (see e.g., Krahmer (2010)). With its computational modeling of the situated context through context variables, mSCRISP formalizes this notion.

From a hearer's perspective, mSCRISP is based on a model of reference comprehension: For a given referring expression and scene, it predicts how easily the hearer will resolve the expression to the referent the speaker had in mind. The choice of a specific set of variables determines the aspects of the context that the model will take into account when making this prediction, and we can add further context variables or take some away. For example, the EqualCosts model we used as a baseline in the evaluation can be seen as an extreme variant of mSCRISP with no context variables. One might thus be able to assess the influence of individual variables on referential effectiveness by correlating the predictions that the model makes using different sets of variables with the comprehension behavior of human hearers.

Conversely, we might learn something about human reference production by examining human-produced referring expressions in the light of the ones that the

model generates under the same setting. Human speakers do not always produce optimally effective references, and a question that arises is to what degree are human-produced references suboptimal. Because mSCRISP always generates correct and distinguishing references according to its model of what the hearer knows about, it explores levels of optimality that go beyond avoiding inappropriate use of privileged ground (e.g., Wardlow Lane and Ferreira (2008)); it seeks to optimize a reference according to a more fine-grained account of the hearer's conceptual accessibility of referents and distractors in a scene (e.g., Fukumura et al. (2010); Arts et al. (2011)). By comparing such different degrees of optimality against human production, empirical research might be able to study the question of just how effective human references are from a new angle.

4.9 Conclusion

In this chapter we presented mSCRISP, a computational model that generates referring expressions that are directly optimized for effectiveness in situated context. Computational models of reference generation often approximate effectiveness as humanlikeness, but there has been recent empirical evidence that human-produced referring expressions may not always be optimally effective. Our model therefore learns to recognize human-produced referring expressions that are effective, and only aims at reproducing those. Because it recomputes the estimated contribution of each attribute of a referent to effectiveness based on the current situated context, mSCRISP does not rely on inflexible attribute preference orders like other state-of-the-art approaches. We have shown that mSCRISP indeed generates more effective referring expressions than baseline models, both in automatic and in full human task performance evaluations. The model formalizes the notion of situated context and could serve as a methodological framework for empirical research on referential effectiveness.

Chapter 5

Exploiting listener gaze to improve situated communication in dynamic virtual environments[1]

Beyond the observation that both speakers and listeners rapidly inspect the visual targets of referring expressions, it has been argued that such gaze may constitute part of the communicative signal. In this chapter, we investigate whether a speaker may, in principle, exploit listener gaze to improve communicative success. In the context of a virtual environment where listeners follow computer-generated instructions, we provide two kinds of support for this claim. Firstly, we show that listener gaze provides a reliable real-time index of understanding even in dynamic and complex environments, and on a per-utterance basis. Secondly, we show that a language generation system that uses listener gaze to provide rapid feedback improves overall task performance in comparison with two systems that do not use gaze. Beyond demonstrating the utility of listener gaze in situated communication, our findings open the door to new methods for developing and evaluating multimodal models of situated interaction.

[1]This chapter is based on: Konstantina Garoufi, Maria Staudte, Alexander Koller, and Matthew Crocker. Exploiting listener gaze to improve situated communication in dynamic virtual environments. Under review for journal publication.

5.1 Introduction

In situated spoken-language interaction—where interlocutors are communicating in a shared physical environment and messages are often grounded in the visually co-present surroundings—it is perhaps unsurprising that gaze and speech are closely intertwined. Speakers tend to fixate objects they are about to mention, while listeners inspect those objects and events that they believe to be the intended referents of the speaker. Perhaps more surprising are the temporal dynamics of this gaze behavior, and its synchronization with the speech signal in particular: Speakers typically fixate objects about one second prior to their mention until just before speech onset (Meyer et al., 1998; Griffin and Bock, 2000); listeners, in turn, begin to fixate candidate referents within about 200 ms of hearing them mentioned (Tanenhaus et al., 1995; Allopenna et al., 1998). While these production- and comprehension-contingent gaze behaviors reveal much about the nature of the cognitive processes that generate them, what is particularly interesting from a communicative perspective is that gaze may form an integral part of the signal itself, complementing speech in much the same way as gesture does. That is to say, interlocutors may potentially exploit the information conveyed by their partner's gaze behavior. For their part, listeners may monitor speaker gaze if this provides reliable information about the referent the speaker intends (Hanna and Brennan, 2007; Staudte and Crocker, 2011). At the same time, the rapid nature of listener gaze, in response to either speaker gaze or speech, raises the possibility that listener gaze may be exploited by speakers to enhance communicative success in real time, as suggested in studies by Clark and Krych (2004).

What is less clear is the extent to which a (human or artificial) speaker in such a situation might be able to enhance referential success, in particular, by using listener gaze as a direct index of whether or not the listener has understood a given referring expression. Such a hypothesis rests on two fundamental assumptions, namely (i) that listener gaze is reliably and rapidly directed towards understood referents in a manner that speakers can detect, and (ii) that such gaze can then be exploited by the speaker to timely resolve misunderstandings or uncertainty on the part of the listener. While studies from the visual world paradigm provide evidence for (i), such findings may be contingent upon the highly simplified and static visual settings. It remains an open question, whether such behaviors generalize to the more complex and dynamic environments in which natural situated communication takes place. Furthermore, such studies emphasize the average gaze behavior over many trials; but in order to exploit listener gaze in real-time interactions, the speaker must be able to decode listener gaze in response to a

single utterance. Evidence for (ii) remains at best suggestive. While Clark and Krych (2004) present clear evidence that interlocutors pay attention to each others' gaze as part of coordinating their dialog and requesting help, they offer no systematic evidence regarding the use of referential gaze. One reason for this is that it is difficult to simultaneously make the setting truly dynamic and accurately track listener gaze. In addition, there are challenges in eliciting sufficiently consistent and numerous referring expressions—and gaze-driven feedback—to make a quantitative assessment of the hypothesis.

In this chapter, we investigate whether the assumptions (i) and (ii) hold true in dynamic, complex environments and on a trial-by-trial basis, and exploit our findings to improve a natural language generation system. To overcome the mentioned limitations of previous studies, we utilize a 3D virtual environment in which the experimental participant may move around freely. The participant follows instructions that are automatically generated in real time by a computational model of a speaker (henceforth, the *system*), and we monitor listener gaze behavior and task performance. Specifically, the system guides the listener through a series of rooms towards a prize. En route, the listener is instructed to press a number of buttons, which are described to the listener by means of referring expressions. The system not only knows where the listener is, which direction they are facing, and which button must be pressed next, but also has access to the real-time gaze behavior of the listener. Thus, when the system generates a referring expression, such as "Push the button to the left of the lamp", it can rapidly exploit listener gaze to determine whether the listener has indeed understood which button is meant, or not, and provide relevant feedback.

Using this setup, we first examine question (i): Is the listener's gaze reliably and rapidly directed towards the understood referents, despite the high complexity of the dynamic 3D environment? In order to answer this question, we map the positions of the listener's gaze on the screen (as reported by an eye-tracker) to objects in the 3D scene. We then record what objects in the scene the listener inspects in response to each utterance of the automated natural language generation system; thus the linguistic stimuli are produced in a systematic, algorithmic way, while still being variable enough to support the communicative requirements of such a task. Our hypothesis was that listeners would rapidly direct their gaze towards the understood referents, in much the same way as in the 2D visual world experiments on language comprehension (e.g., Allopenna et al. (1998)). Our findings confirm this hypothesis, revealing increased inspection of the understood referent in the region immediately following disambiguation of a given referring expression.

We then explore question (ii): Can the speaker's performance in identifying referents to the listener be improved by taking listener gaze into account? To this end, we equip the natural language generation system with the capability to respond to inspections of objects by the listener in real time. When the listener inspects the button to which the system has referred, the system immediately gives positive feedback ("Yes, that one"); when the listener inspects any other button, the system gives negative feedback ("No, not that one"). We compare this system with two baseline systems that either provide no feedback or do not use listener gaze to provide feedback. Our prediction was that the gaze-based system would outperform the other two. We find that the system indeed lowers listener confusion in comparison with the other two, while improving referential success as compared with the system that does not provide feedback, and affording more timely feedback as compared with the alternative feedback-enabled system. These results also speak indirectly to (i), in that they show that listener gaze is, in fact, such a reliable indicator of the listener's comprehension process that it can be exploited for each individual utterance.

With these findings, we contribute to cognitive and computational research in situated dialog, while also advancing current methodologies. Previous results from the visual world paradigm have shown that listeners rapidly fixate referents over distractors—on average—in static scenes, even when scenes are quite complex (Andersson et al., 2011). Our findings go further in demonstrating that listener gaze offers a reliable index of online referential understanding, within a single trial, and in dynamic, task-oriented environments. We show that a computational model of the speaker that exploits such listener gaze in order to provide appropriate feedback results in improved task performance, as revealed by a range of metrics. Not only does this serve as direct evidence for the benefits of gaze monitoring by interlocutors in situated communication, it also addresses one of the fundamental problems in the development of computational dialog systems—namely, how to determine whether the system has been understood, and give proactive feedback to support the user when this fails to happen. Our results indicate that the communicative performance of such systems can be improved by taking real-time eye-tracking information into account.

Plan of the chapter. The rest of the chapter is structured as follows: In the next section, we look at relevant psycholinguistic as well as computational findings in greater detail, in order to put the current study into context. We then introduce our experimental setting and describe our method, before reporting on the results. In particular, we present an analysis of listener inspection patterns during refer-

ential processing, including the processing of relative spatial adjectives, and an evaluation of referential understanding as indicated by a range of task-based metrics. We finally discuss the importance of our findings both for empirical and for computational research, and conclude the chapter.

5.2 Related work

Previous empirical research has shown that listeners align with speakers by visually attending to mentioned objects (Tanenhaus et al., 1995; Allopenna et al., 1998) and, when possible, to what the speaker attends to (Richardson and Dale, 2005; Hanna and Brennan, 2007; Staudte and Crocker, 2011). Numerous eye-tracking studies have demonstrated that listeners process referring expressions incrementally, by taking into account the context to interpret each word rapidly after it has been heard (e.g., Eberhard et al. (1995); Sedivy et al. (1999); Weber et al. (2006); Wolter et al. (2011)).

In analyzing such gaze behavior, experiments have traditionally used simple and static 2D visual-world scenes, and have analyzed the recorded listener gaze offline (e.g., Altmann and Kamide (1999); Knoeferle et al. (2005)). Although certain studies involving a situated speaker have included some dynamics in their stimuli, this is normally constrained to speaker head or eye movements (Hanna and Brennan, 2007; Staudte and Crocker, 2011; Macdonald and Tatler, 2013), and does not account for changes in the surrounding physical (or virtual) environment, as an agent navigates and interacts with it. Furthermore, these studies assume a simplified communicative setting in which the speaker's behavior and utterances are fixed in advance, and do not respond to the listener's eye movements. This means that an important part of the reciprocal nature of interaction, of the kind that naturally arises in collaborative, goal-oriented situations, cannot be captured.

One insightful study that emphasized interactive communication in a dynamic environment was conducted by Clark and Krych (2004). In this experiment, two partners assembled Lego models: The directing participant advised the building participant on how to achieve that goal, while it was manipulated whether or not the director could see the builder's workspace and, thus, use the builder's visual attention as feedback for directions. Clark and Krych found, among other things, that the visibility of the listener's workspace led to significantly more deictic expressions by the speaker and to shorter task completion times. However, the chosen experimental setting introduced large variability in the dependent and

independent variables, making controlled manipulation and fine-grained observations difficult. In fact, we are not aware of any empirical work that has integrated features of real-life communicative settings and the reciprocal nature of listener-speaker adaptation, while still being able to measure relevant eye-movement data.

Computational approaches, on the other hand, model the process of grounding (Clark and Schaefer, 1989), in which a system decides to what extent the user has understood its utterance and whether the communicative goal has been reached. Observing the user behavior to monitor the state of understanding is a key component in this process. A full solution may require plan recognition or abductive or epistemic reasoning (see e.g., Young et al. (1994), Hirst et al. (1994)); in practice, many systems use more streamlined (Traum, 1994) or statistical (Paek and Horvitz, 1999) methods. Spoken dialog systems traditionally focus on the verbal interaction of the system and user, and the user's utterances are therefore the primary source of evidence in the monitoring process (e.g., Skantze and Schlangen (2009); Buß and Schlangen (2010)). In this work, by contrast, we monitor the user's non-verbal reactions, and in particular their gaze.

Finally, in the context of multi-modal communication, previous computational works have employed robots and virtual agents as speakers to explore when and how speaker gaze may help listeners to ground referring expressions (Foster, 2007). For instance, the performance of a system for resolving human-produced referring expressions can be improved by taking the (human) speaker's gaze into account (Iida et al., 2011). Gaze has also been used to track the general dynamics of a dialog, such as turn taking (Jokinen, 2010). Here, we are interested in monitoring the listener's—rather than the speaker's—gaze, in order to determine whether they have understood a referring expression. To our knowledge, there has been little previous research on this; especially in dynamic 3D environments. The closest earlier work of which we are aware has emerged from the recent Challenges on Generating Instructions in Virtual Environments (GIVE; Koller et al. (2010b); Striegnitz et al. (2011)), which have been introduced as a shared task for interactive, situated natural language generation systems. These systems typically approximate listener gaze as visibility of objects on the screen and monitor grounding based on such (or similar, readily observable) data (e.g., Denis (2010); Racca et al. (2011)). Though we also use the GIVE setting in this work, we deviate from earlier approaches in that we propose monitoring the communicative success of utterances based on eye-tracking.

5.3 Methods

5.3.1 The 3D environments

We used the interactive setting of GIVE[2], a task-based game where a human user can move about freely in an indoor virtual environment with a number of interconnected corridors and rooms. In this setting, a 3D view of the environment is displayed on a computer screen as in Fig. 5.1, while the user can walk forward/backward and turn left/right by using the cursor keys. The user can also press different-colored box-shaped buttons attached to the walls, by clicking on them with the computer mouse once they have navigated close enough.

Communication in GIVE arises in the context of a virtual treasure hunt, where the task is to find a trophy inside a hidden safe. To reveal and open the safe, users have to press particular buttons in a certain order; however, as they do not have any prior knowledge of the environment, they have to rely on instructions that a real-time natural language generation system provides to them. The system, by contrast, has complete knowledge of the environment as well as the user's location, and its role is to generate directions and referring expressions to the relevant buttons in order to guide the user through the task.

Crucially, individual rooms in the virtual environment may contain several buttons other than the *target*, which is the button that the user at a certain moment has to press next. Thus, users have to distinguish the target from these other buttons, which we call *distractors*. Next to buttons, rooms also contain a number of *landmark* objects, such as chairs, wall pictures and plants, which cannot directly be interacted with, but can be used in references to nearby targets. We call an entire game, up to the successful discovery of the trophy, an *interaction* of the system and the user. At any point, the user can press the 'H' key on the keyboard to indicate to the system that they are confused (perhaps because they did not understand the previous utterance) and need help. We call a press of the 'H' key a *help request*.

For our experiment, we used three virtual environments by Gargett et al. (2010), which were designed to differ in their spatial and visual properties and to provide reference generation systems with challenges of varying complexity. A top-down map of one of these three environments is shown in Fig. 5.2; this is the environment in which the scene of Fig. 5.1 arose.

[2] `http://www.give-challenge.org/research`.

Figure 5.1: A first-person view of a virtual 3D environment, as seen by users during the interactions.

5.3.2 Recording object inspections

We employed a faceLAB eye-tracking system[3] to record which objects in the virtual environments users inspect during the interactions. At intervals of approximately 15 ms, the system determines the (x,y) position on the screen that the user is looking at. Inferring inspections of objects in the 3D scene from these (x,y) positions is not trivial, because the user may not look precisely at a pixel on the screen that represents this object. The situation is exacerbated by the fact that users can move freely in the virtual environment, and when they move or turn, all objects on the screen shift to the side, sometimes faster than the user's gaze can follow. We therefore applied the following heuristic algorithm to automatically map (x,y) positions to objects in the 3D scene.

When the 3D engine renders the 3D scene onto the 2D screen, it assumes a certain position of the "camera" in the 3D environment; this roughly corresponds to the position of the user's eyes. For each object that is currently visible, the system computes its bounding box, i.e. the smallest box that completely contains

[3] `http://www.seeingmachines.com/product/facelab`.

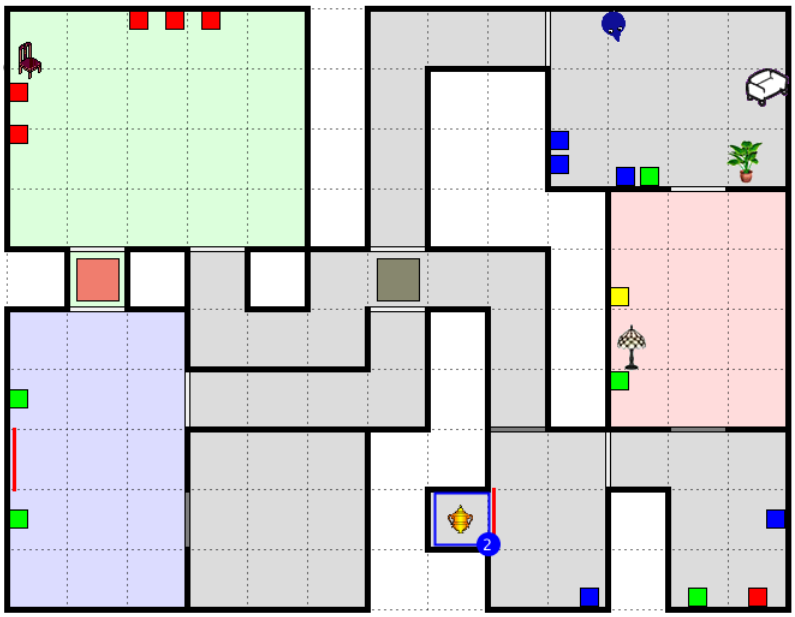


Figure 5.2: A map of the environment in Fig. 5.1; note the user in the upper right room.

the object. It then determines the minimum angle α between the ray from the camera position to some corner of the bounding box and the ray from the camera position to the center of the bounding box. Intuitively, α represents the size of the object on the screen. The system also determines the angle β between the ray from the camera position to the (x,y) position in the screen plane reported by the eye-tracker and the center of the bounding box. Small values of β represent situations in which the user looks directly at the center of an object. An object is a candidate for being inspected if one of β/α or $\beta - \alpha$ is below a certain threshold. This corresponds to checking whether (x,y) is within a circle around the center of the object on the screen whose radius scales with the size of the object on the screen. Among all candidates (if there are any), the system then chooses the object with the smallest β.

The system finally recorded an inspection of the object if it detected that the user continuously looked at the same object for a certain amount of time; for this experiment, we chose a threshold of 300 ms. If such an inspection was interrupted by less than 150 ms, the system considered the second fixation to that object to be a continuation of the initial inspection and thus counted both to the same, "continuous" inspection.

5.3.3 The natural language generation systems

One challenge in investigating speech-mediated listener gaze in a dynamic environment such as ours is that one cannot predict where, exactly, in the virtual environment the experimental participant will be located when they hear a spoken stimulus, or what they will see at that time. This means that we cannot rely on pre-specified stimuli. Instead, we utilize natural language generation systems to algorithmically compute stimuli in real time that are suitable for the particular scene the participant faces at that time. This method allows us to achieve the variability of stimuli that is necessary for a dynamic environment, while at the same time maintaining systematic control over the kind of referring expressions that the participant is exposed to.

We implemented three different such natural language generation systems, which are identical in most respects. All three systems generate simple navigational instructions that guide the user to a location from which the next target is visible (e.g., "Go through the doorway"). Once this has been achieved, they generate an instruction to press this button, which contains an expression referring to it (e.g., "Push the red button"). Additionally, each time the user makes a help request, the systems generate a new, *follow-up* instruction from scratch. Depending on how the spatio-visual context has changed, this new instruction may be the same as the original one or phrased differently. A *rephrasing* follow-up referring expression, for instance, includes a different (and possibly larger) set of attributes than the original one, increasing the chance that the user will understand it. Unlike in the original GIVE setting, which presents written instructions, all generated instructions are converted to speech by the Mary text-to-speech system (Schröder and Trouvain, 2003) and presented via loudspeaker.

The systems' referring expressions are optimized for being easy to understand, according to a corpus-based model of understandability (Garoufi and Koller (2011b); see Chapter 4). They are always correct, unique descriptions of the target as seen by the user at the moment at which generation starts, but vary in terms of their linguistic complexity. While some expressions are as simple as "the button", others include pre-modifiers or post-modifiers or both. Pre-modifiers are either *absolute* adjectives (color; e.g. "red" or "blue") or *relative* spatial adjectives such as "left" or "right". Post-modifiers, on the other hand, often include an embedded expression referring to an object other than the target button (e.g., "the left button to the left of <the blue button>"), which we call an *anchor* object. Both landmark and distractor objects are regularly chosen by the generation systems as anchors.

Two of the three systems attempt to predict whether a user has understood a given referring expression or not. If a system predicts that the user understood the referring expression correctly, it gives the user positive feedback by uttering "Yes, that one". On the other hand, if the system predicts that the user misunderstood the expression, it gives negative feedback by uttering "No, not that one". This feedback is made available *proactively*, i.e. before the user has reacted to the referring expression by pressing a button, and is intended to increase their confidence (if positive) or prevent them from making a mistake (if negative). Because the three natural language generation systems differ only in whether and how they make such predictions to provide feedback to their users, any differences in their task performance can be directly attributed to the presence and quality of this prediction mechanism.

No-feedback

As a baseline, we used a system which does not actively monitor whether a listener seems to have understood a referring expression correctly. The baseline system never provides any (positive or negative) feedback on its own initiative, and will only generate a follow-up referring expression in response to a help request. We call this system the *no-feedback system*.

Movement-based monitoring

The second system attempts to predict whether the user understood a referring expression based on their movements in the virtual environment. This *movement-based system* is intended to represent the monitoring that can be implemented, with a reasonable amount of effort, on the basis of immediately available information in the GIVE setting. In particular, the system employs the following heuristic. If more than one button in a room is visible to the user after a referring expression has been generated, the system remains inactive—it can be difficult to arrive at a reliable prediction in this case. If, however, at some point only a single button in the user's room is visible, then the system starts monitoring the user's *overall distance* from this button, where the overall distance is a weighted sum of the walking distance to the button and the angle the user must turn to face that button. In case the system records that the user has decreased this distance by more than a given threshold since they started moving, it then concludes that the listener has resolved the referring expression as the given button, and goes on to provide the

corresponding (positive or negative) feedback.

Gaze-based monitoring

Lastly, the *eye-tracking-based system* attempts to predict whether a user has understood a referring expression by monitoring their gaze. In contrast to the movement-based system, which only starts monitoring the user under specific circumstances, this system starts its monitoring as soon as the user has heard the referring expression. Specifically, the system draws from an eye-tracker to monitor object inspections as described above. Once it has detected an inspection of a button in the room, the system then generates the corresponding feedback. This system thus operates upon the assumption that listeners visually attend to what they perceive as referents, even when referring expressions are situated in complex and dynamic visual scenes.

Both the movement-based and the eye-tracking-based system withhold their feedback until a first full description of the referent (a *first-mention* referring expression) has been spoken. Additionally, they only provide feedback once for every newly approached or inspected button and will not repeat this feedback unless the user has approached or inspected another button in the meantime.

Example interactions of a user with each of the three systems are presented in Table 5.1. All three interactions were recorded during the systems' attempts to refer to the rightmost blue button shown in Fig. 5.1. The course of the interaction with the eye-tracking-based system, up to the onset of positive feedback, is also illustrated in Fig. 5.3. In this figure, the white circles around the rightmost button (which did not appear on the participant's screen during the experiment) represent the recorded gaze information: Smaller white circles render the trace of fixation coordinates, while the larger circle marks an inspection of the target, which then acts as a trigger to the system's positive feedback.

Full interactions of participants with each of the three systems, as recorded in the virtual environment of Fig. 5.2, are viewable online at:

`http://www.youtube.com/watch?v=pegJxuYqJTI` (Eyetracking),
`http://www.youtube.com/watch?v=Q8ZUr0Fnm3c` (No-feedback),
`http://www.youtube.com/watch?v=BXs_v9s81Mw` (Movement).

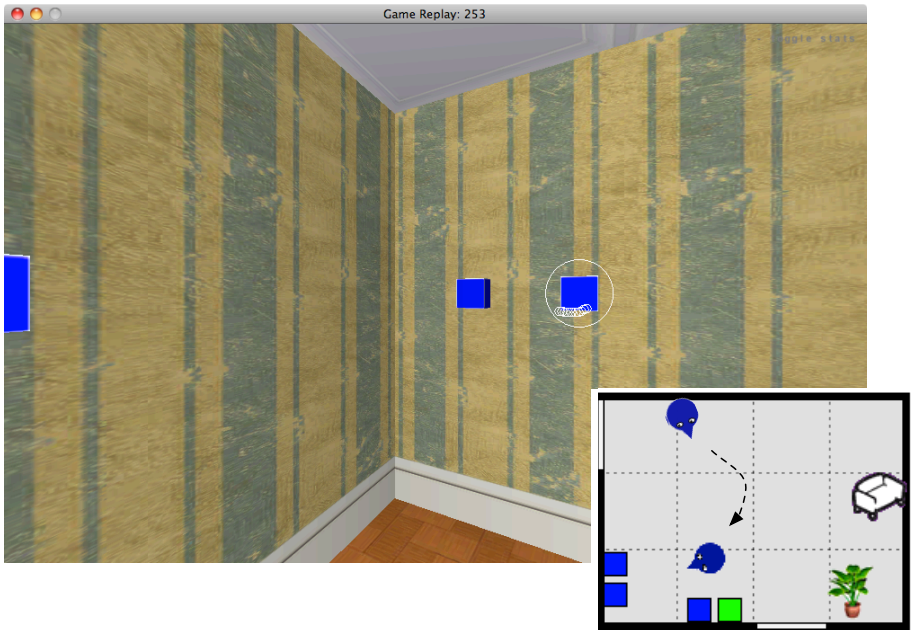


Figure 5.3: The course of a user's interaction with the eye-tracking-based system, following the instruction "Push the right button to the right of the green button" (see Table 5.1). The white circles around the rightmost button represent gaze information, as recorded by the system.

Figure 5.4: A faceLAB eye-tracking system remotely monitored participants' eye movements during the interactions.

System	Interaction
Eyetracking	S: *Push the right button to the right of the green button.* U approaches the pair of blue and green button and inspects one of them S: *No, not that one!* ... (U inspects other buttons in the scene, while S provides appropriate feedback) U inspects the correct button S: *Yes, that one!* U presses the correct button
No-feedback	S: *Push the right button to the right of the green button.* U presses the wrong blue button
Movement	S: *Push the right button to the right of the green button.* U approaches the pair of blue and green buttons; once U is very close to the blue button, it happens to become the only button visible on screen U continues moving closer to the blue button S: *No, not that one!* U has no time to react to S's feedback and presses the wrong blue button

Table 5.1: Example interactions between a participating user (U) and each of the three systems (S). All interactions were recorded during the systems' attempts to refer to the rightmost blue button shown in Fig. 5.1. The course of the interaction with the eye-tracking-based system (up to the onset of positive feedback) is illustrated in Fig. 5.3.

5.3.4 Participants and procedure

Thirty-one students (twelve females), enrolled at Saarland University, were paid to take part in this study. All reported their English skills as fluent, and all were able to complete the task. Their mean age was 27.6 years.

Before the experiment, participants received written instructions that described the task and explained that they would be given directions by a natural language generation system. They were encouraged to request additional help at any time they felt that the systems' directions were not sufficient (by pressing the 'H' key). During their interactions with the systems, a faceLAB eye-tracker remotely monitored participants' eye movements on a 24-inch monitor, as in Fig. 5.4. The

eye-tracker was calibrated using a nine-point fixation stimulus. We disguised the importance of gaze from the participants by telling them that we videotaped them and that the camera needed calibration.

Each participant started with a short practice session to familiarize themselves with the interface and to clarify remaining questions. We then collected three complete interactions, each with a different virtual environment and natural language generation system (the order of interactions was varied according to a Latin square design). Finally, participants received a questionnaire which aimed to assess whether they noticed that they were eye-tracked and that one of the generation systems made use of that. The entire experiment lasted approximately 30 minutes.

5.3.5 Data collection and analysis

We recorded all movements and actions of the participants in the virtual environments as well as all instructions given by the systems using the GIVE software.[4] The software automatically logged the participants' position, orientation and field of view every 30 ms, making it possible to analyze and replay the collected interactions in full. In addition, we recorded the raw eye-tracking data, from which object inspections for the analysis were automatically reconstructed. All events were timestamped based on the time they were recorded by the natural language generation server. Gaze events were detected by the eye-tracker and subsequently sent to the generation system over the local network. We found the network latency between the different machines to be low and fairly constant.

For the analysis of participants' referential processing in the collected data, we segmented the interactions into referential scenes. A *referential scene* (or *trial*) is a section of an interaction that starts at the onset time of an instruction with a first-mention reference to a target, and ends with the participant's reaction (pressing a button or navigating away to another room). Each referential scene contains only a single first-mention referring expression, but may include a number of follow-up referring expressions (whenever the participant requested help) and feedback utterances (whenever the system's feedback mechanism was triggered). Note that, even if they involve the same target, referential scenes may still look different from each other at their onset (and over their course), because participants moved around the virtual environments in different ways.

Further, to analyze how participants' eye movements developed according to

[4] `http://www.give-challenge.org/research/page.php?id=software`

the spoken stimuli, we subdivided referential scenes into five separate *time windows* as follows:

1. speaking onset until determiner onset (i.e., "Push")
2. determiner onset until offset of the head of the referring expression (i.e., "the [red/left/...] button")
3. if the referring expression used an anchor, the time from head offset until the onset of the determiner in the referring expression to the anchor (i.e., "to the left/right of"); if the referring expression did not use an anchor but had a post-modifier, the time from head offset until speaking offset ("in front of you/to your left/to your right"); otherwise empty time window
4. if the referring expression used an anchor, the time from the onset of the determiner in the referring expression to the anchor until speaking offset (i.e., "the picture/blue button/..."); otherwise empty time window
5. speaking offset until +500 ms after speaking offset

As an example, Fig. 5.5 presents a series of snapshots spanning one of the referential scenes recorded with the eye-tracking-based generation system; the depicted environment corresponds to the bottom right room of Fig. 5.2. This referential scene started with the onset of "Push the red button" in Fig. 5.5(b) (time window 1). While the referring expression was still being spoken, in Fig. 5.5(c), the participant was moving and turning towards the referenced target (time window 2). The onset of the system's feedback in Fig. 5.5(d), which occurred approximately 1600 ms after the offset of the referring expression, already falls outside time window 5.

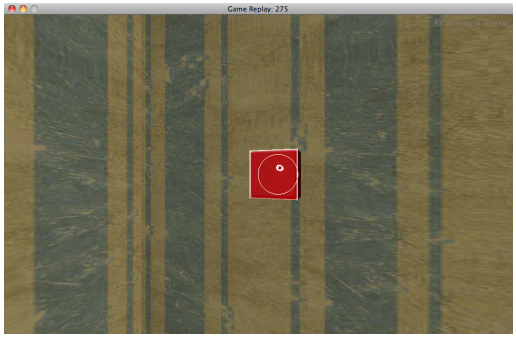
To remove errors in eye-tracker calibration, we included interactions with any system in the analysis only when we were able to detect inspections (to the target or any distractor) in at least 75% of all referential scenes of that interaction. This filtered out 18 interactions out of the 93 we collected. Note that this filter differs from the one applied to previous analyses of this dataset in Staudte et al. (2012b) and Koller et al. (2012), which only excluded poorly calibrated interactions with the eye-tracking-based generation system; this was necessary for an analysis of the participants' eye movement patterns across systems. We also discarded those individual referential scenes in which the systems rephrased their first-mention referring expressions, as the participant's interpretation of those may have been

(a) In response to a navigational instruction, the user is heading towards a certain room.

(b) The user enters the room. The system briefly acknowledges this, and subsequently generates "Push the red button". The onset of this spoken utterance marks the start of the referential scene and time window 1.

(c) While the expression referring to the target is being spoken (time window 2), the user moves towards the target and inspects it.

(d) Approximately 1600 ms after the offset of the referring expression, the eye-tracking-based system reacts to the inspection by means of positive feedback ("Yes, that one").



(e) The user goes on and presses the button. This action marks the end of the referential scene.

(f) The system briefly acknowledges the successful action and instructs the user to navigate away, in search of the next target.

Figure 5.5: A series of snapshots spanning a recorded referential scene with the eye-tracking-based generation system.

influenced by the content of the follow-up expressions. After this filtering, we retained 686 referential scenes from 75 interactions.

Based on these data, inferential statistics were carried out using mixed-effects models from the lme4 package in R (Baayen et al., 2008). Specifically, we used logistic regression for modeling binary data such as accuracy of participants' button presses, linear regression for analyzing durations of object inspections, and Poisson distributed regression for counts such as the number of 'H' keystrokes. Further, fixed effects and interactions (as well as random effects, including intercepts and slopes) were determined through model reduction, which assesses the contribution of a predictor or interaction to a fitted model by running a χ^2-comparison between models with and without the particular predictor(s) (see, for instance, Jaeger (2008)). Random intercepts and slopes were included in models (and explicitly mentioned) only when they accounted for a significant part of the variation in the data. Additionally, p values were calculated through Markov chain Monte Carlo (MCMC) sampling where necessary.

5.4 Results

5.4.1 Inspection of referents

Our first hypothesis is that listeners direct their gaze rapidly and reliably at the object to which they resolved a referring expression, even in dynamic and complex settings such as ours. To evaluate this hypothesis, we compare inspection durations between targets and distractors across the time windows described above. We only consider referential scenes which ended in the participant pressing the target without requesting help, and not in pressing a distractor. In these scenes, the referenced button coincides with the button that the listeners identified as the referent of the given referring expression. This means that, by comparing inspections of the target to inspections of distractors in these scenes, we effectively compare inspections of the understood referent to inspections of other referential candidates, while at the same time ensuring that the listener's interpretation of the given referring expression was correct.

In certain referential scenes, the referring expressions contained a subexpression describing the anchor. If the hypothesis holds, the listener may tend to inspect the anchor in addition to the target. We therefore exclude the anchor (if one exists) from the distractors, and only compare target inspections against

inspections of *non-anchor distractors*, i.e. distractors that were not referred to in the given scene. On the other hand, the set of buttons in the listener's view often changed, as a result of their moves and turns, from one moment to the next (see Fig. 5.5). In some cases, no non-anchor distractors have been visible on screen for the duration of a complete time window. To avoid overestimating the looks at the target, we therefore only consider time windows in which at least one non-anchor distractor has been visible. We then divide inspection times of distractors in a time window by the number of non-anchor distractors that have been visible in that time window, in order to obtain an average over the visually available non-anchor distractor and to provide a representable proportion of inspection time for any one distractor. In addition, as the same time window may have different durations in different scenes (because of different linguistic content), we normalize all inspection durations by the duration of the corresponding time window. Note that time windows may also vary in how many utterance fragments they average over. In particular, window 3 ("to the left of/to your left") involves only utterances that have a post-modifier, while window 4 involves only utterances with reference to an anchor ("the blue button/the picture").

Fig. 5.6 shows the proportion of inspection time spent by participants on the target and on an average distractor button. A linear mixed-effects model fitted to the inspection times with $participant$ as random factor—others such as $virtual{-}environment$ and $target{-}button$ did not add to a better fit of the model—revealed a significant difference between the button type (i.e., target versus distractor) in window 5 (longer target inspection time: Coeff. = .074, SE = .016, t = 4.517, $p(\text{MCMC}) < .001$). The graph indicates that participants initially inspected the target and the average distractor equally. After utterance offset, however, targets (i.e., the objects to which the participants resolved the referring expression) received clearly longer inspection time than the average distractors.

This result suggests that listener gaze, on average, indexes referential understanding in these complex environments immediately after linguistic disambiguation (prior to sentence offset, the target may be ambiguous). Note that the systems' feedback can be ruled out as a cause for the difference in inspection times in window 5. Although the generation systems may, in principle, generate positive or negative feedback within the first 500 ms after utterance offset, this did not happen in the scenes we analyzed here: There are only 15 scenes overall with feedback in time window 5, and all of these were excluded from the analysis of that time window because no non-anchor distractor was visible.

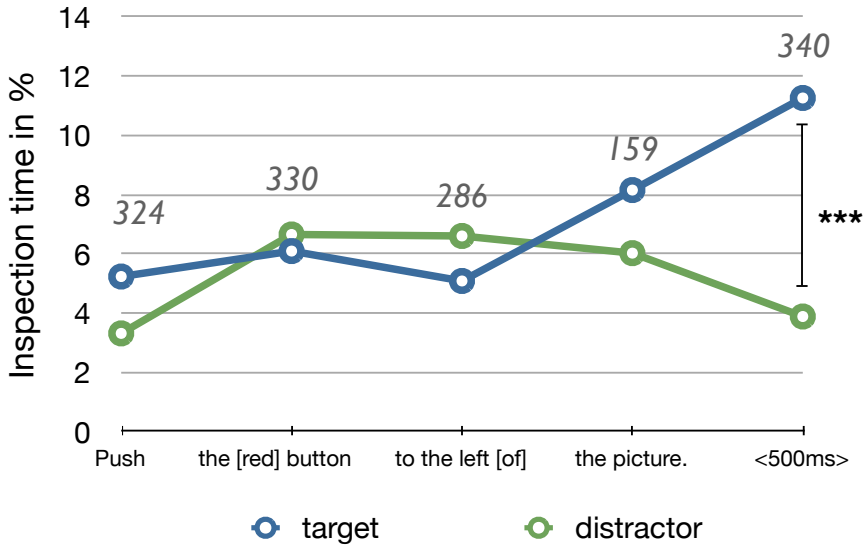


Figure 5.6: Average inspection time (% of time window) spent on target and non-anchor distractor buttons. Grey numbers represent the number of scene fragments falling into each time window. Differences between target and distractor inspection times are statistically significant at ***$p(\text{MCMC}) < .001$.

5.4.2 Visual processing of absolute and relative adjectives

While Fig. 5.6 indicates that the target gets inspected longer than the average distractor in the region immediately following the utterance, several studies within the visual world paradigm have found even earlier incremental influence of the referring expression on listeners' visual attention. Pre-nominal adjectives, for example, may disambiguate the referent even before the head noun appears: In the utterance "Push the red button", the adjective "red" may already be sufficient for listeners to identify the target (e.g., Eberhard et al. (1995)). We therefore examined our data in more detail to determine whether we could find evidence for similar behavior.

As a first step, notice that an early rise in target inspection times can only be expected at a time at which the referring expression has become unambiguous. In our data, this can only be assumed for referring expressions without post-modifiers. Thus, we focus on those utterances that are of the form "Push the red/blue/... button". In these utterances, the mean proportional inspection time on

target and non-anchor distractor buttons during time window 2 is 9.3% and 7.2%, respectively, and the difference is statistically not significant. We then consider a second factor, namely the fact that the natural language generation systems often included relative spatial adjectives in the referring expressions (e.g., "the left button"). The interpretation of relative adjectives such as "left" inherently involves a relation between the target and other objects in the scene, whereas absolute adjectives such as "red" directly relate to the target, without implicating other objects. One might assume that processing relative adjectives therefore leads to increased looks at objects other than the target. Indeed, previous research has shown that during the processing of relative terms listeners may start fixating referents later and spend more time fixating other objects instead (e.g., Sedivy et al. (1999)). We thus sub-categorize our data according to whether referring expressions contain a relative pre-modifier, or an absolute one, and specifically compare inspection time to the average distractor for each modifier type.

Fig. 5.7 plots these inspections times for time window 2, but also the preceding and following time windows (windows 1 and 5; notice that windows 3 and 4 are empty when there are no post-modifiers). The graph shows that in time window 1, distractor buttons receive equal amounts of attention in each pre-modifier condition. It further shows that the actual mentioning of the pre-modifier in time window 2 results in a significant increase of distractor inspection time for the relative case (11.39%) over the absolute case (2.79%, Coeff. $= -.086$, SE = .030, t = 2.913, $p(\text{MCMC}) < .01$). Distractors are still looked at longer after the reference (time window 5) when the pre-modifier was a relative one, though this effect now seems to decline (Coeff. $= -.034$, SE = .018, t $= -1.928$, $p(\text{MCMC}) = .054$). This suggests that listeners rapidly seek to map spoken instructions onto the visual environment differentially, depending on the absolute versus relative nature of the expression.

5.4.3 Referential understanding

Our second hypothesis was that monitoring listener gaze can improve the referential success of the speaker's utterances. We explore this hypothesis by comparing the performance of the eye-tracking-based generation system—which actively monitors listener gaze—against that of the other two systems, according to several metrics of task performance.

On evaluating subjective responses that participants gave in post-task questionnaires, we did not find any significant preferences for a particular system.

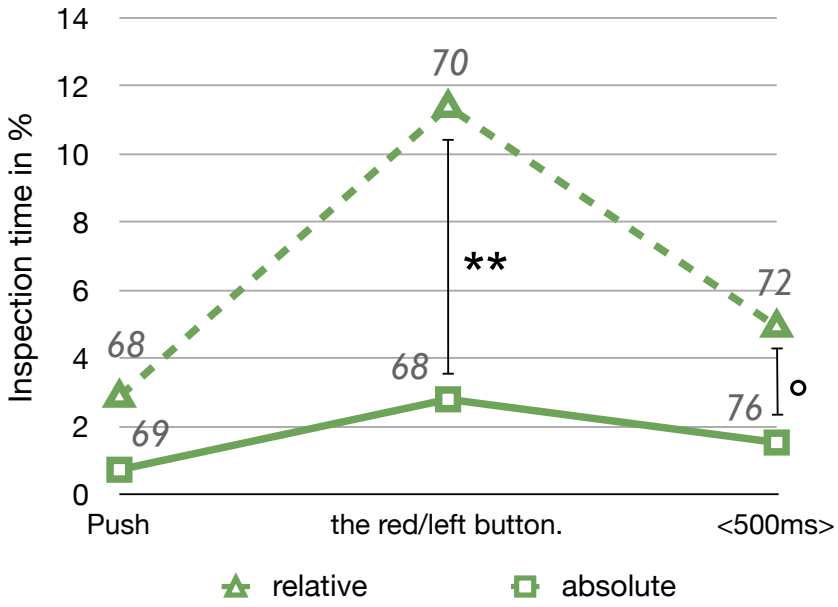


Figure 5.7: Average inspection time (% of time window) spent on distractor buttons, divided according to the type of adjectival pre-modifier used in the noun phrase (absolute or relative). Grey numbers represent the number of scene fragments falling into each time window. Differences in inspection times during processing of relative adjectives as compared to absolute adjectives are statistically significant at $^{**}p(\text{MCMC}) < .01$, $^{\circ}p(\text{MCMC}) < .1$.

Roughly the same number of participants chose each of the systems on questions such as "Which system did you prefer?" When asked for differences between the systems in free-form questions, no participant mentioned the systems' reaction to their eye gaze—though some did notice the feedback or lack thereof. We take this to mean that the participants did not realize that they were being eye-tracked, or at least that they did not consciously adjust their behavior based on any such belief. Below, we therefore focus on objective metrics that do not depend on participants' judgments.

Confusion rates

One measure of the ease of referential understanding is the frequency with which participants requested help, prompting systems to generate a new instruction. Thus, measuring the occurrence of 'H' keystrokes is an indication of the amount

of confusion that a participant experienced—or, conversely, the clarity and effectiveness of a system's instructions.

The average number of 'H' keystrokes per interaction is displayed in Fig. 5.8. A model with a Poisson distribution fitted to the keystroke data per system shows significant differences between the eye-tracking-based and the no-feedback system (Coeff. = .987, SE = .231, Wald's Z = 4.27, $p < .001$), as well as between the eye-tracking-based and the movement-based system (Coeff. = .487, SE = .247, Wald's Z = 1.968, $p < .05$). In both cases, users of the eye-tracking-based system appear less confused.

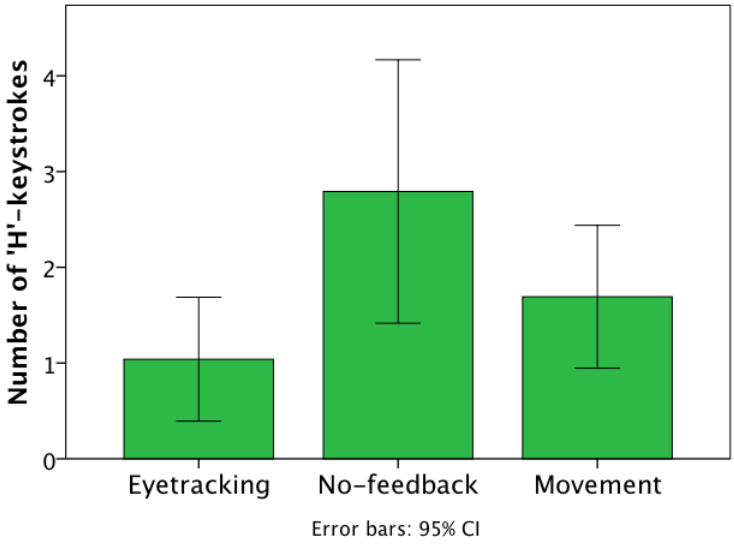


Figure 5.8: Average number of 'H' keystrokes per interaction, by system.

Referential success

An even more direct way to measure referential understanding is the ratio of system-generated references that participants were able to resolve correctly. We count a referring expression as *successful* if (a) the first button that the participant pressed after hearing the expression was the system's intended referent and (b) the participant did not request help between hearing the first-mention referring expression and pressing the correct button.

Trial	Success (%)	Feedback onset (ms)	Trial duration (ms)	Number of scenes
Eyetracking	91.52	-	7260	224
with feedback	92.19	1853	7569	205
positive	98.48	1937	5687	132
negative	84.93	1702	10973	73
without feedback	84.21	-	3928	19
No-feedback	82.88*	-	6601	222
Movement	88.75	-	7041	240
with feedback	94.53	2508***	7039	201
positive	99.40	2415 *	6201	168
negative	78.79	2981***	11309	33
without feedback	58.97°	-	7049*	39

Table 5.2: Mean referential success rates, feedback onset times and trial durations, broken down by presence and type of feedback. Differences to the eye-tracking system are significant at ***$p < .001$, **$p < .01$, *$p < .05$, °$p < .1$. The number of referential scenes falling under each category is provided in the last column.

The results of this evaluation are displayed in Table 5.2, under "Success". A logistic mixed-effects model fitted to the referential success data revealed a main effect of system ($\chi^2(2) = 6.693, p < .05$). Random effects included in this model are $participant$ and $target{-}button$; the latter corresponds to the traditional by-item variation and captures the heterogeneity of the rooms and scenes in which targets are situated. These random intercepts (but not the slopes) account for a significant part of the variation in the dependent variable $success$. Pairwise comparisons show that the eye-tracking-based system performs significantly better than the no-feedback system in terms of referential success (Coeff. = $-.905$, SE = .354, Wald's Z = $-2.55, p < .05$), while no significant difference is found overall between the eye-tracking-based and the movement-based system (Coeff. = $-.412$, SE = .371, Wald's Z = $-1.11, p = .267$).

We further sought to assess how referential success may have been influenced by the systems' feedback characteristics, as determined by the type of the first feedback they provided (positive versus negative) and its temporal variation. Table 5.2 thus also shows means and significant differences between the eye-tracking-based and the other systems with respect to these measures. Specifically, the second column displays the average *onset of feedback* relative to the offset of a referring expression, i.e., how soon after uttering an instruction such

as “Press the button to the left of the picture” the system provided feedback. In the third column, the average *trial duration* (i.e., the average duration of referential scenes) reflects the efficiency of the interactions and how this relates to the feedback given (or not) by each system. The last column additionally provides the number of scenes that fall into each of these categories, giving an overview of how frequently each feedback type has occurred.

In this table we observe that the no-feedback system has a numerically shorter trial duration (though also a lower success rate) than the eye-tracking-based system. That is, users of this system tended to press a button faster but more often pressed a wrong one. Regarding the movement-based system, the comparison is more complex: In trials in which feedback was provided, the movement-based system generated its feedback significantly later than the eye-tracking-based system, regardless of the feedback type (this model includes random intercepts and slopes for $participant$ and $target{-}button$: Coeff. = 829.9, SE = 349.0, t = 2.378, $p(\text{MCMC}) < .001$, with p values calculated for the model without random slopes). Negative feedback—a system’s means of preventing misunderstandings—was provided only in 13.75% of the trials with the movement-based system, in contrast to 32.59% of the trials with the eye-tracking-based system. When positive feedback was provided, its delayed onset for the movement-based system as compared with the eye-tracking-based system is associated with a prolonged trial duration, even though this difference is not significant when including random slopes for $participant$ and $target{-}button$ in the model (Coeff. = 410.6, SE = 272.0, t = 1.51). This numerical difference emphasizes the potential influence of early-timed feedback on referential performance.

On the other hand, looking at scenes in which the two feedback-based systems failed to provide feedback, a marginally significantly worse success rate can be observed for the movement-based system (Coeff. = -2.375, SE = 1.356, Wald’s Z = $-1.752, p = .080$). In addition, such trials lasted significantly longer for that system (Coeff. = -2813, SE = 1109, t = 2.537, $p(\text{MCMC}) < .05$). Since the movement-based system can only provide feedback when no distractors remain visible—which may not be possible in scenes where targets are closely surrounded by distractors—, a higher complexity of the scenes without feedback may be a cause for the lower performance of this system in these scenes.

Further task performance metrics

Finally, we considered a number of additional objective metrics, including the total number of user actions (i.e., presses of buttons and the final pickup of the trophy), what distance a user traveled, how long an interaction lasted, and how long a user was idle (i.e., did not move or turn in the virtual environment) during an interaction. While these metrics provide only partially significant results, they contribute to a more complete picture of how the eye-tracking-based feedback affects task performance.

Because the three virtual environments were of different complexity, we normalized the number of actions, distance and duration by dividing a value for a given interaction by the minimum value for all interactions in the same virtual environment. The idleness metric was normalized according to the total (raw) duration of a given interaction. The resulting metrics are shown in Table 5.3. We find that users of the eye-tracking-based system performed significantly fewer actions than those of the no-feedback system (Coeff. = $.160$, SE = .067, t = $2.42, p(\text{MCMC}) < .05$); there are also trends that users of that system traveled the shortest distance, needed the least overall time, and spent the least time idle.

The only measure deviating from this trend is movement speed, which is given in the last column of Table 5.3. For all successful referential scenes, we computed this measure by dividing the overall distance (in GIVE distance units) between the target and the user's location at the onset of the scene by the scene's duration (in seconds); it thus corresponds to the average speed at which users were able to appropriately react to the systems' referring expressions. A main effect of movement speed ($\chi^2(2) = 6.45, p < .05$) shows that participants moved significantly more slowly when getting eye-tracking-based feedback than when getting no feedback at all (Coeff. = $.033$, SE = .014, t = $2.348, p(\text{MCMC}) < .05$).

System	Number of actions (norm.)	Distance (norm.)	Duration (norm.)	Idleness (norm.)	Movement speed (GIVE units / sec.)
Eyetracking	1.07	1.23	1.50	0.687	0.491
No-feedback	1.24*	1.27	1.63	0.689	0.521*
Movement	1.13	1.24	1.53	0.692	0.506

Table 5.3: Mean values of additional task performance metrics. Differences to the eye-tracking-based system are significant at *$p < .05$.

5.5 Discussion

The aim of this study was to investigate the hypotheses (i) that listener gaze can offer a reliable index of online referential understanding, within a single trial, and in dynamic, interactive settings, and (ii) that such listener gaze can be useful for the speaker as early evidence that they may not have been understood, enabling them to respond proactively and improve their chances of successful communication. Our findings provide strong support for both hypotheses and have a number of implications for future cognitive and computational research.

5.5.1 Key findings

In particular, concerning (i), we found that listeners in our 3D environments inspected the object to which they resolved a given referring expression significantly more often than the average distractor object, immediately after the offset of the expression. This confirms that listener gaze is rapidly directed towards understood referents not only in static 2D visual scenes (as has been shown by earlier studies in the visual world paradigm; e.g., Allopenna et al. (1998)), but also in more dynamic and task-oriented settings. Our approach demonstrates the feasibility of investigating speech-mediated gaze behavior in complex and realistic task-oriented settings, and suggests that the rapid and robust mapping of speech to the visual environment via gaze is not limited to the highly simplified and static setting of traditional visual world studies.

Examining in more detail a subset of referring expressions that get disambiguated upon head offset, we found further evidence for incremental language processing: While they were processing relative terms (whose interpretation inherently involves a relation between the referent and other objects in the scene), listeners directed their gaze to distractor objects significantly longer than when processing absolute terms. This pattern appears to be in line with earlier findings of incremental semantic interpretation in the visual world. For instance, Sedivy et al. (1999) found that, while processing scalar relative adjectives such as “tall”, listeners may start fixating the referent considerably later than otherwise expected, and spend more time fixating competitor objects. However, the *typicality* of the referent (i.e., how representative it is of a prototypical object of its class) had an impact on listener gaze in their study. In contrast, we examined context-dependent spatial adjectives (“left” and “right”), to which no typicality effects apply; their interpretation is based exclusively on a listener’s spatio-visual context and the

noun phrase that is being modified. Though spatial relations such as "above" have been investigated before (e.g., Burigo and Knoeferle (2011)), we are not aware of earlier research that has provided insights into the visual processing of non-scalar relative adjectives.

We were also able to confirm hypothesis (ii): Our eye-tracking-based language generation system outperformed the two baselines in terms of avoiding listener confusion; additionally, it improved referential success in comparison with the no-feedback system and it enabled earlier feedback than the movement-based system. The system was able to generate feedback and, in particular, negative feedback—a means of repairing misunderstandings—more often than the alternative feedback-enabled system. Trials without feedback were significantly shorter and had higher success rates than in the movement-based system, presumably due to the fact that that system's feedback mechanism was not triggered in some of the most challenging referential scenes. Finally, users interacting with the eye-tracking-based system were able to complete their tasks with significantly fewer actions than those of the no-feedback system.

The performance improvement that the system derives from generating gaze-based feedback also speaks indirectly to (i). Whenever our speaker model gives feedback based on an incorrect judgment about how the listener resolved a reference, this feedback will likely be misleading and cause the listener to make more mistakes than if they had received no feedback at all. The high referential success rate of the eye-tracking-based system indicates that this has not occurred frequently in the interactions we collected. This implies that listener gaze can serve as a reliable index of understanding in each individual referential scene, i.e. on a trial-by-trial basis, and not only when averaged over numerous trials.

While our findings do not prove that human speakers will exploit listener gaze to provide feedback, they do demonstrate that it is possible to construct a computational model of the speaker that effectively does so in real time, and that this has a number of clear benefits for the listener—and the interaction overall—, as summarized above. Our eye-tracking-based system is, to our knowledge, the first language generation system that was designed to this end. As the comparison with the movement-based system shows, such early feedback generation could not easily be achieved without using eye-tracking; especially at such low implementation cost. The high frequency of the eye-tracking-based feedback further suggests that, as a ubiquitous and direct index of referential understanding, listener's gaze may indeed be more readily available and easier to interpret than other non-verbal (and possibly verbal) cues.

5.5.2 Future directions

By choosing a more complex and realistic design than traditional visual-world experimental settings, we were introduced to new challenges in the analysis of listeners' eye movements. In our dynamic scenes, referents and possible distractors constantly shifted on the screen as participants navigated through the virtual environments, introducing noise to the eye-tracking signal. We excluded from analysis a portion of the interactions in which the eye-tracking data were found to be poor (19%), but the accuracy of the recorded inspections in the remaining data could likely still be improved by a more refined method of mapping screen positions onto the visual environment. In contrast to simple visual scenes where specific objects have been placed as e.g. referents and distractors throughout trials, in our scenes the number and type of objects on display often varied even over the course of a single referring expression. To still be able to compare inspection times of relevant objects, we addressed this variability by averaging over objects that have been visually available within a given time window. This step could perhaps be further refined, e.g. by introducing a threshold on how long an object has been visually available before it can be considered as a candidate for inspection.

At the same time, the interactive, non-deterministic nature of our scenes made the use of pre-specified linguistic stimuli (as typically employed in visual-world studies) impossible. Though our referring expression generation model offered a sufficient number of stimuli for a detailed analysis with respect to different types of pre-modifiers, exercising more systematic control over its output could help us obtain a more balanced collection of listener behavior for specific formulations and visual contexts. While we focused on the absolute versus relative nature of an object's attributes here, others such as viewer-centered attributes (e.g., "to your left") would also be of interest. An examination of anchor processing (e.g., in terms of whether the anchor corresponds to a distractor, an immovable landmark or a movable landmark) would contribute towards a more complete theory of referential processing. Another avenue for future research would be to examine the reliability of listener gaze as a signal of situated language understanding beyond referring expressions (e.g., in response to navigational instructions).

On the other hand, linguistic stimuli alone could not possibly account for a listener's eye movements in full, especially when situated in a complex task-based environment. Together with the linguistic aspects, other aspects of the communicative setting (e.g., the spatio-visual context, the nature of the task and the history of the interaction) are also likely to play a crucial role. Even the process of eye-tracking itself—especially if a participant is aware that their conversational

partner is practicing it—might influence their eye movements; all these are, we believe, interesting directions for future cognitive research.

In terms of system performance, one finding that seemed to go against the general trend was that users of the eye-tracking system moved significantly more slowly on their way to a target than users of the no-feedback system. We see two possible explanations for this. First, it may be that users needed some time to listen to the feedback and process it, or were encouraged by it to look at more objects. A second explanation is that this may not be indicative of a difference in the quality of the systems' behavior, but a difference in the populations over which the mean speed was computed: The speed was only averaged over scenes in which the users resolved a referring expression correctly. Since the eye-tracking system achieved success in many cases in which the no-feedback system did not, these were presumably complex scenes in which the user had to work harder to infer the correct referent. Though the eye-tracking system performs better than the no-feedback system in terms of a wide range of metrics, we currently cannot rule out that this particular metric might reflect a limitation of the system. This issue bears more careful analysis.

On the whole, we see far-reaching implications for computational research in situated interaction. Eyetracking has very recently started becoming mainstream technology in intelligent devices (see, e.g., Park et al. (2013); Horning et al. (2013)); if a system can employ it to predict its communicative success or failure, it might be able to enhance the interaction in new ways. The proof-of-concept system we presented here made simple use of this technology and only generated relatively unspecific feedback (e.g., "No, not that one"), even in the presence of multiple distractors. Its performance could likely be improved by providing more specific and tailored feedback (e.g., "No, the BLUE button"). Beyond the linguistic content, further improvements might be possible if the timing of the feedback was sensitive to the situational context and the individual needs of different users. The findings of this study could directly support a rapid feedback strategy that is sensitive to the semantics of particular expressions and the history of eye movements. For instance, if a system is aware that users tend to direct their eyes to distractor objects while hearing "left" but not while hearing "red", it may be capable of diagnosing misunderstandings earlier—even before finishing its utterance. This would eventually enable a system to react to misunderstandings in a more effective fashion, perhaps even eliminating the need for corrective feedback altogether by modifying its original utterance online.

5.6 Conclusion

In this chapter, we have investigated the overarching hypothesis that referential performance in task-oriented situated interaction should be enhanced when the speaker is able to monitor listener gaze, both to assess whether they have been understood and to offer appropriate feedback. We found that listeners in dynamic scenes reliably fixate what they interpret as the referent of a particular referring expression, in a way that can be distinguished from general inspection of the scene. A computational model of the speaker that exploits such listener gaze in order to provide early feedback resulted in considerably improved task performance, as revealed by a range of metrics. More broadly, and perhaps importantly, with this work we have opened the door to new methods both for building cognitive models of situated communication and for contributing to the development of more effective human-computer interaction. In particular, we have shown how a computational model that instantiates a theoretical claim (or several variants thereof) can become part of the experimental evaluation itself. When investigating models of situated language interaction, which fundamentally calls for a dynamic and non-deterministic—yet sufficiently controlled—setting, such methods become increasingly relevant.

Chapter 6

Conclusion

In this final chapter, we briefly summarize our approach and main findings, and we identify a few directions for future work.

6.1 Summary

In this thesis, we have developed a planning-based approach to the interactive generation of natural language in situated context, addressing the challenges of controlling, adapting to, and monitoring the context. Our approach enables a computer system to generate effective discourse in real time and provide assistance to a user in identifying given task-related entities in their surroundings, as e.g. in (1):

(1) a. SYSTEM: "Walk three steps forward and then turn right."
USER: *(the user walks and turns)*

b. SYSTEM: "OK. You're looking for the upper silver-colored lamp in front of you."
USER: *(the user is being distracted by another silver-colored lamp in front of them, which uses halogen)*

c. SYSTEM: "No, not that one!"
USER: *(the user's eyes move upwards to the other silver-colored lamp)*
SYSTEM: "Yes, that one!"
USER: *(the user finds what they were looking for, successfully completing their task)*

To develop our approach, we started with the observation that goal-directed language production as in (1) constitutes a form of action, and can be planned in the same way as the physical actions of a robot can. We surveyed over three decades of research that has drawn natural language generation (Reiter and Dale, 2000) and automated planning (Ghallab et al., 2004) together, and identified the strengths and weaknesses of such approaches. This led us to conclude that modern planning methods have reached levels of efficiency that may support real-time communicative planning at a high degree of linguistic analysis. Though many problems remain unaddressed, we identified possible ways of advancing this line of work. Deterministic planning—interleaved with plan execution monitoring and re-planning—emerged as a potentially promising method in interactive settings; the combination of symbolic and statistical techniques is arguably a fruitful way of handling the uncertainty that natural-language communication involves.

We then built on the CRISP sentence planning approach (Koller and Stone, 2007) to develop an interactive generation system for situated discourse. To this end, we extended CRISP's LTAG-based (Joshi and Schabes, 1997) syntax-semantics interface to integrate non-linguistic information together with information of linguistic nature. We assumed, for the purposes of planning, that perlocutionary effects of communicative acts will come true as intended, and used an off-the-shelf classical planner (Koller and Hoffmann, 2010) to generate discourses efficiently. From this modeling, we gained two main advantages. First, we were able to generate context-dependent referring expressions by keeping track of both linguistically and non-linguistically introduced distractors during a unified generation process. Second, we developed the first, to our knowledge, full-fledged generation system that can deliberately manipulate the non-linguistic context of communicative scenes in order to make it more favorable for subsequent references to task-related objects. This distributes a user's cognitive load of interpreting a reference over multiple utterances rather than one long referring expression.

The above approach can control aspects of the situated context but cannot make linguistic choices that are tailored to such aspects. We thus utilized a human-human interaction corpus (Gargett et al., 2010) to learn how to make such choices. We assessed human-produced references for their helpfulness to hearers in situated context and learned to distinguish between less helpful and more helpful references. Unlike traditional approaches, this model does not mimic human choices blindly—it only does so when there is indication that these choices are effective. We then integrated the learned model into a planning-based generation system, using the deterministic formalism of metric planning (Fox and Long, 2003), and associating attributes of a referent with an estimation of how preferable they are

in the given context. The resulting system, which combines symbolic and statistical reasoning, goes beyond the state of the art by tackling the problem of making non-trivial linguistic choices in a complex and realistic setting.

Our optimistic approach to estimating the perlocutionary effects of utterances needed to be complimented with an adequate mechanism for monitoring the situated context and reacting to unexpected states. We demonstrated that monitoring a user's gaze with remote eye-tracking technology can offer such a mechanism. We showed that the user's gaze provides a reliable index of online referential understanding even in complex and dynamic situated environments. By doing so, we presented the first—to our knowledge—language generation system that monitors fine-grained user gaze cues in order to enhance its referential effectiveness. We found that exploiting gaze enables the generation of appropriate feedback rapidly and on a per-utterance basis, which results in considerably improved task performance as revealed by a range of metrics.

Overall, we believe that our work has implications for future computational and empirical research in situated communication. We chose to develop and evaluate our approach in virtual environments (Koller et al., 2010b; Striegnitz et al., 2011), which provide rich and dynamic situated context while making the influences of different aspects of the context measurable. The setting therefore retains much of the complexity of real physical scenes, while still being sufficiently controlled. By operating in such context, our language generation model may provide a methodological framework for future studies in referential effectiveness or multi-modal human-computer interaction.

Finally, our deterministic approach to dealing with the uncertainties involved in natural language communication allowed us to retain efficiency while solving non-trivial planning problems. The integration of statistical reasoning about the effectiveness of referential choices into this approach, and its complementation with reliable monitoring of the user's understanding, offered us a way of addressing the challenges of situated language generation. It is worth noting that deterministic planning is also being actively explored as a viable solution to real-life problems in areas outside linguistics. For instance, Nebel et al. (2013) apply a deterministic planning approach to the problem of autonomous robot control in the household domain, which features uncertainty due to incomplete information and multiple action outcomes. By simplifying a non-deterministic problem to a deterministic problem in a continual planning loop, we obtain a scalable solution that may bring us closer to effective real-world systems—be they robotic, communicative, or both.

6.2 Outlook

In this work, we have identified three challenges in generating effective discourse in situated context: controlling, adapting to, and monitoring the context. Each of these challenges suggests an avenue for future exploration, and so does the question of how we can scale up to real-life settings. Beyond the directions for future research that we discussed in the previous chapters, we list some further ideas below.

6.2.1 Controlling the situated context

We presented a strategy for controlling the non-linguistic, and in particular the visual, context of a reference. Controlling other aspects of the context may also facilitate a user's processing. For instance, as an alternative to the discourse "Turn right. You're looking for the red lamp", it might be advantageous, in certain contexts, to generate the discourse "You're looking for a red lamp. It's to your right"; this discourse manipulates the linguistic context and makes use of the givenness hierarchy (Gundel et al., 1993). Such "referring in installments" has been observed in the GIVE setting (Striegnitz et al., 2012), and it would be interesting to investigate the effectiveness of this strategy as compared with other context-manipulating strategies. A related question is whether the use of "incrementally informative" (Fernández, 2013) referring expressions, which allow the hearer to rule out distractors online as they process the expression, may also result in lower cognitive load. On the other hand, if an object is already salient, e.g. because the user has interacted with it at an earlier point, referring in installments or incrementally might be ill-suited. An open problem is to model the set of domain entities that the user is attending to and decide on the optimal strategy accordingly (Poesio, 1993). While planning context-manipulating discourse, an important question raised by Cawsey (1991) is how to update the attentional state of the user in accordance with the generated discourse, and modify the remaining discourse "as the perceived context changes".

6.2.2 Adapting to the situated context

In optimizing the effectiveness of referential choices in situated context, we focused on the problem of selecting appropriate attributes, given contextual aspects that may influence the user's conceptual accessibility of a referent. As empirical

research increasingly sheds light on human referential production and processing mechanisms, other factors to take into account are for instance the codability of the referent's attributes (Viethen et al., 2012) and the typicality of a particular property (Mitchell et al., 2013a). Adapting to user profiles (e.g., Janarthanam and Lemon (2010)) is another desirable capability. A range of referential choices, such as the use of negation, plurals, vagueness, and different types of spatial relations (e.g., "near", "across") remains to be assessed for its effectiveness. Beyond reference, our full-fledged generation approach could be extended to optimize linguistic choices of different nature jointly. For instance, reference generation and generation of navigational instructions could be optimized in combination (Eberle, 2013). One obstacle in modeling effective discourse is that it can be challenging to assess the effectiveness of human-produced utterances, since this involves fine-grained observations about the hearer's processing. A direction worth exploring to address this problem—at least for the generation of spatial language—might be the consideration of spatial ability tests, as the performance of speakers in such tests has been found in some cases to correlate with the effectiveness of their direction-giving strategies (Gargett et al., 2010; Brennan et al., 2013).

6.2.3 Monitoring the situated context

While monitoring the situated context following a system's utterance, we investigated the usefulness of the user's gaze as a ubiquitous and readily available cue. Combining gaze information with other sources of evidence of a user's understanding may be beneficial. Modalities such as gesture, posture, and head nods all have the potential to contribute to a system's overall belief state tracking (Williams, 2012). This belief state need not be restricted to referential understanding, on which we focused here, but may cover other aspects such as the understanding of navigational instructions. In further exploring the usefulness of gaze with this respect, experimental findings suggest that people's eye movements might even be indicative of their prediction of others' actions, when they observe them performing tasks (Flanagan and Johansson, 2003). Based on such monitoring, it is important to identify the sources of misunderstanding or non-understanding and devise appropriate recovery strategies (Marge and Rudnicky, 2011). A question we have not directly addressed is the one of deciding when and how to switch between planning, plan execution monitoring, and re-planning during this process. For instance, we have only investigated post-utterance feedback mechanisms here, but, in the face of trouble, mid-utterance feedback mechanisms (Skantze and Schlangen, 2009) might be more effective.

6.2.4 Scaling up

Finally, though our approach has achieved real-time performance and has been evaluated favorably in terms of its effectiveness, implementing this approach in a real-life setting, as the one envisioned in the beginning of Chapter 1, would certainly pose new challenges. In scaling up to more expressive grammars, it remains to be seen how one could appropriately represent "the compositional structure and meaning of utterances in an action formalism" (Stone, to appear) and preserve the syntax-semantics interface (Gardent et al., 2011). In larger and even more complex settings, real-time generation may be difficult to sustain. It might be useful to try other planning formalisms such as hierarchical planning (Nau et al., 2003) to manage this complexity. Eye-tracking in the real world is also a project for future research; great advancements in wearable eye-tracking technology over the last few years may aid in this investigation (e.g., Foulsham and Kingstone (2012); Macdonald and Tatler (2013); Horning et al. (2013)). Ultimately, the user's perception of a conversational system may be influenced by other factors in addition to the effectiveness of its utterances—e.g., its likeability (Hone and Graham, 2000). Determining a system's overall optimal language-mediated behavior—given particular tasks, users, and situations—is, we believe, an exciting avenue for future work.

Bibliography

Alexandre Albore, Héctor Palacios, and Héctor Geffner. A translation-based approach to contingent planning. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, Pasadena, CA, 2009.

James F. Allen and C. Raymond Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15(3):143–178, 1980.

James F. Allen, George Ferguson, and Amanda Stent. An architecture for more realistic conversational systems. In *Proceedings of the 6th International Conference on Intelligent User Interfaces*, Santa Fe, NM, 2001.

Paul Allopenna, James Magnuson, and Michael Tanenhaus. Tracking the time course of spoken word recognition using eye movements: Evidence for continuous mapping models. *Journal of Memory and Language*, 38:419–439, 1998.

Gerry T.M. Altmann and Yuki Kamide. Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, 73(3):247–264, 1999.

Richard Andersson, Fernanda Ferreira, and John M. Henderson. I see what you're saying: The integration of complex speech and scenes during language comprehension. *Acta Psychologica*, 137(2):208–216, 2011. Special Issue: Visual search and visual world: Interactions among visual attention, language, and working memory.

Douglas E. Appelt. Planning English referring expressions. *Artificial Intelligence*, 26(1):1–33, 1985a.

Douglas E. Appelt. *Planning English sentences*. Cambridge University Press, Cambridge, England, 1985b.

Carlos Areces, Alexander Koller, and Kristina Striegnitz. Referring expressions as formulas of description logic. In *Proceedings of the 5th International Natural Language Generation Conference*, Salt Fork, OH, 2008.

Anja Arts, Alfons Maes, Leo Noordman, and Carel Jansen. Overspecification facilitates object identification. *Journal of Pragmatics*, 43(1):361–374, 2011.

John L. Austin. *How to do things with words*. Oxford University Press, 1962.

R. Harald Baayen, Douglas J. Davidson, and Douglas M. Bates. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59:390–412, 2008.

Daniel Bauer and Alexander Koller. Sentence generation as planning with probabilistic LTAG. In *Proceedings of the 10th International Workshop on Tree Adjoining Grammar and Related Formalisms*, New Haven, CT, 2010.

Luciana Benotti. Clarification potential of instructions. In *Proceedings of the 10th Annual SIGDIAL Meeting on Discourse and Dialogue*, London, UK, 2009.

Luciana Benotti and Patrick Blackburn. Classical planning and causal implicatures. In M. Beigl, H. Christiansen, T. Roth-Berghofer, A. Kofod-Petersen, K. Coventry, and H. Schmidtke, editors, *Modeling and Using Context*, volume 6967 of *LNCS*, pages 26–39. Springer, 2011.

Dan Bohus and Alexander I. Rudnicky. Integrating multiple knowledge sources for utterance-level confidence annotation in the CMU Communicator spoken dialog system. Technical report, Carnegie Mellon University, 2002.

Johan Boye, Morgan Fredriksson, Jana Götze, Joakim Gustafson, and Jürgen Königsmann. Walk this way: Spatial grounding for city exploration. *Proceedings of the 4th International Workshop on Spoken Dialog Systems*, 2012.

Michael Bratman. *Intention, plans, and practical reason*. Harvard University Press, 1987.

Susan E. Brennan and Joy E. Hanna. Partner-specific adaptation in dialog. *Topics in Cognitive Science*, 1(2):274–291, 2009.

Susan E. Brennan, Katharina S. Schuhmann, and Karla M. Batres. Collaboratively setting perspectives and referring to locations across multiple contexts. In *Proceedings of the PRE-CogSci 2013 Workshop on the Production of Referring Expressions*, Berlin, Germany, 2013.

Michael Brenner. Creating dynamic story plots with continual multiagent planning. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, Atlanta, GA, 2010.

Michael Brenner and Ivana Kruijff-Korbayová. A continual multiagent planning approach to situated dialogue. In *Proceedings of the 12th Workshop on the Semantics and Pragmatics of Dialogue*, London, UK, 2008.

Michael Brenner and Bernhard Nebel. Continual planning and acting in dynamic multiagent environments. *Autonomous Agents and Multi-Agent Systems*, 19(3): 297–331, 2009.

Gordon Briggs and Matthias Scheutz. A hybrid architectural approach to understanding and appropriately generating indirect speech acts. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, Bellevue, WA, 2013.

Sarah Brown-Schmidt. Partner-specific interpretation of maintained referential precedents during interactive dialog. *Journal of Memory and Language*, 61(2): 171–190, 2009.

Harry Bunt. Context and dialogue control. *Think Quarterly*, 3(1):19–31, 1994.

Michele Burigo and Pia Knoeferle. Visual attention during spatial language comprehension: Is a referential linking hypothesis enough? In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, Boston, MA, 2011.

Okko Buß and David Schlangen. Modelling sub-utterance phenomena in spoken dialogue systems. In *Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue*, pages 33–41. 2010.

Donna Byron. Understanding referring expressions in situated language: Some challenges for real-world agents. In *Proceedings of the 1st International Workshop on Language Understanding and Agents for Real World Interaction*, Sapporo, Japan, 2003.

Alison Cawsey. Generating interactive explanations. In *Proceedings of the 9th AAAI National Conference on Artificial Intelligence*, Anaheim, CA, 1991.

David Chen and Igor Karpov. The GIVE-1 Austin system. In *Proceedings of the Generation Challenges Session at the 12th European Workshop on Natural Language Generation*, Athens, Greece, 2009.

Jennifer Chu-Carroll and Sandra Carberry. Conflict detection and resolution in collaborative planning. In M. Wooldridge, J. Müller, and M. Tambe, editors, *Intelligent Agents II. Agent Theories, Architectures, and Languages*, volume 1037 of *LNCS*, pages 111–126. Springer, 1996.

Herbert H. Clark and Meredyth A. Krych. Speaking while monitoring addressees for understanding. *Journal of Memory and Language*, 50(1):62–81, 2004.

Herbert H. Clark and Edward F. Schaefer. Contributing to discourse. *Cognitive Science*, 13(2):259–294, 1989.

Herbert H. Clark and Deanna Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22(1):1–39, 1986.

Philip R. Cohen and Hector J. Levesque. Rational interaction as the basis for communication. In P.R. Cohen, J. Morgan, and M.E. Pollack, editors, *Intentions in Communication*, Bradford books, pages 221–255. MIT Press, 1990.

Philip R. Cohen and C. Raymond Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3:177–212, 1979.

Amanda Coles, Andrew Coles, Angel García Olaya, Sergio Jiménez, Carlos Linares López, Scott Sanner, and Sungwook Yoon. A survey of the Seventh International Planning Competition. *AI Magazine*, 33(1):83–88, 2012.

Robert Dale and Ehud Reiter. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263, 1995.

Laurence Danlos. Conceptual and linguistic decisions in generation. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*, Stanford, CA, 1984.

Sammy Davis-Mendelow, Jorge Baier, and Sheila McIlraith. Assumption-based planning: Generating plans and explanations under incomplete knowledge. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, Bellevue, WA, 2013.

Alexandre Denis. Generating referring expressions with reference domain theory. In *Proceedings of the 6th International Natural Language Generation Conference*, Dublin, Ireland, 2010.

Marie E. desJardins, Edmund H. Durfee, Charles L. Ortiz, and Michael J. Wolverton. A survey of research in distributed, continual planning. *AI Magazine*, 20 (4):13–22, 1999.

Nina Dethlefs and Heriberto Cuayáhuitl. Hierarchical reinforcement learning for adaptive text generation. In *Proceedings of the 6th International Natural Language Generation Conference*, Dublin, Ireland, 2010.

Nina Dethlefs and Heriberto Cuayáhuitl. Combining hierarchical reinforcement learning and Bayesian networks for natural language generation in situated dialogue. In *Proceedings of the 13th European Workshop on Natural Language Generation*, Nancy, France, 2011.

Nina Dethlefs, Heriberto Cuayáhuitl, and Jette Viethen. Optimising natural language generation decision making for situated dialogue. In *Proceedings of the 12th Annual SIGdial Meeting on Discourse and Dialogue*, Portland, OR, 2011.

David DeVault, Kenji Sagae, and David Traum. Incremental interpretation and prediction of utterance meaning for interactive dialogue. *Dialogue & Discourse*, 2(1):143–170, 2011.

Lee Raymond Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

Christian Dornhege, Patrick Eyerich, Thomas Keller, Sebastian Trüg, Michael Brenner, and Bernhard Nebel. Semantic attachments for domain-independent planning systems. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling*, Thessaloniki, Greece, 2009.

Markus Dräger and Alexander Koller. Generation of landmark-based navigation instructions from open-source data. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Avignon, France, 2012.

Kathleen M. Eberhard, Michael J. Spivey-Knowlton, Julie C. Sedivy, and Michael K. Tanenhaus. Eye movements as a window into real-time spoken language comprehension in natural contexts. *Journal of Psycholinguistic Research*, 24(6):409–436, 1995.

Kira Eberle. Joint optimization of generating route instructions and referring expressions. Master's thesis, University of Potsdam, 2013.

Paul E. Engelhardt, S. Baris Demiral, and Fernanda Ferreira. Over-specified referring expressions impair comprehension: An ERP study. *Brain and Cognition*, 77(2):304–314, 2011.

Raquel Fernández. Rethinking overspecification in terms of incremental processing. In *Proceedings of the PRE-CogSci 2013 Workshop on the Production of Referring Expressions*, Berlin, Germany, 2013.

Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

J. Randall Flanagan and Roland S. Johansson. Action plans used in action observation. *Nature*, 424(6950):769–771, 2003.

Mary Ellen Foster. Enhancing human-computer interaction with embodied conversational agents. In C. Stephanidis, editor, *Universal Access in Human-Computer Interaction. Ambient Interaction*, volume 4555 of *LNCS*, pages 828–837. Springer, 2007.

Tom Foulsham and Alan Kingstone. Goal-driven and bottom-up gaze in an active real-world search task. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, Santa Barbara, CA, 2012.

Maria Fox and Derek Long. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 20: 61–124, 2003.

Kumiko Fukumura, Roger P. G. van Gompel, and Martin J. Pickering. The use of visual context during the production of referring expressions. *The Quarterly Journal of Experimental Psychology*, 63(9):1700–1715, 2010.

Claire Gardent, Benjamin Gottesman, and Laura Perez-Beltrachini. Using regular tree grammars to enhance sentence realisation. *Natural Language Engineering*, 17(2):185–201, 2011.

Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. The GIVE-2 Corpus of Giving Instructions in Virtual Environments. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valletta, Malta, 2010.

Konstantina Garoufi. Position paper at the YRRSDS 2012. In *Proceedings of the 8th Annual Young Researchers' Roundtable on Spoken Dialogue Systems*, Seoul, South Korea, 2012.

Konstantina Garoufi. Planning-based models of natural language generation. *Language and Linguistics Compass*, to appear.

Konstantina Garoufi and Alexander Koller. Controlling the spatio-visual context in situated natural language generation. In *Abstracts of the International Conference on Space in Language*, Pisa, Italy, 2009.

Konstantina Garoufi and Alexander Koller. Automated planning for situated natural language generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010.

Konstantina Garoufi and Alexander Koller. Combining symbolic and corpus-based approaches for the generation of successful referring expressions. In *Proceedings of the 13th European Workshop on Natural Language Generation*, Nancy, France, 2011a.

Konstantina Garoufi and Alexander Koller. The Potsdam NLG systems at the GIVE-2.5 Challenge. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, Nancy, France, 2011b.

Konstantina Garoufi and Alexander Koller. Generation of effective referring expressions in situated context. *Language and Cognitive Processes*, to appear.

Konstantina Garoufi, Maria Staudte, Alexander Koller, and Matthew Crocker. Exploiting listener gaze to improve situated communication in dynamic virtual environments. Under review for journal publication.

Albert Gatt, Ielka van der Sluis, and Kees van Deemter. Evaluating algorithms for the generation of referring expressions using a balanced corpus. In *Proceedings of the 11th European Workshop on Natural Language Generation*, Schloss Dagstuhl, Germany, 2007.

Albert Gatt, Anja Belz, and Eric Kow. The TUNA-REG Challenge 2009: Overview and evaluation results. In *Proceedings of the 12th European Workshop on Natural Language Generation*, Athens, Greece, 2009.

Albert Gatt, Emiel Krahmer, Roger P.G. van Gompel, and Kees van Deemter. Production of referring expressions: Preference trumps discrimination. In *Proceedings of the 35th Annual Conference of the Cognitive Science Society*, Berlin, Germany, 2013.

Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.

Dave Golland, Percy Liang, and Dan Klein. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, 2010.

Martijn Goudbeek and Emiel Krahmer. Preferences versus adaptation during referring expression generation. In *Proceedings of the ACL 2010 Conference Short Papers*, Uppsala, Sweden, 2010.

H. Paul Grice. Logic and conversation. In I. P. Cole and J. L. Morgan, editors, *Syntax and Semantics*, volume 3, pages 41–58. Academic Press, 1975.

Zenzi M. Griffin and Kathryn Bock. What the eyes say about speaking. *Psychological Science*, 11:274–279, 2000.

Barbara J. Grosz and Candace L. Sidner. Plans for discourse. In P.R. Cohen, J. Morgan, and M.E. Pollack, editors, *Intentions in Communication*, Bradford books, pages 417–444. MIT Press, 1990.

Markus Guhe. Generating referring expressions with a cognitive model. In *Proceedings of the Workshop on the Production of Referring Expressions: Bridging the gap between computational and empirical approaches to reference*, Amsterdam, The Netherlands, 2009.

Jeanette K. Gundel, Nancy Hedberg, and Ron Zacharski. Cognitive status and the form of referring expressions in discourse. *Language*, 69(2):274–307, 1993.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009.

Joy E. Hanna and Susan E. Brennan. Speakers' eye gaze disambiguates referring expressions early during face-to-face conversation. *Journal of Memory and Language*, 57:596–615, 2007.

Nick Hawes, Hendrik Zender, Kristoffer Sjöö, Michael Brenner, Geert-Jan M. Kruijff, and Patric Jensfelt. Planning and acting with an integrated sense of space. In *Proceedings of the 1st International Workshop on Hybrid Control of Autonomous Systems*, Pasadena, CA, 2009.

Sarah L. Haywood, Martin J. Pickering, and Holly P. Branigan. Do speakers avoid ambiguities during dialogue? *Psychological Science*, 16(5):362–366, 2005.

Peter A. Heeman and Graeme Hirst. Collaborating on referring expressions. *Computational Linguistics*, 21(3):351–382, 1995.

Raquel Hervas and Mark Finlayson. The prevalence of descriptive referring expressions in news and narrative. In *Proceedings of the ACL 2010 Conference Short Papers*, Uppsala, Sweden, 2010.

Graeme Hirst, Susan McRoy, Peter Heeman, Philip Edmonds, and Diane Horton. Repairing conversational misunderstandings and non-understandings. *Speech communication*, 15(3):213–229, 1994.

Jörg Hoffmann. Extending FF to numerical state variables. In *Proceedings of the 15th European Conference on Artificial Intelligence*, Lyon, France, 2002.

Jörg Hoffmann. Where "ignoring delete lists" works: Local search topology in planning benchmarks. *Journal of Articial Intelligence Research*, 24:685–758, 2005.

Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14: 253–302, 2001.

Christian Hofner and Günther Schmidt. Path planning and guidance techniques for an autonomous mobile cleaning robot. *Robotics and Autonomous Systems*, 14(2):199–212, 1995.

Kate S. Hone and Robert Graham. Towards a tool for the subjective assessment of speech system interfaces (SASSI). *Natural Language Engineering*, 6(3&4): 287–303, 2000.

Robert D. Horning, Thomas Ohnstein, and Bernard Fritz. Wearable eye tracking system, 2013. Patent EP 2 226 703 B1.

Eduard H. Hovy. Approaches to the planning of coherent text. In C. L. Paris, W. R. Swartout, and W. C. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pages 83–102. Kluwer Academic Publishers, 1991.

Eduard H. Hovy. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63(12):341–385, 1993.

Ryu Iida, Masaaki Yasuhara, and Takenobu Tokunaga. Multi-modal reference resolution in situated dialogue by integrating linguistic and extra-linguistic clues. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, Chiang Mai, Thailand, 2011.

T. Florian Jaeger. Categorical data analysis: Away from ANOVAs (transformation or not) and towards logit mixed models. *Journal of Memory and Language*, 59 (4):434–446, 2008. Special Issue: Emerging Data Analysis.

Srinivasan Janarthanam and Oliver Lemon. Learning lexical alignment policies for generating referring expressions for spoken dialogue systems. In *Proceedings of the 12th European Workshop on Natural Language Generation*, Athens, Greece, 2009.

Srinivasan Janarthanam and Oliver Lemon. Learning to adapt to unknown users: Referring expression generation in spoken dialogue systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010.

Srinivasan Janarthanam, Oliver Lemon, and Xingkun Liu. A web-based evaluation framework for spatial instruction-giving systems. In *Proceedings of the ACL 2012 System Demonstrations*, Jeju Island, Korea, 2012.

Kristiina Jokinen. Non-verbal signals for turn-taking and feedback. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valletta, Malta, 2010.

Pamela W. Jordan and Marilyn A. Walker. Learning content selection rules for generating object descriptions in dialogue. *Journal of Artificial Intelligence Research*, 24(1):157–194, 2005.

Aravind K. Joshi. The relevance of Tree Adjoining Grammar to generation. In G. Kempen, editor, *Natural Language Generation*, volume 135 of *NATO ASI Series*, pages 233–252. Springer, 1987.

Aravind K. Joshi and Yves Schabes. Tree-Adjoining Grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 69–123. Springer, 1997.

Daniel Jurafsky. Pragmatics and computational linguistics. In L. R. Horn and G. Ward, editors, *The Handbook of Pragmatics*, pages 578–604. Blackwell Publishing Ltd, 2004.

Daniel Jurafsky and James H. Martin. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, second edition, 2000.

Hans Kamp and Barbara Partee. Prototype theory and compositionality. *Cognition*, 57(2):129–191, 1995.

John D. Kelleher and Geert-Jan M. Kruijff. Incremental generation of spatial referring expressions in situated dialog. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, 2006.

Boaz Keysar, Shuhong Lin, and Dale J. Barr. Limits on theory of mind use in adults. *Cognition*, 89(1):25–41, 2003.

David Kirsh and Paul Maglio. On distinguishing epistemic from pragmatic action. *Cognitive science*, 18(4):513–549, 1994.

Pia Knoeferle, Matthew Crocker, Martin Pickering, and Christoph Scheepers. The influence of the immediate visual context on incremental thematic role-assignment: evidence from eye-movements in depicted events. *Cognition*, 95: 95–127, 2005.

Alexander Koller and Jörg Hoffmann. Waking up a sleeping rabbit: On natural-language sentence generation with FF. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling*, Toronto, Canada, 2010.

Alexander Koller and Ronald Petrick. Experiences with planning for natural language generation. *Computational Intelligence*, 27(1):23–40, 2011.

Alexander Koller and Matthew Stone. Sentence generation as a planning problem. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, 2007.

Alexander Koller, Andrew Gargett, and Konstantina Garoufi. A scalable model of planning perlocutionary acts. In P. Łupkowski and M. Purver, editors, *Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010*, pages 9–16. Polish Society for Cognitive Science, 2010a.

Alexander Koller, Kristina Striegnitz, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. The First Challenge on Generating Instructions in Virtual Environments. In E. Krahmer and M. Theune, editors,

Empirical Methods in Natural Language Generation, volume 5790 of *LNCS*, pages 337–361. Springer, 2010b.

Alexander Koller, Konstantina Garoufi, Maria Staudte, and Matthew Crocker. Enhancing referential success by tracking hearer gaze. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Seoul, South Korea, 2012.

Ruud Koolen, Albert Gatt, Martijn Goudbeek, and Emiel Krahmer. Factors causing overspecification in definite descriptions. *Journal of Pragmatics*, 43(13): 3231–3250, 2011.

Ruud Koolen, Martijn Goudbeek, and Emiel Krahmer. The effect of scene variation on the redundant use of color in definite reference. *Cognitive Science*, 37 (2):395–411, 2013.

Emiel Krahmer. What computational linguists can learn from psychologists (and vice versa). *Computational Linguistics*, 36(2):285–294, 2010.

Emiel Krahmer and Mariët Theune. Efficient context-sensitive generation of referring expressions. In K. van Deemter and R. Kibble, editors, *Information Sharing: Reference and Presupposition in Language Generation and Interpretation*, pages 223–264. CSLI Publications, 2002.

Emiel Krahmer, Sebastiaan van Erk, and André Verleg. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72, 2003.

Christian Kray, Christian Elting, Katri Laakso, and Volker Coors. Presenting route instructions on mobile devices. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, Miami, FL, 2003.

Gerasimos Lampouras and Ion Androutsopoulos. Using integer linear programming for content selection, lexicalization, and aggregation to produce compact texts from OWL ontologies. In *Proceedings of the 14th European Workshop on Natural Language Generation*, Sofia, Bulgaria, 2013.

Diane J. Litman and James F. Allen. A plan recognition model for subdialogues in conversations. *Cognitive Science*, 11(2):163–200, 1987.

Robert E. Longacre. *The grammar of discourse*. Plenum Press, New York, NY, 1983.

Ross G. Macdonald and Benjamin W. Tatler. Do as eye say: Gaze cueing and language in a real-world social interaction. *Journal of Vision*, 13(4):1–12, 2013.

Alfons Maes, Anja Arts, and Leo Noordman. Reference management in instructive discourse. *Discourse Processes*, 37(2):117–144, 2004.

Zofia Malisz, Marcin Włodarczak, Hendrik Buschmeier, Stefan Kopp, and Petra Wagner. Prosodic characteristics of feedback expressions in distracted and non-distracted listeners. In *Proceedings of The Listening Talker. An Interdisciplinary Workshop on Natural and Synthetic Modification of Speech in Response to Listening Conditions*, Edinburgh, UK, 2012.

Tomasz Marciniak and Michael Strube. Beyond the pipeline: Discrete optimization in NLP. In *Proceedings of the 9th Conference on Computational Natural Language Learning*, Ann Arbor, MI, 2005.

Matthew Marge and Alexander I. Rudnicky. Towards overcoming miscommunication in situated dialogue by asking questions. In *Proceedings of the AAAI Fall Symposium on Building Representations of Common Ground with Intelligent Agents*, Arlington, VA, 2011.

Vivien Mast, Cui Jian, and Desislava Zhekova. Elaborate descriptive information in indoor route instructions. In *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, Sapporo, Japan, 2012.

Mark Maybury. Communicative acts for explanation generation. *International Journal of Man-Machine Studies*, 37(2):135–172, 1992.

Drew McDermott. The 1998 AI Planning Systems Competition. *AI Magazine*, 21 (2):35–55, 2000.

David D. McDonald and James D. Pustejovsky. TAG's as a grammatical formalism for generation. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, IL, 1985.

Antje S. Meyer, Astrid M. Sleiderink, and Willem J. M. Levelt. Viewing and naming objects: Eye movements during noun phrase production. *Cognition*, 66 (2):B25–B33, 1998.

Marlene Meyer, Robrecht P. R. D. van der Wel, and Sabine Hunnius. Higher-order action planning for individual and joint object manipulations. *Experimental Brain Research*, 225(4):579–588, 2013.

Margaret Mitchell. Class-based ordering of prenominal modifiers. In *Proceedings of the 12th European Workshop on Natural Language Generation*, Athens, Greece, 2009.

Margaret Mitchell, Ehud Reiter, and Kees van Deemter. Typicality and object reference. In *Proceedings of the 35th Annual Conference of the Cognitive Science Society*, Berlin, Germany, 2013a.

Margaret Mitchell, Ehud Reiter, and Kees van Deemter. Attributes in visual object reference. In *Proceedings of the PRE-CogSci 2013 Workshop on the Production of Referring Expressions*, Berlin, Germany, 2013b.

Johanna D. Moore and Cécile L. Paris. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–694, 1993.

Yukiko I. Nakano, Gabe Reinstein, Tom Stocky, and Justine Cassell. Towards a model of face-to-face grounding. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, Sapporo, Japan, 2003.

Dana Nau, Tsz-Chiu Au, Okhtay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu, and Fusun Yaman. SHOP2: An HTN Planning System. *Journal of Artificial Intelligence Research*, 20:379–404, 2003.

Bernhard Nebel, Christian Dornhege, and Andreas Hertle. How much does a household robot need to know in order to tidy up? In *Proceedings of the AAAI Workshop on Intelligent Robotic Systems*, Bellevue, WA, 2013.

Tim Paek and Eric Horvitz. Uncertainty, utility, and misunderstanding: A decision-theoretic perspective on grounding in conversational systems. In *AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems*, 1999.

Hector Palacios and Hector Geffner. Compiling uncertainty away in conformant planning problems with bounded width. *Journal of Artificial Intelligence Research*, 35:623–675, 2009.

Ivandré Paraboni and Kees van Deemter. Reference and the facilitation of search in spatial domains. *Language and Cognitive Processes*, to appear.

Ivandré Paraboni, Kees van Deemter, and Judith Masthoff. Generating referring expressions: Making referents easy to identify. *Computational Linguistics*, 33 (2):229–254, 2007.

Kyungdae Park, Jiyoung Kang, Sanghyuk Koh, Mijung Park, Saegee Oh, and Chihoon Lee. Method for operating user functions based on eye tracking and mobile device adapted thereto, 2013. Patent US 2013/0135196 A1.

Thomas Pechmann. Incremental speech production and referential overspecification. *Linguistics*, 27:89–110, 1989.

C. Raymond Perrault and James F. Allen. A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics*, 6(3–4):167–182, 1980.

Ronald Petrick and Fahiem Bacchus. A knowledge-based approach to planning with incomplete information and sensing. In *Proceedings of the 6th International Conference on Artificial Intelligence Planning and Scheduling*, Menlo Park, CA, 2002.

Ronald Petrick and Mary Ellen Foster. What would you like to drink? Recognising and planning with social states in a robot bartender domain. In *Proceedings of the 8th International Workshop on Cognitive Robotics at AAAI*, Toronto, Canada, 2012.

Ola Pettersson. Execution monitoring in robotics: A survey. *Robotics and Autonomous Systems*, 53(2):73–88, 2005.

Massimo Poesio. A situation-theoretic formalization of definite description interpretation in plan elaboration dialogues. *Situation theory and its applications*, 3:339–374, 1993.

Joseph Polifroni, Lynette Hirschman, Stephanie Seneff, and Victor Zue. Experiments in evaluating interactive spoken language systems. In *Proceedings of the Workshop on Speech and Natural Language*, Harriman, NY, 1992.

Paul Portner. Imperatives and modals. *Natural Language Semantics*, 15(4):351–383, 2007.

David Nicolás Racca, Luciana Benotti, and Pablo Duboue. The GIVE-2.5 C Generation System. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, Nancy, France, 2011.

Ehud Reiter and Robert Dale. *Building natural language generation systems*. Cambridge University Press, 2000.

Daniel C. Richardson and Rick Dale. Looking to understand: The coupling between speakers' and listeners' eye movements and its relationship to discourse comprehension. *Cognitive Science*, 29(6):1045–1060, 2005.

Mark Owen Riedl and R. Michael Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268, 2010.

Verena Rieser and Oliver Lemon. Natural language generation as planning under uncertainty for spoken dialogue systems. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, Athens, Greece, 2009.

Christian Roßnagel. Cognitive load and perspective-taking: Applying the automatic-controlled distinction to verbal communication. *European Journal of Social Psychology*, 30(3):429–445, 2000.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd edition, 2009.

Michael F. Schober and Herbert H. Clark. Understanding by addressees and overhearers. *Cognitive Psychology*, 21(2):211–232, 1989.

Marc Schröder and Jürgen Trouvain. The German text-to-speech synthesis system MARY: A tool for research, development and teaching. *International Journal of Speech Technology*, 6:365–377, 2003.

Niels Schütte, John Kelleher, and Brian Mac Namee. Visual salience and reference resolution in situated dialogues: A corpus-based evaluation. In *Proceedings of the AAAI 2010 Fall Symposium on Dialog with Robots*, Arlington, VA, 2010.

Julie C. Sedivy, Michael K. Tanenhaus, Craig G. Chambers, and Gregory N. Carlson. Achieving incremental semantic interpretation through contextual representation. *Cognition*, 71(2):109–147, 1999.

James Shaw and Vasileios Hatzivassiloglou. Ordering among premodifiers. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, College Park, MD, 1999.

Gabriel Skantze and David Schlangen. Incremental dialogue processing in a micro-domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Athens, Greece, 2009.

Dustin Smith and Henry Lieberman. Generating and interpreting referring expressions as belief state planning and plan recognition. In *Proceedings of the 14th European Workshop on Natural Language Generation*, Sofia, Bulgaria, 2013.

Philipp Spanger, Masaaki Yasuhara, Ryu Iida, and Takenobu Tokunaga. Using extra-linguistic information for generating demonstrative pronouns in a situated collaboration task. In *Proceedings of the PRE-CogSci 2009 Workshop on the Production of Referring Expressions*, Amsterdam, The Netherlands, 2009.

Maria Staudte and Matthew Crocker. Investigating joint attention mechanisms through human-robot interaction. *Cognition*, 120(2):268–291, 2011.

Maria Staudte, Matthew Crocker, Alexander Koller, and Konstantina Garoufi. Grounding spoken instructions using listener gaze in dynamic virtual environments. In *Abstracts of the 5th Workshop on Embodied and Situated Language Processing*, Newcastle, UK, 2012a.

Maria Staudte, Alexander Koller, Konstantina Garoufi, and Matthew Crocker. Using listener gaze to augment speech generation in a virtual 3D environment. In *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, Sapporo, Japan, 2012b.

Manfred Stede. Discourse processing. *Synthesis Lectures on Human Language Technologies*, 4(3):1–165, 2011.

Mark Steedman and Ronald Petrick. Planning dialog actions. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium, 2007.

Amanda Stent. Computational approaches to the production of referring expressions: Dialog changes (almost) everything. In *Proceedings of the PRE-CogSci 2011 Workshop on the Production of Referring Expressions*, Boston, MA, 2011.

Laura Stoia, Donna Byron, Darla Shockley, and Eric Fosler-Lussier. Sentence planning for realtime navigational instructions. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, New York, NY, 2006a.

Laura Stoia, Darla Shockley, Donna Byron, and Eric Fosler-Lussier. Noun phrase generation for situated dialogs. In *Proceedings of the 4th International Natural Language Generation Conference*, Sydney, Australia, 2006b.

Laura Stoia, Darla Shockley, Donna Byron, and Eric Fosler-Lussier. SCARE: A situated corpus with annotated referring expressions. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco, 2008.

Matthew Stone. Economy in embodied utterances. In L. Goldstein, editor, *Brevity*. Oxford University Press, to appear.

Matthew Stone and Christine Doran. Sentence planning as description using Tree Adjoining Grammar. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, Spain, 1997.

Matthew Stone and Bonnie Webber. Textual economy through close coupling of syntax and semantics. In *Proceedings of the 9th International Workshop on Natural Language Generation*, Niagara-on-the-Lake, Canada, 1998.

Matthew Stone, Christine Doran, Bonnie Webber, Tonia Bleam, and Martha Palmer. Microplanning with communicative intentions: The SPUD system. *Computational Intelligence*, 19(4):311–381, 2003.

Kristina Striegnitz and Filip Majda. Landmarks in navigation instructions for a virtual environment. In *Proceedings of the Generation Challenges Session at the 12th European Workshop on Natural Language Generation*, Athens, Greece, 2009.

Kristina Striegnitz, Alexandre Denis, Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Mariët Theune. Report on the Second Second NLG Challenge on Generating Instructions in Virtual Environments (GIVE-2.5). In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, Nancy, France, 2011.

Kristina Striegnitz, Hendrik Buschmeier, and Stefan Kopp. Referring in installments: A corpus study of spoken object references in an interactive virtual environment. In *Proceedings of the 7th International Natural Language Generation Conference*, Utica, IL, 2012.

Michael K. Tanenhaus, Michael J. Spivey-Knowlton, Kathleen M. Eberhard, and Julie C. Sedivy. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268:1632–1634, 1995.

Ariane Tom and Michel Denis. Referring to landmark or street information in route directions: What difference does it make? In W. Kuhn, M. Worboys,

and S. Timpf, editors, *Spatial Information Theory. Foundations of Geographic Information Science*, volume 2825 of *LNCS*, pages 362–374. Springer, 2003.

Kenneth Train. *Discrete choice methods with simulation*. Cambridge University Press, second edition, 2009.

David Traum. *A computational theory of grounding in natural language conversation*. PhD thesis, University of Rochester, 1994.

David Traum. Computational models of grounding in collaborative systems. In *AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems*, 1999.

David Traum and Elizabeth Hinkelman. Conversation acts in task-oriented spoken dialogue. *Computational intelligence*, 8(3):575–599, 1992.

Kees van Deemter. Generating referring expressions that involve gradable properties. *Computational Linguistics*, 32(2), 2006.

Kees van Deemter, Albert Gatt, Ielka van der Sluis, and Richard Power. Generation of referring expressions: Assessing the incremental algorithm. *Cognitive Science*, 36(5):799–836, 2012.

Ielka van der Sluis and Emiel Krahmer. Generating multimodal references. *Discourse Processes*, 44(3):145–174, 2007.

Jette Viethen. *The generation of natural descriptions: Corpus-based investigations of referring expressions in visual domains.* PhD thesis, Macquarie University, 2011.

Jette Viethen and Robert Dale. The use of spatial relations in referring expression generation. In *Proceedings of the 5th International Natural Language Generation Conference*, Salt Fork, OH, 2008.

Jette Viethen, Robert Dale, Emiel Krahmer, Mariët Theune, and Pascal Touset. Controlling redundancy in referring expressions. In *Proceedings of the 6th International Language Resources and Evaluation Conference*, Marrakech, Morocco, 2008.

Jette Viethen, Martijn Goudbeek, and Emiel Krahmer. The impact of colour difference and colour codability on reference production. In *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, Sapporo, Japan, 2012.

Marilyn Walker, Jerry Wright, and Irene Langkilde. Using natural language processing and discourse features to identify understanding errors in a spoken dialogue system. In *Proceedings of the 17th International Conference on Machine Learning*, Stanford, CA, 2000.

Marilyn Walker, Owen Rambow, and Monica Rogati. Training a sentence planner for spoken dialogue using boosting. *Computer Speech & Language*, 16(3–4): 409–433, 2002.

Liane Wardlow Lane and Victor S. Ferreira. Speaker-external versus speaker-internal forces on utterance form: Do cognitive demands override threats to referential success? *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 34(6):1466–1481, 2008.

Andrea Weber, Bettina Braun, and Matthew Crocker. Finding referents in time: Eye-tracking evidence for the role of contrastive accents. *Language and Speech*, 49(3):367–392, 2006.

Jason D. Williams. A belief tracking challenge task for spoken dialog systems. In *Proceedings of the NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community*, Montreal, Canada, 2012.

Lynsey Wolter, Kristen Skovbroten Gorman, and Michael K. Tanenhaus. Scalar reference, contrast and discourse: Separating effects of linguistic discourse from availability of the referent. *Journal of memory and language*, 65(3):299–317, 2011.

Wei Xu and Alexander I. Rudnicky. Task-based dialog management using an agenda. In *Proceedings of the ANLP/NAACL Workshop on Conversational systems*, Seattle, WA, 2000.

Sungwook Yoon, Alan Fern, and Robert Givan. FF-Replan: A baseline for probabilistic planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Providence, RI, 2007.

Michael Young and Johanna Moore. Does discourse planning require a special-purpose planner? In *Proceedings of the AAAI Workshop on Planning for Inter-Agent Communication*, Seattle, WA, 1994.

Michael Young, Johanna Moore, and Martha Pollack. Towards a principled representation for discourse plans. In *Proceedings of the 16th Annual Meeting of the Cognitive Science Society*, Atlanta, GA, 1994.

Hendrik Zender, Geert-Jan M Kruijff, and Ivana Kruijff-Korbayová. Situated resolution and generation of spatial referring expressions for robotic assistants. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, Pasadena, CA, 2009.

Appendix A

Grammar specification examples

This appendix supplements Chapters 3 and 4 with a specification of example lexical entries and corresponding planning operators. Lexicons follow LTAG, are hand-written, and are encoded in XML; planning operators are the result of automatic conversion (`https://code.google.com/p/crisp-nlg`) and are encoded in PDDL.

A.1 The lexicon of Fig. 3.4

```
<!-- grammar -->

<tree id="i.transimperative">
  <node cat="s" sem="self">
     <node cat="vp" sem="self">
      <leaf cat="v" type="anchor" sem="self"/>
   <leaf cat="np" type="substitution" sem="obj" />
     </node>
   </node>
</tree>

<tree id="i.intransimperative">
  <node cat="s" sem="self">
     <node cat="vp" sem="self">
      <leaf cat="v" type="anchor" sem="self"/>
     </node>
   </node>
</tree>
```

```
<tree id="a.sentconjunction">
  <node cat="s" sem="self">
    <leaf cat="s" type="foot" sem="self"/>
    <leaf cat="conj" type="anchor" sem="self" />
    <leaf cat="s" type="substitution" sem="other" />
  </node>
</tree>

<!-- lexicon -->

<entry word='push' pos='v'>
  <tree refid="i.transimperative">
    <semreq>visible(p,o,obj)</semreq>
    <pragcond>player-position(p)</pragcond>
    <pragcond>player-orientation(o)</pragcond>
    <prageff>premod-index-zero(id(obj))</prageff>
    <impeff>push(obj)</impeff>
    <param type="positiontype">p</param>
    <param type="orientationtype">o</param>
  </tree>
</entry>

<entry word='turn_left' pos='v'>
  <tree refid="i.intransimperative">
    <pragcond>player-orientation(o1)</pragcond>
    <pragcond>next-orientation-left(o1,o2)</pragcond>
    <prageff>not(player-orientation(o1))</prageff>
    <prageff>player-orientation(o2)</prageff>
    <impeff>turn_left()</impeff>
    <param type="orientationtype">o1</param>
    <param type="orientationtype">o2</param>
  </tree>
</entry>

<entry word='and' pos='conj'>
  <tree refid="a.sentconjunction">
    <pragcond>next-referent(self,other)</pragcond>
  </tree>
</entry>
```

A.2 The planning operators of Fig. 3.5

```
(:action init-transimperative-push
   :parameters (?x - individual  ?u - syntaxnode  ?x1 - individual
```

```
      ?u1 - syntaxnode  ?un - syntaxnode  ?p - positiontype
      ?o - orientationtype  )
   :precondition (and (current ?u1) (next ?u1 ?un) (referent ?u ?x)
      (subst s ?u) (visible ?p ?o ?x1) (player-position ?p)
      (player-orientation ?o))
   :effect (and (not (current ?u1)) (current ?un)
      (not (subst s ?u)) (todo-1 imp-push ?x1)
      (premod-index-zero ?u1) (subst np ?u1) (referent ?u1 ?x1)
      (forall (?y - individual  ) (when (and (not (= ?y ?x1))
      (visible ?p ?o ?y)) (distractor ?u1 ?y)))
      (canadjoin s ?u) (canadjoin v ?u) (canadjoin vp ?u))
)

(:action init-intransimperative-turn_left
   :parameters (?x - individual  ?u - syntaxnode
      ?o2 - orientationtype  ?o1 - orientationtype  )
   :precondition (and (referent ?u ?x) (subst s ?u)
      (player-orientation ?o1) (next-orientation-left ?o1 ?o2))
   :effect (and (not (subst s ?u)) (todo-0 imp-turn_left)
      (not (player-orientation ?o1)) (player-orientation ?o2)
      (canadjoin s ?u) (canadjoin v ?u) (canadjoin vp ?u))
)

(:action aux-sentconjunction-and
   :parameters (?x - individual  ?u - syntaxnode  ?x1 - individual
      ?u1 - syntaxnode  ?un - syntaxnode  )
   :precondition (and (current ?u1) (next ?u1 ?un) (referent ?u ?x)
      (canadjoin s ?u) (next-referent ?x ?x1))
   :effect (and (not (current ?u1)) (current ?un)
      (not (mustadjoin s ?u)) (subst s ?u1) (referent ?u1 ?x1)
      (canadjoin s ?u) (canadjoin conj ?u))
)
```

A.3 The lexicon of Fig. 3.6

```
<!-- grammar -->

<tree id="a.an">
  <node cat="n" sem="self">
    <leaf cat="a" type="anchor" sem="self"/>
    <leaf cat="n" type="foot" sem="self" />
  </node>
</tree>
```

```
  <!-- lexicon -->

  <entry word='left' pos='a'>
    <tree refid='a.an'>
      <pragcond>forall(y,not(and(distractor(id(self),y),
         left-of(y,self))))</pragcond>
      <pragcond>and(left-of(self,z),
         distractor(id(self),z))</pragcond>
      <prageff>forall(y,when(left-of(self,y),
         not(distractor(id(self),y))))</prageff>
      <prageff>not(premod-index-zero(id(self)))</prageff>
      <prageff>not(premod-index-one(id(self)))</prageff>
      <prageff>premod-index-two(id(self))</prageff>
      <var>y</var>
      <param type="individual">z</param>
    </tree>
  </entry>

  <entry word='red' pos='a'>
    <tree refid='a.an'>
      <semcontent>red(self)</semcontent>
      <pragcond>not(premod-index-two(id(self)))</pragcond>
      <prageff>not(premod-index-zero(id(self)))</prageff>
      <prageff>premod-index-one(id(self))</prageff>
    </tree>
  </entry>
```

A.4 The planning operators of Fig. 3.6

```
(:action aux-an-left
   :parameters (?x - individual  ?u - syntaxnode  ?z - individual)
   :precondition (and (referent ?u ?x) (canadjoin n ?u)
      (forall (?y - individual) (not (and (distractor ?u ?y)
      (left-of ?y ?x)))) (and (left-of ?x ?z) (distractor ?u ?z)))
   :effect (and (not (mustadjoin n ?u)) (forall (?y - individual)
      (when (left-of ?x ?y) (not (distractor ?u ?y))))
      (not (premod-index-zero ?u)) (not (premod-index-one ?u))
      (premod-index-two ?u) (canadjoin n ?u) (canadjoin a ?u))
)

(:action aux-an-red
   :parameters (?x - individual  ?u - syntaxnode  )
   :precondition (and (referent ?u ?x) (canadjoin n ?u) (red ?x)
      (not (premod-index-two ?u)))
```

```
   :effect (and (not (mustadjoin n ?u))
      (not (needtoexpress-1 pred-red ?x))
      (not (premod-index-zero ?u)) (premod-index-one ?u)
      (forall (?y - individual  ) (when (not (and (red ?y)))
      (not (distractor ?u ?y)))) (canadjoin n ?u) (canadjoin a ?u))
)
```

A.5 The lexicon of Fig. 4.4

```
<!-- grammar -->

<tree id="a.an">
  <node cat="n" sem="self">
    <leaf cat="a" type="anchor" sem="self"/>
    <leaf cat="n" type="foot" sem="self" />
  </node>
</tree>

<!-- lexicon -->

<entry word='red' pos='a'>
  <tree refid='a.an'>
    <semcontent>red(self)</semcontent>
    <pragcond>needtodecide(absolute-attr,self)</pragcond>
    <pragcond>not(premod-index-two(id(self)))</pragcond>
    <prageff>not(premod-index-zero(id(self)))</prageff>
    <prageff>premod-index-one(id(self))</prageff>
    <prageff>not(needtodecide(absolute-attr,self))</prageff>
    <costclass>absolute</costclass>
  </tree>
</entry>

<entry word='non_absolute_red' pos='a'>
  <tree refid='a.an'>
    <pragcond>needtodecide(absolute-attr,self)</pragcond>
    <prageff>not(needtodecide(absolute-attr,self))</prageff>
    <costclass>non-absolute</costclass>
  </tree>
</entry>
```

A.6 The planning operators of Fig. 4.4

Assuming, for a given referential scene, the cost assignment of Table 4.3, we obtain planning operators as follows:

```
(:action aux-an-red
   :parameters (?x - individual  ?u - syntaxnode  )
   :precondition (and (referent ?u ?x) (canadjoin n ?u) (red ?x)
      (not (premod-index-two ?u)) (needtodecide absolute-attr ?x))
   :effect (and (not (mustadjoin n ?u))
      (not (needtoexpress-1 pred-red ?x))
      (not (premod-index-zero ?u)) (premod-index-one ?u)
      (not (needtodecide absolute-attr ?x))
      (forall (?y - individual) (when (not (and (red ?y)))
      (not (distractor ?u ?y)))) (canadjoin n ?u) (canadjoin a ?u)
      (increase (total-cost) 0.03))
)

(:action aux-an-non_absolute_red
   :parameters (?x - individual  ?u - syntaxnode  )
   :precondition (and (referent ?u ?x)
      (needtodecide absolute-attr ?x))
   :effect (and (not (needtodecide absolute-attr ?x))
      (increase (total-cost) 0.00))
)
```