REFERRING EXPRESSION GENERATION TOWARDS MEDIATING SHARED PERCEPTUAL BASIS IN SITUATED DIALOGUE

By

Rui Fang

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science - Doctor of Philosophy

2015
Situated human-robot dialogue has received increasing attention in recent years. In situated dialogue, robots/artificial agents and their human partners are co-present in a shared physical world. Robots need to automatically perceive and make inference of the shared environment. Due to its limited perceptual and reasoning capabilities, the robot’s representation of the shared world is often incomplete, error-prone, and significantly mismatched from that of its human partner’s. Although physically co-present, a joint perceptual basis between the human and the robot cannot be established. Thus, referential communication between the human and the robot becomes difficult.

Robots need to collaborate with human partners to establish a joint perceptual basis, referring expression generation (REG) thus becomes an important problem in situated dialogue. REG is the task of generating referring expressions to describe target objects such that the intended objects can be correctly identified by the human. Although extensively studied, most existing REG algorithms were developed and evaluated under the assumption that agents and humans have access to the same kind of domain information. This is clearly not true in situated dialogue. The objective of this thesis is investigating how to generate referring expressions to mediate mismatched perceptual basis between humans and agents.

As a first step, a hypergraph-based approach is developed to account for group-based spatial relations and uncertainties in perceiving the environment. This approach outperforms a previous graph-based approach with an absolute gain of 9%. However, while these graph-based approaches perform effectively when the agent has perfect knowledge or perception of the environment (e.g.,
84%), they perform rather poorly when the agent has imperfect perception of the environment (e.g., 45%). This big performance gap indicates that when the agent applies traditional approaches (which usually generate a single minimum description) to generate referring expressions to describe target objects, the intended objects often cannot be correctly identified by the human.

To address this problem, motivated by collaborative behaviors in human referential communication, two collaborative models are developed - an episodic model and an installment model - for REG. In both models, instead of generating a single referring expression to describe a target object as in the previous work, it generates multiple small expressions that lead to the target object with a goal to minimize the collaborative effort. In particular, the installment model incorporates human feedback in a reinforcement learning framework to learn the optimal generation strategies. Our empirical results have shown that the episodic model and the installment model outperform our non-collaborative hypergraph-based approach with an absolute gain of 6% and 21% respectively.

Lastly, the collaborative models are further extended to embodied collaborative models for facilitate human-robot interaction. These embodied models seamlessly incorporate robot gesture behaviors (i.e., pointing to an object) and human's gaze feedback (i.e., looking at a particular object) into the collaborative model for REG. The empirical results have shown that when robot gestures and human verbal feedback is incorporated, the new collaborative model achieves over 28% absolute gains compared to the baseline collaborative model. This thesis further discusses the opportunities and challenges brought by modeling embodiment in collaborative referential communication in human-robot interaction.
ACKNOWLEDGMENTS

First and foremost I want to thank my Ph.D. advisor Dr. Joyce Chai. I thank her for taking me on as a PhD student in computer science. I thank her for teaching me how to select research problems, think them critically, solve them and present their solutions. I thank her for teaching me how to write research papers. She has guided me with her vast experience and invaluable suggestions, lightened up the way in my darkest times and encouraged me a lot in my Ph.D. life. I can truly say that without her, this thesis would never have happened.

I would also like to express my thanks to all members in the Language and Interaction Research (LAIR) Group. I thanks all of you for your valuable discussions and comments. Your collaboration and constructive criticism have been tremendous assets throughout my Ph.D. study.

Thanks also to my parents who have been extremely understanding and supportive of my Ph.D. studies. I thanks my little cute daughter TiTi for coming to my life during my Ph.D. study. Finally, the warmest of thanks to my wonderful wife, Jingmin Liu, who has encouraged me so much over these years. I thank her for her understanding and patience while I was away from home during the period of my Ph.D. study. I owe everything to her, without her everlasting love, this thesis would never be completed. To her, I dedicate this work.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>viii</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
</tbody>
</table>

## Chapter 1 Introduction
1.1 Referring Expression Generation ........................................... 1
1.2 REG in Situated Dialogue ................................................. 4
1.3 Contributions of this Dissertation .................................. 8
1.4 Outline ................................................................. 10

## Chapter 2 Related Work
2.1 Greedy Search Approaches .................................................. 13
2.2 Incremental Approach ....................................................... 15
2.3 Graph-based Approaches .................................................... 18
2.4 Plan-based approaches ...................................................... 23
2.5 Reinforcement Learning based Approaches ........................... 25
2.6 Other Approaches .......................................................... 28
2.7 Evaluation of REG approaches ........................................... 32
   2.7.1 Corpora for REG evaluation ........................................... 32
   2.7.1.1 The TUNA corpus ...................................................... 32
   2.7.1.2 The GRE3D3 corpus .................................................. 33
   2.7.1.3 the GIVE-2 Corpus .................................................. 34
   2.7.2 Evaluation Metrics .................................................... 35
   2.7.2.1 Human-likeness Metrics ........................................... 35
   2.7.2.2 Identification Accuracy .......................................... 37

## Chapter 3 Graph-based Approach for REG
3.1 Introduction ................................................................. 38
3.2 Hyper-graph based REG ...................................................... 39
   3.2.1 Motivation ............................................................... 39
   3.2.2 Hypergraph Representation of the Perceived Scenes .............. 41
   3.2.3 Hypergraph Pruning .................................................... 42
   3.2.4 Symbolic Descriptors for Attributes ................................ 45
      3.2.4.1 Lexicon with Grounded Semantics ............................. 46
      3.2.4.2 Attribute Descriptors and Cost Functions ................. 47
   3.2.5 Graph Matching for REG ............................................. 48
3.3 Empirical Evaluations .................................................... 50
   3.3.1 Evaluation Setup ..................................................... 50
      3.3.1.1 Mismatch Views .................................................... 50
5.4 Conclusions ................................................................. 108

Chapter 6 Future Directions .............................................. 109
  6.1 Dialogue Context ...................................................... 109
  6.2 Joint Optimization .................................................... 109
  6.3 Online Learning ...................................................... 110

REFERENCES ................................................................. 111
LIST OF TABLES

Table 3.1: Results with different cost functions ........................................ 55

Table 3.2: Results of comparing perfect perception and imperfect perception of the
shared world. ......................................................................................... 57

Table 3.3: Results of comparing minimum effort and extra effort using hypergraphs . 58

Table 4.1: Features used in the installment model. We use the following abbrevia-
tions, o: an object to describe, re: generation strategy for an object, lm:
landmark object, tar: the target object, sp: spatial relation .................... 75

Table 4.2: Comparison of accuracy for referent identification based on expressions
generated from three models. ................................................................. 79

Table 5.1: Features used in the installment model [1]. .................................. 91

Table 5.2: Comparison of accuracy for the eight experiment conditions. .......... 99

Table 5.3: Accuracy of object identification game, when gesture is present (R+) or
absent (R-) .......................................................................................... 100

Table 5.4: Rate of successful object identification game, with different feedback
conditions ............................................................................................ 101

Table 5.5: Statistics for landmark and intended object in Gaze feedback only condi-
tion (G), Gaze and Verbal feedback condition (VG) ............................. 102

Table 5.6: Rate of different gaze patterns, under Verbal Only condition and Verbal
and Gesture Condition ...................................................................... 104
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>The pipeline for Natural Language Generation [2].</td>
<td>3</td>
</tr>
<tr>
<td>Figure 1.2</td>
<td>The experiment setup in [3] for the investigation of collaborative referring</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>under mis-matched perceptual capabilities.</td>
<td></td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>An algorithm for the Greedy Heuristic: the Full Brevity algorithm</td>
<td>14</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>The algorithm that takes REG as a search problem</td>
<td>15</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>The Incremental Algorithm</td>
<td>16</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>an example domain represented as a labeled directed graph: object 1 is a</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>cup and it is to the left of object 2 which is a box.</td>
<td></td>
</tr>
<tr>
<td>Figure 2.5</td>
<td>The graph-based REG algorithm</td>
<td>20</td>
</tr>
<tr>
<td>Figure 2.6</td>
<td>Deictic gesture pointing to the scene [4, 5]</td>
<td>21</td>
</tr>
<tr>
<td>Figure 2.7</td>
<td>Deictic gesture graph of the scene [4, 5]</td>
<td>21</td>
</tr>
<tr>
<td>Figure 2.8</td>
<td>An example scene from the TUNA corpus [6]. An example description</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>for the target object: ‘the black chair on the right’.</td>
<td></td>
</tr>
<tr>
<td>Figure 2.9</td>
<td>An example scene from the GRE3D corpus [7]. A description generated</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>by human: ‘the small green ball on the top of a box’.</td>
<td></td>
</tr>
<tr>
<td>Figure 2.10</td>
<td>An example scene from GIVE-2 corpus [8]. An example instruction given</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>by human: ‘keep going through the room and make a left, click that yellow</td>
<td></td>
</tr>
<tr>
<td></td>
<td>button on your left’.</td>
<td></td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>An original scene and its impoverished scene processed by CV algorithm</td>
<td>40</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>An example of hypergraph representing the perceived scene (a partial scene</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>only including object 7, 8, 9, 11, 13 for Figure 3.1(a)).</td>
<td></td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Hypergraph based REG</td>
<td>49</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>The procedure of generating the impoverished image from the original image</td>
<td>51</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>a web-based interface shown to the Mechanical turk workers</td>
<td>52</td>
</tr>
<tr>
<td>Figure 4.1:</td>
<td>An original scene shown to the human and the re-rendering of the robot’s internal representation from [9]. Each object/group has an ID associated for the identification purpose.</td>
<td>66</td>
</tr>
<tr>
<td>Figure 4.2:</td>
<td>The Episodic Model for REG</td>
<td>84</td>
</tr>
<tr>
<td>Figure 4.3:</td>
<td>The Agent-Environment Interface in MDP [10]</td>
<td>85</td>
</tr>
<tr>
<td>Figure 4.4:</td>
<td>The Installment Model for REG</td>
<td>85</td>
</tr>
<tr>
<td>Figure 4.5:</td>
<td>a web-based interface shown to the Mechanical Turk workers for the episodic model</td>
<td>86</td>
</tr>
<tr>
<td>Figure 4.6:</td>
<td>a web-based interface shown to the Mechanical Turk workers for the installment model</td>
<td>86</td>
</tr>
<tr>
<td>Figure 4.7:</td>
<td>The number of objects correctly identified at an interval of every 200 training sessions</td>
<td>87</td>
</tr>
<tr>
<td>Figure 4.8:</td>
<td>The distribution of failure cases at the 1st, 2nd, 3rd and after the 3rd expression in the episodic model</td>
<td>87</td>
</tr>
<tr>
<td>Figure 5.1:</td>
<td>An example of situated setup and referential communication</td>
<td>92</td>
</tr>
<tr>
<td>Figure 5.2:</td>
<td>The average number of installments per game in each of the feedback conditions</td>
<td>103</td>
</tr>
<tr>
<td>Figure 5.3:</td>
<td>The average fixation time (in seconds) for the conditions with and without gesture</td>
<td>106</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Referring Expression Generation

Natural Language Generation (NLG) is one of the major sub-fields in natural language processing [2]. NLG is the task of generating natural language surface forms from a machine representation of the non-linguistic information such as knowledge base or logical form. Traditionally, NLG systems have used a pipeline architecture (e.g., Figure 1.1) that divides the generation process into a number of distinct stages for decision making. While different system designers have chosen a different number of modules (see [2] for a detailed overview of generation tasks), the core tasks of any NLG system are typically the choice of semantic concepts, their organization into distinct sub-messages and their realization as a string of words. These three tasks are commonly referred to as Document planning, Microplanning and surface realization each with its own subtasks. Below is an overview of the generation tasks that are performed within each stage.

- Document planning

1. **Content determination** is the task of converting the raw data input to the specific kind of data objects (or ‘messages’) that serve as a basis for the subsequent generation tasks. An important subtask here is to decide which input information to express in the text.

2. **Document structuring** (or discourse planning) is the process of ordering the information to be expressed and determining the structure of the output text. The result is a
A text plan, often in the form of a tree structure representing the order and grouping of the messages, and the relations between them.

- **Microplanning**

  1. **Lexicalisation** is the task of choosing the right words to express the input information. Often, a concept can be expressed using different words or phrases; the task of the lexicalization procedure is to choose the one that is most appropriate in the given context.

  2. **Aggregation** is the process of deciding which information to put in one sentence. Pieces of information that form separate input messages may be joined together and expressed using one sentence which, for instance, contains a conjunction or a relative clause.

  3. **Referring expression generation** is the task of creating phrases to identify domain entities. This task involves choosing the type of expression (e.g., a pronoun or a definite description) and choosing the properties to include in the description.

- **Surface realization**

  1. **Linguistic realization** is the task of creating grammatical sentences. This involves the application of syntactic and morphological rules that determine aspects like word order and agreement.

  2. **Structure realization**, is the task of converting the output text into some required external format (e.g., by adding HTML mark-up), which should be correct according to the rules of syntax, morphology, and orthography.
From the pipelined architecture of NLG, we can see that Referring Expression Generation (REG) is a subtask of NLG, which involves creating referring expressions (noun phrases) can identify a referent to a listener. These expressions generally take the form of a definite noun phrase such as ‘the small yellow ball’. However, when we consider REG in isolation, it quickly becomes clear that things are more complicated than this. It appears that some of the other subtasks within the NLG pipeline are in actual fact tasks that have to be tackled within REG itself as well. A NLG system will as a minimum have to perform the tasks of content determination (selecting the properties of the target referent to be mentioned), lexicalization (choosing the words to represent the properties), and linguistic realization (constructing a grammatically correct noun phrase). By far the majority of work in REG focuses on the content determination subtask.

The Task of REG. The main function of referring expressions is to distinguish the target referent from the objects around it. Other functions that referring expressions can fulfill, such as describing, convincing, informing or directing, are beyond the current main scope of REG.

Assume that a finite domain $O$ consists of objects with attributes $A$. For example, $O = \{o_1; o_2; o_3\}$ and $A = \{Type; Color; Position\}$. The attribute-value pair for each object in domain $O$ are

- $o_1: \{Type – apple, Color – red, Position – left\}$
- $o_2: \{Type – apple, Color – yellow, Position – left\}$
\( \alpha_3 : \{ \text{Type} – \text{banana}, \text{Color} – \text{yellow}, \text{Position} – \text{right} \} \)

The REG task can now be defined as: given a target (or referent) object \( \text{tar} \in O \), find a set of attribute-value pairs \( L \) whose conjunction is true of the target but not of any of the distractors (i.e., \( O – r \)), the domain objects different from the target). \( L \) is called a distinguishing description of the target. For example, suppose that \( o_1 \) is the target (and hence \( o_2, o_3 \) are the distractors), then \( L \) could, for example, be either (Type-apple, Color-red) or (Color-yellow, Position-left), which could be realized as ‘the red apple’ or ‘the yellow object on the left’.

### 1.2 REG in Situated Dialogue

Situated human-robot dialogue has received increasing attention in recent years. In situated dialogue, robots/artificial agents and their human partners are co-present in a shared physical world. Robots need to automatically perceive and make inference of the shared environment. Due to its limited perceptual and reasoning capabilities, the robot’s representation of the shared world is often incomplete, error-prone, and significantly mismatched from that of its human partner’s. Although physically co-present, a joint perceptual basis between the human and the robot cannot be established [11]. Thus, referential communication between the human and the robot becomes difficult.

To investigate collaborative referring under mis-matched perceptual capabilities, previously, we have conducted experiments on human-human interaction [3]. In these experiments, we have two human subjects play a set of naming games. One subject (referred to as the human-player) is provided with an original image containing over ten objects (Figure 1(a)). Several of these objects have secret names. The other subject (referred to as the robot-player) only has access to an
impoverished image of the same scene (Figure 1(b)) to mimic the lower perceptual capability of a robot. The human-players goal is to communicate the names of target objects to the robot-player so that the robot-player knows which object in his view has what name. The impoverished image was automatically created by applying standard computer vision algorithms and thus may contain different types of processing errors (e.g., mis-segmentation and/or mis-recognition). Using this setup, we have collected a set of dialogues. The following shows an example dialogue segment (collected using the images shown in Figure 1.2)
H: there is basically a cluster of four objects in the upper left, do you see that

R: yes

H: ok, so the one in the corner is a blue cup

R: I see there is a square, but fine, it is blue

H: alright, I will just go with that, so and then right under that is a yellow pepper

R: ok, I see apple but orangish yellow

H: ok, so that yellow pepper is named Brittany

R: uh, the bottom left of those four? Because I do see a yellow pepper in the upper right

H: the upper right of the four of them?

R: yes

H: ok, so that is basically the one to the right of the blue cup

R: yeah

H: that is actually an apple, it is green, I guess it has some amount of yellow on it, but

that is a green apple and it is named Ashley

This example demonstrates two important characteristics regarding referential communication under mismatched perceptual capabilities.

- First, conversation partners rely on both object-specific properties (e.g., object class, color) and spatial relations to describe objects in the environment. Spatial expressions include not only the binary relations (e.g., the one to the left of the blue cup), but also the group-based references [12, 13, 14, 15] (e.g., the upper right of the four of them).

- Second, because the shared perceptual basis is missing here, the partners make extra efforts
to refer and ground references. For example, the human-player go through step-by-step inst-
allments [16] to come to the targeted object. The robot-player often proactively provides
what he perceives from the environment. The human-player and the robot-player collab-
orate with each other through iterative presentation-acceptance phases as described in the
Contribution Model proposed in [17, 11].

These observations indicate that, since robots need to collaborate with human partners to estab-
lish a joint perceptual basis, referring expression generation (REG) becomes an equally important
problem in situated dialogue. Robots have much lower perceptual capabilities of the environment
than humans.

In order for a robot to effectively generate referential descriptions about the environment so
that its human partner can understand which objects are being referred to, the approach to REG in
situated dialogue should have at least the following features:

- should capture not only binary relations but also group-based relations.

- should go beyond traditional approaches that purely rely on one-shot description by a single
utterances. For example, it should incorporate the step-by-step collaborative dynamics from
the discourse as the conversation proceeds.

- should have the error-tolerance mechanism to make the referential communication successful.

There has been a tremendous amount of work on REG in the last two decades. REG has tradi-
tionally been formulated as a problem of generating a single noun phrase (possibly with multiple
modifiers and prepositional phrases) that can uniquely describe a target object among multiple ob-
jects [18, 19] so that addressees (i.e., humans) can correctly identify the intended object given this
expression.
Although extensively studied, most existing REG algorithms were developed and evaluated under the assumption that agents and humans have access to the same kind of domain information. For example, many experimental setups [20, 7, 21, 22] were developed based on a visual world for which the internal representation is assumed to be known and can be represented symbolically.

However, this assumption no longer holds in situated dialogue with robots. There are two important distinctions in situated dialogue:

- **First, the perfect knowledge of the environment is not available to the agent ahead of time.** The agent needs to automatically make inferences to connect recognized lower-level visual features with symbolic labels or descriptors. Both recognition and inference are error-prone and full of uncertainties.

- **Second, in situated dialogue the agent and the human have mismatched representations of the environment.** The agent needs to take this difference into consideration to identify the most reliable features for REG.

Given these two distinctions, two research questions pop up:

- How well state-of-the-art REG approaches are applicable under mismatched perceptual basis in situated dialogue?

- If state-of-the-art REG approaches are not applicable, how shall we develop REG approaches that are suitable for situated dialogue?

### 1.3 Contributions of this Dissertation

My thesis is to address the above two research questions. The main contributions of this thesis are:
A hypergraph approach to REG, which is an extension of a well-known graph-based approach [23] that has shown to be effective in previous work [24, 25]. We extended regular graph representation into hypergraph representation to account for group-based spatial relations that are important for visual descriptions. Unlike the previous regular graph approach, we incorporated uncertainties in vision perception into cost functions. Our empirical results have demonstrated that:

1. In order to address the agent’s limited perceptual capability in situated dialogue, REG algorithms will need to take into account uncertainties in perception and reasoning. (More specifically, the cost functions should be tied to the agent’s ability to perceive and infer the environment.)

2. Group-based information appears more reliable and thus should be modeled by an approach that deals with automated perception of spatially rich scenes.

3. Hypergraph-based approaches perform effectively under perfect perception. But with around 40% performance drop under imperfect perception, this significant performance degradation calls for new solutions to REG under situated dialogue.

Two computational collaborative models for REG: an episodic model and an installment model. Motivated by collaborative behaviors in human-human referential communication, our collaborative models, instead of generating a single referring expression to describe a target object as in the previous work, generate multiple small noun phrases that gradually lead to the target object with a goal to minimize the collaborative effort. Our empirical results have demonstrated that:

1. The weights of features which affect the decision of generating referring expressions can be learned through online interaction with participants.
2. The integration of human feedback in REG model improves the accuracy of the referential identification task. Our current model could serve as a baseline for further work that incorporates richer dialogue interaction strategies.

- Embodied collaborative models for REG, which take into account of embodiment and apply to physical interaction. In particular, these models seamlessly incorporate robot gesture behaviors (i.e., pointing to an object) and human’s gaze feedback (i.e., looking at a particular object) into the collaborative model for REG. Our empirical results have demonstrated that among different strategies incorporating robot gestures and human feedback, when robot gestures and human verbal feedback is incorporated, the new embodied collaborative model achieves over 28% absolute gains compared to the baseline collaborative model.

1.4 Outline

The remainder of this thesis is structured as follows:

- Chapter 2 describes earlier work related to this thesis, including previous work in REG categorized in several groups: the greedy search approaches, incremental approaches, graph-based approaches, plan-based approach, reinforcement learning based approaches and others. Some of them directly inspire our study in this thesis.

- Chapter 3 describes the hypergraph approach to REG, a extension to a previous well known graph-based approach. We show how to represent the perceived environment as a hypergraph, and how we incorporated uncertainties in vision perception into the hypergraph. We then formalize the REG as a subgraph isomorphism problem, and present the algorithm that search for the best solution for the problem. The evaluation of the hypergraph approach
under mis-matched perception basis is also presented. We conclude with the call for new approach to REG in situated dialogue.

- Chapter 4 describes the two collaborative models for REG, which do not follow the traditional paradigm of REG algorithm. Instead, the models are inspired by the behaviors observed in human-human referential communication. The evaluation and comparison with the hypergraph based approach are present under the mis-matched perception basis. We also discuss the importance of human feedback in collaborative referring process and possible future directions.

- Chapter 5 describes Embodied collaborative models for REG, which take into account of embodiment and apply to physical interaction. We shows how these models seamlessly incorporate robot gesture behaviors (i.e., pointing to an object) and human’s gaze feedback (i.e., looking at a particular object) into the collaborative model for REG. Our empirical results also demonstrated that, when robot gestures and human verbal feedback is incorporated, the new embodied collaborative model achieves over 28% absolute gains compared to the baseline collaborative model.

- Chapter 6 describes my future plan towards the situated human-robot dialogue.
Chapter 2

Related Work

Research in REG primarily focuses on the sub-task of content determination, which selects a set of properties that may be used to construct the final surface expression. This content determination task is generally affected by the Grices’s maxims [26], as shown below:

- **Quantity**

  1. Make your contribution as informative as is required.

  2. Do not make your contribution more informative than is required.

- **Quality**

  1. Do not say what you believe to be false.

  2. Do not say that for which you lack adequate evidence.

- **Relation**

  1. Be relevant.

- **Manner**

  1. Avoid obscurity of expression.

  2. Avoid ambiguity.

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

In practice, the content determination task for REG is generally optimized to meet two main different goals:

- **Human-likeness**: be identical to those a human would say in the same situation.

- **Distinction**: be unique to the intended referent with respect to its distractors.

With this in mind, next we will give a brief review of the REG algorithms.

2.1 Greedy Search Approaches

The first classical algorithm is the Full Brevity algorithm (Dale 1989). The algorithm (as shown in Figure 2.1) deploys a search over the set of all properties of the target referent to choose the smallest possible subset that can distinguishes the target from its all distractors. In other words, it aims to search for the shortest distinguishing referring expression for the intended object. In order to find such minimal descriptions, the algorithm first checks whether there exists a single attribute of the intended object that can rule out all the distractors. If this fails, the algorithm further checks all possible combinations of two attributes to see if this can uniquely identify the target, and so on. The algorithm attempts to meet principles from Grice’s conversational maxims [26]:

- **Maxim of Quantity** states that a referring expression must contain enough information to distinguish the target referent unambiguously from all distractors and enable the listener to resolve the reference. And a referring expression should not contain more information than necessary for identification, as that might result in unintended implication for the listener.
Let $L$ be the set of properties to be realised in our description; let $P$ be the set of properties known to be true of our intended referent $r$ (we assume that $P$ is non-empty); and let $C$ be the set of distractors (the contrast set). The initial conditions are thus as follows:

- $C = \{(all \text{ distractors})\}$;
- $P = \{(all \text{ properties true of } r)\}$;
- $L = \{\}$

In order to describe the intended referent $r$ with respect to the contrast set $C$, we do the following:

1. Check Success:
   - $\text{if } |C| = 0 \text{ then return } L$ as a distinguishing description
   - $\text{elseif } P = \emptyset \text{ then fail}$
   - $\text{else goto Step 2.}$

2. Choose Property:
   - $\textbf{for each } p_i \in P \textbf{ do: } C_i \leftarrow C \cap \{x| p_i(x)\}$
   - Chosen property is $p_j$, where $C_j$ is the smallest set.
   - $\text{goto Step 3.}$

3. Extend Description (wrt the chosen $p_j$):
   - $L \leftarrow L \cup \{p_j\}$
   - $C \leftarrow C_j$
   - $P \leftarrow P \setminus \{p_j\}$
   - $\text{goto Step 1.}$

Figure 2.1: An algorithm for the Greedy Heuristic: the Full Brevity algorithm

- **Maxim of Relation** specifies that a referring expression should be pertinent to the abilities of the listener by only including contents that the listener knows or can perceive easily.

Full brevity is a computationally intractable problem, it has been superseded by the more popular Incremental Algorithm. However, many ideas and concepts from Full Brevity algorithm. One of this is viewing a number of REG algorithms as search problem and can be a unified as in a search framework [27]. The underlying mechanism of REG approach is a search over the properties of the intended referent. A REG problem is then defined as a state consisting of the description $L$ constructed so far, the set of distractors $D$ that L might refer to, and the set of properties $Pr$ that have not been included in $L$ yet. In the initial state, $L$ is empty, $D$ contains all other objects in the
domain and $Pr$ contains all properties known to be true of $r$. In the goal state, $L$ contains some of the properties from $Pr$ and $D$ is empty, which means that $L$ only refers to $r$. In Bohnet and Dale’s model, the search proceeds from state to state by moving properties between $Pr$ and $L$ and adjusting $D$ accordingly. The search process and its outcome can be influenced by means of a cost function and different ways of queuing the properties in $P$ and expanding the search graph. This procedure can be be sketched as [28] general algorithm for search:

![Algorithm](image)

Figure 2.2: The algorithm that takes REG as a search problem

### 2.2 Incremental Approach

The Incremental Algorithm (IA) [18] which takes as an input a domain $O$ containing domain objects with their corresponding properties, and a preference list $P$ representing the order in which the attributes will be considered for selection. One particular object in $O$ is the target $r$ to be described by a description (i.e., a uniquely identifying set of properties), and the remaining objects are assumed to be distractors. The goal of the algorithm is to compute a list of attributes $L$ such that $L$ uniquely describes $r$ and no other distractor in $O$. Figure 2.1 show the IA algorithm.
The IA works by iterating over the list of preferred attributes P and adding to the description L under construction the attributes of the target object r that help ruling out at least one distractor. The algorithm succeeds when all the distractors have been ruled out (e.g., at the point in which L can uniquely identify r), it fails if all the properties have been processed and the distractor set is not empty.

Due to its computational efficiency and simplicity as well as its psycholinguistic plausibility, the IA has become the most implemented and built upon REG algorithm in the literature. In the
following I describe some of these extensions of the IA.

[29, 30] proposed an extension of the IA that takes into account the discourse salience of the target referent and produce referring expressions that contain binary relations to other objects. Theune and Krahmer’s approach works by assigning a salience score to all objects according to the topic distinction and Centering Theory [31]. They alter the success criterion of the algorithm and only let it stop when there is no distractor left that is as or more salient than the target referent.

[32] study the logical completeness of IA in terms of the Boolean operators of negation and disjunction. He then extended IA to be able to generate referring expressions that contain negated properties, such as ‘the apple that is not yellow’, and descriptions of sets of objects, such as ‘the orange books’, or descriptions which contains a logical disjunction of properties such as ‘the yellow apple and the orange book’. The algorithm proceeds in stages, if the atomic properties and shorter disjunctions could not sufficient to distinguish the target set. The algorithm will try to generate longer disjunctions of properties.

[33, 34] used the IA for the re-generation of referring expressions in newspaper text taken from the Penn Treebank. Their approach is different from other approaches to REG in that it is aimed at being useful for applications such as question-answering or summarization, and therefore takes text as input rather than a well-defined knowledge base. Their approach has several characteristics:

- Instead of processing words with properties at a semantic level, their approach works with words on a lexical level.

- They redefined the definition of discriminatory power (which they called the discriminating quotient) and incorporated it into the IA. The preference order in their approach is sorted not according to the psycholinguistic finding of human’s preference, but in a way that properties which are less similar to those of the distractors according to WordNet are considered before
those that are more similar to the distractor’s properties.

- They suggested a different approach to incorporate discourse salience in the IA. In their approach, it takes into account the salience of the distractor objects in the mechanism that determines the preference order: a property that distinguishes the target referent from a salient distractor is worth more than one that distinguishes it from one with relatively low salience.

- They extended the IA in a way that the preference order is dynamically adjusted at runtime, the properties are also sorted according to their discriminatory quotient at the run-time.

[35] used the IA for incremental generation of spatial referring expressions in situated dialog, they extended the IA in two directions:

- First, a model of object salience is integrated by modifying the condition under which a description is distinguishing:

- Second, instead of constructing a full relational scene model, they incrementally construct a series of reduced scene used by the algorithm.

### 2.3 Graph-based Approaches

Most early REG algorithms represent domain in a very simple type of knowledge representation: a knowledge base that contains the set of all objects in the domain, and each object has a set of properties that are true of it, and a set of spatial relations among objects. Sometimes this is varied by instead listing for each property which entities have this property. This knowledge base can be represented by a propositional database. But research in Knowledge base suggest that such way of knowledge modeling may not be wise in the long run. There are some researchers that try to
formalize REG with different mathematical and computational frameworks. These formalizations are different from traditional REG approaches in several ways:

- The representation of the domain information

- The representation of the semantic content of a referring expression

- The way to search for a distinguishing description

One notable algorithm is the graph-based approach [23], which represents the domain including the target referent and distractor objects as a labeled directed graph and the REG is re-formalized as the subgraph construction problem. In the domain, each object is represented as a node, the properties of an object are represented as self-looping arcs on the object node, and spatial relations between objects are represented as arcs between nodes.

The graph representation of a visual scene models each object of the scene as a vertex in the scene graph. Attributes of an object such as type, color, type or size are represented as self-looping arcs on the corresponding vertex. They are labeled with the attribute names and the values the object holds. Relations between objects, for example above or to the left of, are modeled as
Figure 2.5: The graph-based REG algorithm

arcs between the corresponding vertices. Figure 2.4) shows a sample scene graph containing two objects: a cup and a box.

Two different graphs play a role in the algorithm: a scene graph representing the domain, and a referring graph representing the content of a referring expression. To generate a distinguishing description, the graph-based algorithm searches for a subgraph of the scene graph that uniquely identifies the target referent, called a distinguishing graph. Starting with the subgraph only containing the vertex which represents the target referent, it performs a breadth-first search over the edges connected to the subgraph found so far. It searches the space exhaustively, but uses a Branch and Bound cost-based heuristic to effectively prune the search space. Informally, a subgraph refers to the target referent if and only if it can be ‘placed over’ the domain graph in such a way that the
subgraph vertex representing the target object can be ‘placed over’ the vertex of the target in the
domain graph, and each of the labelled edges in the subgraph can be ‘placed over’ a correspond-
ing edge in the domain graph with the same label and same direction. Furthermore, a subgraph
is distinguishing if and only if it can be ‘placed over’ exactly one vertex in the domain graph.
The informal notion of one graph being ‘placed over’ another corresponds to the mathematical
graph-theoretic concept of subgraph isomorphism.

Figure 2.6: Deictic gesture pointing to the scene [4, 5]

Figure 2.7: Deictic gesture graph of the scene [4, 5]
The graph-based approach generates referring expressions that can identify the target object. One natural extension of the graph-based approach is to generate multimodal expressions to identify the target object, e.g., pointing gesture together with referring expression. [4, 5] extended the graph-based framework to integrates pointing gestures with referring expressions. (Figure 2.6, 2.7 shows the basis idea). Their method is based on psycholinguistic study finding that people combine gestural and verbal information when referring. Their approach represents pointing gestures of different preciseness as self-loop arcs on the target referent in the domain graph. The more imprecise the pointing gesture, the more of the target’s closest neighbors are included in the gesture and therefore have the same pointing edge. The cost of pointing edges is determined by the size of the target referent and the distance that the pointing device has to travel to make the pointing gesture associated with it. The more imprecise the pointing gesture, the less effort is involved in bringing the pointing device into the correct position and therefore the cheaper the pointing gesture.

Based on the observation of the experiment that the verbal information in such multimodal referring expressions is often redundant when the pointing gesture is taken into consideration, [36] further proposed an adaptation of their algorithm to allow the generation of over-specified multimodal descriptions. They use a certainty score that represents the speaker’s estimate of how likely the listener is to misinterpret the referring expression under construction. The certainty score of a property is determined by a hierarchy over the domain attributes, whereby absolute attributes, such as color, have a higher certainty score than relative ones such as size. The certainty score of a pointing gesture is dependent on its preciseness and calculated in a similar way to the cost of pointing gestures. The certainty score of a referring expression is the sum of the certainty scores of the properties and pointing gestures contained in it. As long as the overall score is below a certain threshold, more properties have to be included, even if the description is already distinguishing from a logical point of view.
Another extension to graph-based approach is Conceptual Graphs-based generation of referring expressions [37]. In this work, the author used Conceptual Graphs as the representational formalism for referring expression generation. A Conceptual Graph consists of a bipartite graph, which specifies the factual knowledge about a domain, and a number of support hierarchies which capture ontological knowledge about the types of entities and relations that can be involved in the domain graphs. There are several advantages:

- First, conceptual graphs are firmly anchored in First-Order Logic, which allows tapping into existing ontology, and to perform automatic inference.

- Second, binary relations are handled in the same way as atomic properties, and relations of any higher order can be represented and treated in a natural way without complex adaptations of the knowledge representation.

### 2.4 Plan-based approaches

Planning approaches to REG typically define a set of logical constraints, capturing linguistic or contextual knowledge, and then use planning approaches to find one or several ways to meet these constraints.

[38] treated the processes of generating referring expressions and identifying their referents as a collaborative activity [16]. The collaborative process starts with one participant presenting an initial referring expression. The other participant would accept it, reject it, or postpone the judgment. If a presentation is not accepted, then either participant needs to refashion it. This refashioned expression is then judged again, and the process iterates until the current presentation is accepted. Based on this process Heeman developed a plan-based computational approach to collaborative REG, where referring is modeled as goal-directed behavior. Their approach - based
on Clark and Wilkes-Gibbs’s model [16] - is a computational mode of how dialogue participants collaborates in making a successful referring action.

[39] investigated the collaborating on references to objects that are not mutually known to the conversational participants (such as references to landmarks in direction-giving dialogues). Edmonds focuses on dialogue situations where an agent’s first attempt at describing a referent is judged insufficient by the recipient and the agents collaborate on expanding the description to provide further information, Edmonds views referent identification as a collaborative activity and formalized it within the planning framework.

A more recent approach to constraint satisfaction is [40], who present an approach to generating scripted dialogues. Their approach addresses particularly the case where several conflicting constraints need to be satisfied together. The challenge here is to balance both constraints against each other if it is not possible to satisfy them both optimally.

[41] presented an approach for context-sensitive referring expression generation, the approach is based on a theoretical model of salience grounded in functional and cognitive linguistics. The model takes into account two measures of salience, one focusing on speaker-related salience and one focusing on hearer-related salience. Given a set of candidate realizations for a referring expression, the best candidate can be chosen according to both salience measures.

[42] developed a plan-based model for response generation during collaborative planning. Their approach is based on a recursive Propose-Evaluate-Modify framework for modeling collaboration.

[43] present an approach based on Lexicalized Tree Adjoining Grammar, where the planner’s goal is to get a grammatically complete derivation tree, which is also satisfying a set of semantic and pragmatic constraints. To reach this goal, a surface realization needs to be found. All semantic and pragmatic constraints introduced by a surface form are directly encoded into the planner so
that they can be taken into account for choosing the best surface realization given a concept and context as well as the constraints specified by the realization itself. There are two advantages in this approach:

1. Any planning algorithm can be used which can make generation process very quick and efficient.

2. Sentence planning and realization are combined into a single process.

[44] extend this approach to situated generation where the goal of the planner is to generate non-ambiguous referring expressions in a virtual 3D environment. This goal can be achieved by making use of all the available linguistic and non-linguistic context. [45] develop a maximum entropy based model - which is an extension of the model - to deal with alternative surface realizations. Their model aims to find the surface realizations that a user will consider easiest to resolve.

2.5 Reinforcement Learning based Approaches

Reinforcement learning approaches optimizes a sequential decision making problems by mapping situations to actions with a goal to maximize a long-term reward. The main ingredients of a reinforcement learning approach contains:

- a set of states S that represents the agents knowledge of the previous and current environment.

- a set of actions A that represents the agents potential behaviors under a certain state.

- a state transition function T that specifies the next state of the agent for taking an action in a state.
• a reward function \( R \) that assigns a numeric value to the agent for taking an action in a state.

• a policy that specifies what actions to take in what states.

We refer to [10] for a comprehensive overview of the reinforcement learning approaches.

Reinforcement Learning has been successfully applied for adaptive Natural Language Generation[46, 47, 48, 49, 50, 51], where reinforcement Learning is used to automatically optimize the generation policy under uncertainty. We now review some of them.

[47] present an optimization approach in a tourist information domain for information presentation. Their system has two main choices to consider:

1. which information presentation strategy to choose among recommendation, comparison, summary, and any ordered combination of these strategies.

2. which attributes to include in the presentation of food quality, location, price range, service quality and cuisine type.

The main trade-off the learning system needs to optimize is to present as enough information to users as possible to give them the feeling of having a good overview of the options, but at the same time utterances need to be short and understandable. The learning system was trained using a reward function that optimizes the sentence length, the user reaction and the number of database hits presented. For the user reaction, the authors have hand-coded numeric values associated with user’s each possible reaction (e.g., user selects an item, user adds further constraints to the search, etc.). The reward function and the n-gram-based simulated environment in which the agent was trained were estimated from data collected in a Wizard-of-Oz setting. Learning was performed using the SHARSA algorithm. In a simulation-based evaluation, the learned policy were shown to outperform a supervised learning baseline.
[46] present a approach for referring expression generation in troubleshooting dialogue domains, where their approach is able to tailor its generation actions according the users individual prior knowledge of the domain. The setting in the dialogue domain is about a system helping a user in setting up a broadband connection. This dialogue domain enables the many choices for referring to technical objects. The choices range from using technical jargon, such as saying the broadband filter, to using referential descriptions, such as the big black box. An important and interesting characteristic here is that users knowledge will dynamically be updated during the interaction due to a learning effect. Their system would then needs to adapt to a constantly changing user model. The crucial trade-off system encounters is therefore to adapt maximally to the users individual information need. For example, the system should try not to use terms that the user does not understand, it should also not engage in unnecessarily long descriptions. Training of the learning agent was done using the on-policy SHARSA algorithm [10] against a user simulation that was estimated from data collected in a Wizard-of-Oz setting. The user’s responses are simulated based on the system’s last dialogue act, the users prior knowledge of the domain as well as previous clarification requests.

[52] use reinforcement learning to learn the policy for generation of temporal expressions. They train a learning agent in a simulated environment. The simulated environment was estimated from human’s ratings of temporal expressions. They use a hand-crafted reward function that assesses the user preference, the likelihood that the user requests clarification and the length of the description. They showed that users interacting with the learned policy have a significantly higher subjective task success rate and user satisfaction score than other baseline approaches.

[51] presented a Hierarchical reinforcement learning models for task-oriented Natural language generation in the navigation instruction domain. the generation of a navigation instruction is a mix of navigation strategies generation and referring expression generation. Their approach jointly
optimize the navigation strategies (e.g., high level navigation such as "go down the hallway", low level navigation such as "turn right") and referring expression generation (e.g., mention color or spatial relation of the referent) in direction-Giving dialogues within virtual 3D worlds [8].

2.6 Other Approaches

[53] proposed a computational model for REG. The model treated the REG task as a constrained search problem, which takes into account the syntax, lexical choice and content selection as the search constraint. The model was evaluated using synthesized images.

[21] treat language generation as a game-theoretic model in which a set of semantic and pragmatic constraints need to be satisfied. They address the task of generating spatial descriptions so as to allow the hearer to identify a target object as easily as possible. They present results showing that a generation model that takes the pragmatic effects of its utterances on the hearer into account significantly outperforms a model that does not.

[13] formalized a referring expression as a sequence of groups which reflects the subject’s narrowing down strategy in identifying the referent object. Given a domain and a referent object, the method constructs a sequence of groups starting from the largest group containing the referent, and recursively narrowing down the group until only the referent is identified. The authors extended this approach [14] to handle more referring strategies, e.g., descriptions with only the referent object or in combination with binary relations between objects. The entire sequence is then rendered linguistically.

[54] present a minimally supervised learning approach to the generation of route instructions in outdoor environments. They use the Expectation-Maximizing algorithm to align geographical route representations with corresponding linguistic realizations that were taken from an anno-
tated corpus of human data. They argue that route instruction generation can be a highly context-dependent task given the adaptations that an NLG system is required to make to specific spatial configurations and environments also including diverse spatial objects that may be present.

[55] describe a approach to generate text descriptions of web resources, where the context for generation is represented in a declarative way through the construction of policies. The context in their model is defined by permissions, prohibitions and obligations regarding particular users or resources. In their text descriptions generation domain, different users of the platform may have different permissions to access information that are stored on the platform, and this need to be taken into account by the NLG system.

[56] introduced a chart-based overgenerate-and-rank approach to REG which separates the representational form of the knowledge base from that of the referring expression being built. This allows the algorithm to logically infer information that is not represented explicitly in the knowledge base, rather than adding such information to the knowledge representation, as was done in extensions to the IA and the graph-based framework described above.

The algorithm first builds up all possible combinations of properties to describe all objects and sets of objects in the domain. It does this by recursively combining basic descriptions using logical connectives and relations between objects, always keeping track of the extension sets of the logical forms constituting the descriptions. The combinations are stored in a chart to facilitate reuse of intermediate results. The chart is pruned by the requirement that every combination of properties produced has to be realizable linguistically, making this one of the rare approaches that integrate surface realization with content selection. To guide the search for ‘optimal’ referring expressions, Varges suggested a number of constraints that either apply during the chart-building process or filter out unwanted solutions afterwards. [57] extended his algorithm to be able to produce quantified expressions, which is, as they show, impossible in approaches such as the IA.
and the graph-based approach.

Description Logics constitute yet another approach to the generation of referring expressions that is based on viewing the semantics of a referring expression as a logical formula. Description Logics are a well-established First-Order Logic formalism and have been used to combine basic formulas (or descriptions) via logical connectives to generate descriptions for all objects and sets of objects in parallel [58].

[59] present an approach for route instruction generation in urban environments which takes different properties of the user and the environment into account. In their approach, route instructions are adapt to different degree of prior knowledge that users have about the navigation environment (e.g., their familiarity with the area). Moreover, their approach when generates route instructions, take into account the saliency of geographical properties in the environment.

[60] present an approach for generating referring expressions for an interaction task in a 3D virtual environment. Their algorithm is based on a decision tree classifier using features of the environment. The features include the users viewing angle, their distance from the target referent, number of the target referent’s distractors, as well as its visibility and dialogue history. They present results from a human rating study in which the referring expressions generated by their algorithm were rated better.

Recently, there is an increasing interest in generating multimodal reference. For example, based on the findings that deictic gesture is useful in referential communication [61, 62, 63, 64] explored the role of salience in the generation of referring expressions with gesture. [65, 66] combined deictic gesture and speech to generate a referring act. [5] extended a graph-based non-collaborative REG model to generate multimodal reference.

Motivated by psycholinguistic findings that eye gaze is tightly linked to human language processing in both language interpretation and production [67], [68] developed an interactive REG
system that tracks the hearer’s eye gaze to monitor hearer’s interpretation of the referring expressions the system generates. Their REG system is embodied in nature, but the system was built on a virtual environment where the agent and the human are assumed to have the same representation of the shared environment. In addition, their system only generates verbal descriptions, not multimodal referring expressions.

More recently, work in REG has identified the importance of modeling the variation observed in referring expressions generated by human, instead of generating a single referring expression for the intended referent, [69, 70] proposed approaches that can generate a distribution of referring expressions for a referent object or a set of objects.

An earlier work by [71] has looked into the problem of mismatched knowledge between conversation partners for REG. The approach is a direct extension of the incremental algorithm [18]. However, this work only provides a proof of concept example to illustrate the idea. No empirical evaluation was given.

All these previous works have motivated our present investigation. We are interested in REG under mismatched perceptual basis between conversation partners, where the agent has imperfect perception and knowledge of the shared environment. In particular, we first took a well-studied graph-based approach [23] and extended it to incorporate group spatial relations and uncertainties associated with automated perception of the environment. The reason we chose a graph-based approach is that graph representations are widely used in the fields of computer vision (CV) and pattern recognition to represent spatially rich scenes. Nevertheless, the findings from this investigation provide insight to other approaches. Motivated by collaborative behaviors observed in human referential communication, we then develop two collaborative models - an episodic model and an installment model - for REG. In both models, instead of generating a single referring expression to describe a target object as in the previous work, it generates multiple small expressions.
that lead to the target object with a goal to minimize the collaborative effort. In particular, our installment model incorporates human feedback in a reinforcement learning framework to learn the optimal generation strategies and obtains better performance. We have also developed embodied collaborative models for REG. These models seamlessly incorporate robot gesture behaviors (i.e., pointing to an object) and human’s gaze feedback (i.e., looking at a particular object) into the collaborative model for REG. We examined different strategies incorporating robot gestures and human feedback. Our empirical results have shown that when robot gestures and human verbal feedback is incorporated, the new collaborative model achieves over 28% absolute gains compared to the baseline collaborative model. This paper discusses the opportunities and challenges brought by modeling embodiment in collaborative referential communication in human-robot interaction.

2.7 Evaluation of REG approaches

The evaluation of REG evaluation could be task-based where human rate the usefulness of generated referring expressions for a particular task (e.g., identifying objects), or corpus-based, where generated expressions are compared to human-produced expressions in a given data set. As more and more suitable data sets are available now, corpus-based evaluation becomes the dominant paradigm. we next introduce the currently available data set, and discuss what metrics are used to evaluate the REG algorithms on those data set.

2.7.1 Corpora for REG evaluation

2.7.1.1 The TUNA corpus

The TUNA corpus [6] is a data set which was collected via a web-based experiment. During the experiment both singular and plural descriptions for objects were gathered by showing partici-
Figure 2.8: An example scene from the TUNA corpus [6]. An example description for the target object: ‘the black chair on the right’.

pants one or two targets. Targets were always displayed together with distractors, and the resulting domain objects were randomly positioned in a 3 x 5 grid. An example scene are shown in Figure 2.8. The TUNA corpus contains two different domains: a furniture and a people domain. The furniture domain has pictures of furniture and household items, and the items in the domain were manipulated so that besides type attribute (e.g., desk, chair, fan), color, orientation and size are also systematically varied. In the people domain, the number of possible attributes and values is much larger. This domain contains a set of black and white photographs of people. Properties of these photographs include gender, head orientation, shirt suit and tie, beard, age, glasses, hair and locations. The TUNA data set has formed the basis of three shared REG challenges.

2.7.1.2 The GRE3D3 corpus

Psycholinguistic findings suggesting that the use of spatial relations causes a higher cognitive load for both speaker and listener. So there exists a prevalent hypothesis that spatial relations should only be used as a last resort. Many REG algorithms don’t attempt to use spatial relations between
the target referent and landmark objects in the descriptions. The GRE3D3 and corpus [7] were created to explore the usefulness of spatial relations in REG. The corpus is based on visual stimuli of simple 3D scenes containing a small number of geometric objects as shown in Figure 2.9. They were collected by directing participants to a website, which allowed the collection of a very large number of samples. The GRE3D3 contains 720 referring expression. The authors created a second corpus, GRE3D7, which contains 4480 referring expressions and is by far the largest existing collection of context-independent distinguishing descriptions.

2.7.1.3 the GIVE-2 Corpus

Interpreting and generating natural-language instructions in a situated environment have received significant attention in the past few years. Most recently, the Challenge on Generating Instructions in Virtual Environments (GIVE) [72] has attracted considerable interest in the Natural Language Generation community. GIVE is a shared task in which NLG systems must generate real-time instructions that guide a user in a virtual world. It offers novel possibilities of exploring the linguistic and non-linguistic issues involving situated NLG, while supporting a solid approach NLG system evaluation. The GIVE-2 corpus [8] is a multilingual corpus of human instruction giving in virtual
environments, It is designed to support the development of NLG systems for the GIVE Challenge.

The corpus consists of 63 American English and 45 German written discourses in which one subject guided another in a treasure hunting task in the spirit of the GIVE-2 virtual worlds as shown in Figure 2.10.

This corpus exhibits varied instruction-giving behavior, and can thus serve both as a source of data and as point of comparison for for natural language generation systems.

### 2.7.2 Evaluation Metrics

The emergence of corpora discussed earlier has greatly facilitated the empirical evaluation of REG algorithms. We now describe how corpora can be used for evaluation purposes.

#### 2.7.2.1 Human-likeness Metrics

Human-generated language is by definition natural, and also generally understandable. How to compare human references with those produced by a REG algorithm? One way is looking for
measures that compute the overlap at the level of properties, one source of inspiration may come from the kind of measures researchers in biology and information retrieval have been using (van Rijsbergen 1979).

One of these is the Dice (1945) coefficient, which was originally proposed as a measure of ecologic association between species, and applied to REG by Gatt et al. (2007). The Dice coefficient is computed by scaling the number of properties (that is: attribute-value pairs) that two descriptions have in common, by the size of the two sets combined:

\[
dice(A, B) = \frac{2 \times |A \cap B|}{|A| + |B|},
\]

The Dice measure ranges from 0 (no agreement, i.e., no properties shared between A and B) to 1 (complete agreement; A and B share all properties). It is a simple and straightforward measure for overlap, but it does have at least two disadvantages. First, it assumes that all properties are independent and that all are equally different from each other. Suppose a human participant referred to \(o_1\) in our example domain as the red apple on the left, and consider the following two references produced by a REG algorithm: the red apple and the apple on the left. Both omit one property from the human reference and thus have the same dice score. But only the former is distinguishing; the latter is not. This problem could be solved by adopting a binary weighted version of the Dice metric which would multiply the Dice score with 1 for a distinguishing description and with 0 for a non-distinguishing one.

Another metric is MASI (Measuring Agreement on Set-valued Items) [73]:

\[
MASI(A, B) = \lambda \times \frac{|A \cap B|}{|A \cup B|},
\]

The weighting function \(\theta\) biases the score in favor of similarity where one set (apple) is a subset
of the other (fruit):

\[
\begin{align*}
\lambda &= \begin{cases} 
0 & \text{if } A \cap B = \emptyset \\
1 & \text{if } A = B \\
2/3 & \text{if } A \subset B \text{ or } B \subset A \\
1/3 & \text{otherwise.}
\end{cases}
\end{align*}
\]

### 2.7.2.2 Identification Accuracy

Besides human-likeness, it is also important to explore to what extent the generated expressions help addressees identify a referent target. This can be studied experimentally in several different ways. For example [74] first show a generated description to participants. After the participants read the description, a visual scene appears and the participants are asked to choose the intended referent. This experiment gives rise to three extrinsic evaluation metrics: the reading time, the identification time and the accuracy, defined as the percentage of generated referring expressions where the referents are correctly identified.
Chapter 3

Graph-based Approach for REG

This chapter focuses on graph-based approach for REG in traditional paradigm.

3.1 Introduction

Towards mediating a shared perceptual basis in situated dialogue, our previous work [3] has conducted experiments to study referential communication between partners with mismatched perceptual capabilities. We simulated mismatched capabilities by making an original scene (Figure 3.1(a)) available to a director (simulating higher perceptual calibre) and a corresponding impoverished scene (Figure 3.1(b)) available to a matcher (simulating lowered perceptual calibre). The impoverished scene is created by re-rendering automated recognition results of the original scene by a CV algorithm. An example of the original scene and an impoverished scene is shown in Figure 3.1. Using this setup, the director and the matcher were instructed to collaborate with each other on some naming games. Through these games, they collected data on how partners with mismatched perceptual capabilities collaborate to ground their referential communication.

The setup in [3] is intended to simulate situated dialogue between a human (like the director) and a robot (like the matcher). The robot has a significantly lowered ability in perception and rea-

1The majority of this chapter was published in the following paper: Rui Fang, Changsong Liu, Lanbo She, Joyce Y. Chai. Towards Situated Dialogue: Revisiting Referring Expression Generation. Conference on Empirical Methods in Natural Language Processing (EMNLP), Seattle, USA, 2013.
soning. The robot’s internal representation of the shared world will be much like the impoverished scene which contains many recognition errors.

We have pointed out earlier that there are two important distinctions in situated dialogue:

- **First, the perfect knowledge of the environment is not available to the agent ahead of time.** The agent needs to automatically make inferences to connect recognized lower-level visual features with symbolic labels or descriptors. Both recognition and inference are error-prone and full of uncertainties.

- **Second, in situated dialogue the agent and the human have mismatched representations of the environment.** The agent needs to take this difference into consideration to identify the most reliable features for REG.

Given these two distinctions, it is not clear whether state-of-the-art REG approaches are applicable under mismatched perceptual basis in situated dialogue. In this chapter, we first extend a state-of-the-art REG algorithm with hypergraph-based representations and illustrate how uncertainties from automated perception can be incorporated. We then describe an empirical study using Amazon Mechanical Turk for evaluating generated referring expressions. Finally we present evaluation results and discuss potential future directions.

### 3.2 Hyper-graph based REG

#### 3.2.1 Motivation

The data from [3, 75] shows that different strategies were used by conversation partners to produce referential descriptions. Besides directly describing attributes or binary relations with a relatum, they often use group-based descriptions (e.g., *a cluster of four objects on the right*). This is mainly
due to the fact that some objects are simply not recognizable to the matcher. Binary spatial relationships sometimes are difficult to describe the target object, so the matcher must resort to group information to distinguish the target object from the rest of the objects. For example, suppose the matcher needs to describe the target object 5 in Figure 3.1(b), he/she may have to start by indicating the group of three objects at the bottom and then specify the relationship (i.e., top) of the target object within this group.

The importance of group descriptions has been shown not only here, but also in previous works on REG [13, 14, 15]. While the original graph-based approach can effectively represent attributes and binary relations between objects [23], it is insufficient to capture within-group or between-group relations. Therefore, to address the low perceptual capabilities of artificial agents, we introduce hypergraphs to represent the shared environment. Our approach has two unique characteristics compared to previous graph-based approaches:

- A hypergraph representation is more general than a regular graph. Besides attributes and binary relations, it can also represent group-based relations.
Unlike previous work, here the generation of hypergraphs are completely driven by automated perception of the environment. This is done by incorporating uncertainties in perception and reasoning into cost functions associated with graphs.

Next we give a detailed account on hypergraph representation, cost functions incorporating uncertainties, and the search algorithm for REG.

### 3.2.2 Hypergraph Representation of the Perceived Scenes

A directed hypergraph \( G \) is a tuple of the form: \( G = \langle X, A \rangle \), in which

\[
X = \{x_m\}
\]

\[
A = \{a_i = (t_i, h_i) \mid t_i \subseteq X, h_i \subseteq X\}
\]

Similar to regular graphs, a hypergraph consists of a set of nodes \( X \) and a set of arcs \( A \). However, different from regular graphs, each arc in \( A \) is considered as a hyperarc in the sense that it can capture relations between any two subsets of nodes: a tail \( (t_i) \) and a head \( (h_i) \). Therefore, a hypergraph is a generalization of a regular graph. It becomes a regular graph if the cardinalities of both the tail and the head are restricted to one for all hyperarcs. While regular graphs are commonly used to represent binary relations between two nodes, hypergraphs provide a more general representation for n-ary relations among multiple nodes.

We use hypergraphs to represent the agent’s perceived physical environment (also called scene hypergraphs). More specifically, each perceived object is represented by a node in the graph. Each perceived visual attribute of an object (e.g., color, size, type information) or a group of objects (e.g., number of objects in the group, location) is captured by a self-looping hyperarc. Hyperarcs are also used to capture the spatial relations between any two subsets of nodes, whether it is a
relation between two objects, or between two groups of objects, or between one or more objects within a group of objects.

For example, Figure 3.2 shows a hypergraph created for part of the impoverished scene shown in Figure 3.1(b) (i.e., the upper right corner including objects 7, 8, 9, 11, and 13). One important characteristic is that, because the graph is created based on an automated vision recognition system, the values of an attribute or a relation in the hypergraph are numeric (except for the type attribute). For example, the value of the color attribute is the RGB distribution extracted from the corresponding visual object, the value of the size attribute is the width and height of the bounding box and the value of the location attribute is a function of spatial coordinates. These numerical features will be further converted to symbolic labels with certain confidence scores (described later in Section 3.2.4.2).

### 3.2.3 Hypergraph Pruning

The perceived visual scene can be represented as a complete hypergraph, in which any pair of two subsets of nodes are connected by a hyperarc. However, such a complete hypergraph is not only computationally inefficient but also unnecessary. Instead of keeping all possible n-ary relations (i.e., hyperarcs), we only retain those relations that are likely used by humans to produce referring expressions, based on two heuristics.

The first heuristic is based on perceptual principles, also called the Gestalt Laws of perception [77], which describe how people perceive groups and objects in visual information. The word ‘Gestalt’ means whole or complete in German. The laws of grouping are shown below.

- **Law of Proximity.** The law of proximity states that objects that lie close together are often perceived as groups. Even if their shapes, sizes, and object types are different, they will
be perceived as a group if they are close together. And objects that are far apart are often perceived as separate objects.

- **Law of Similarity.** The law of similarity states that similar objects are likely to be perceptually grouped together. This similarity can occur in the form of shape, size, color, shading or other qualities. Objects of similar shape, size or color are more likely to form groups than objects differing along these dimensions.

- **Law of Closure.** The law of closure states that human vision perceive objects such as shapes, letters, pictures as a whole even when they are incomplete or partially occluded by other objects. For example, when parts of a shape’s border are missing, human perception tends to see the shape as completely and thus fill in the missing visual gap.

- **Law of Symmetry.** The law of symmetry states that the human vision perceives objects as being symmetrical and forming around a center point. Therefore, when two symmetrical parts are unconnected the human vision tend to perceptually connect them to form a coherent shape.

- **Law of Common Fate.** The law of common fate states that human vision perceive the moving elements as a whole when visual elements are seen moving in the same direction at the same rate.

- **Law of Continuity.** The law of continuity states that objects tended to be grouped together if they form a line or follow an established direction. When there is an intersection between objects, human vision tend to perceive each object as a single object, even the objects are overlapped with each other.
• **Law of Good Gestalt.** The law of good gestalt states that objects tend to be perceptually grouped together if they form a pattern that is simple, regular, symmetrical and in order.

• **Law of Past Experience.** The law of Past Experience states that objects tend to be grouped together if they were observed together in the past. If objects tend to be observed within small temporal intervals, the objects are more likely to be grouped together.

• **Law of Figure and Ground.** The law of figure and ground states that if human vision’s attention is on part of a scene, the remaining vision information tends to be ignored and becomes part of the background.

Among these laws, two well known principles of perceptual grouping are proximity and similarity. Based on these two principles, previous works have developed different algorithms for perceptual grouping [78, 79]. In our investigation, we adopted Gatt’s algorithm [79], which has shown to be more accurate for spatial grouping. Given the results from spatial grouping, we only retain hyper-arcs that represent spatial relations between two objects, between two perceived groups, between one object and a perceived group, or between one object and the group it belongs to.

The second heuristic is based on the observation that, given a certain orientation, people tend to use a relatum that is closer to the referent than more distant relata. In other words, it is less likely to refer to an object relative to a distant relatum when there is a closer relatum. For example, when referring to the stapler (object 9 in Figure 3.1(a)), it is more likely to use “the stapler above the battery” than “the stapler above the cellphone”. Based on this observation, we prune the hypergraphs by only retaining hyperarcs between an object and their closest relata for each possible orientation.

Figure 3.2 shows the resulting hypergraph for representing a subset of objects (7, 8, 9, 11, and 13) in Figure 3.1(a).
3.2.4 Symbolic Descriptors for Attributes

As mentioned earlier, the values of attributes of objects and their relations are numerical in nature. In order for the agent to generate natural language descriptions, the first step is to assign symbolic labels or descriptors to those attributes and relations. Next we describe how we use a lexicon with grounded semantics in this process.
3.2.4.1 Lexicon with Grounded Semantics

Grounded semantics provides a bridge to connect symbolic labels or words with lower level visual features [80]. Previous work has developed various approaches for grounded semantics mainly for the reference resolution task, i.e., identifying visual objects in the environment given language descriptions [81, 82, 83, 84, 3]. For the referring expression generation task here, we also need a lexicon with grounded semantics.

In our lexicon, the semantics of each category of words is defined by a set of semantic grounding functions that are parameterized on visual features. For example, for the color category it is defined as a multivariate Gaussian distribution based on the RGB distribution. Specific words such as green, red, or blue have different means and co-variances as the following:

\[
\begin{align*}
\text{color : red} &= f_r(\vec{v}_{\text{color}}) = N(\vec{v}_{\text{color}} | \mu_1, \Sigma_1) \\
\text{color : green} &= f_g(\vec{v}_{\text{color}}) = N(\vec{v}_{\text{color}} | \mu_2, \Sigma_2) \\
\text{color : blue} &= f_b(\vec{v}_{\text{color}}) = N(\vec{v}_{\text{color}} | \mu_3, \Sigma_3)
\end{align*}
\]

The above functions define how likely a set of recognized visual features (i.e., \(\vec{v}_{\text{color}}\)) describing the color dimensions (i.e., RGB distribution) is to match the color terms red, green, and blue.

For the spatial relation terms such as above, below, left, right, the semantic grounding functions take both vertical and horizontal coordinates of two objects, as follows \(^2\):

\[
\begin{align*}
\text{spatialRel : above(a, b)} &= f_{\text{above}}(\vec{v}_{\text{a loc}}, \vec{v}_{\text{b loc}}) \\
&= \begin{cases} 
1 - \frac{|x_a - x_b|}{400} & \text{if } y_a < y_b; \\
0 & \text{otherwise.}
\end{cases}
\end{align*}
\]

\(^2\)The size of the overall scene is 800x800.
\[ \text{spatialRel}: \text{left}(a,b) = f_{\text{left}}(\vec{v}_{\text{aloc}}, \vec{v}_{\text{bloc}}) = \begin{cases} 
1 - \frac{|y_a - y_b|}{400} & \text{if } x_a < x_b; \\
0 & \text{otherwise}. 
\end{cases} \]

Using the above convention, we have defined semantic grounding functions for size category words (e.g., small and big) and absolute position words (e.g., top, below, left, and right). In addition, we use object recognition models [85] to define class type category words such as apple and orange used in our domain.

### 3.2.4.2 Attribute Descriptors and Cost Functions

Given the lexicon with grounded semantics as described above, the numerical attributes captured in the scene hypergraph can be converted to symbolic descriptors. For each attribute (e.g., color) or relation, the corresponding visual feature vector (i.e., \(\vec{v}_{\text{color}}\)) is plugged into the semantic grounding functions for the corresponding category of words. The word that best describes the attribute is chosen as the descriptor for that attribute. For example, given an RGB color distribution \(\vec{v}_{\text{color}}\), we can find the color descriptor as follows:

\[ \text{color}: \quad \w^* = \arg \max_{\text{red}, \text{green}, \text{blue}} f_w(\vec{v}_{\text{color}}), \]

For each attribute or relation, we can find a best descriptor in this manner. In addition, we also obtain a numerical value (returned from the semantic grounding functions) that measures how well this descriptor describes the corresponding visual features. Intuitively, one would choose a descriptor that closely matches the visual features. Based on this intuition, we define the cost for
each attribute $A$ as the following:

$$cost(A) = 1 - f_{w^*}(v_A)$$

where $w^*$ is the best descriptor for the attribute.

Given an attribute, the better the descriptor matches the extracted visual features, the lower the cost of the corresponding hyperarc.

### 3.2.5 Graph Matching for REG

Now the hypergraph representing the perceived environment has symbolic descriptors for its attributes and relations together with corresponding costs. Given this representation, REG can be formulated as a graph matching algorithm similar to that described in [23].

We use the same Branch and Bound algorithm described in [23]. Figure 3.3 shows our hypergraph based referring expressions generation approach.

The cost of a hypothesis hypergraph $H$ is denoted as $cost(H)$. It is defined as:

$$cost(H) = \sum_{x \in X} cost(x) + \sum_{a \in A} cost(a)$$

The cost function is monotonic. This implies that extending a hypergraph $H$ with an hyperarc $a$ can only result in a more expensive hypergraph, that is, $cost(H + a) > cost(H)$. The monotonicity of the cost function helps reduce the searching space.

The distractor set contains every node $d$ in $G$ that is not the target node $v$, and can “refer to” $v$ in $H$. By “refer to”, we mean, $\exists G' \subset G$, and $H \cong G'$, the bijection $b : X' \to X$ between nodes of $H$ and $G'$, such that $b(d) = v$. 

48
In this approach, a hypothesis hypergraph (starting with one node representing the target object) is gradually expanded by adding in a least cost hyperarc from the scene hypergraph. At each expansion, the hypothesis graph is matched against the scene hypergraph to decide whether it matches any nodes other than the target node in the scene hypergraph. The expansion stops if the hypothesis graph does not cover any other nodes except for the target node. At this point, the hypothesis graph captures all the content (e.g., attributes and relations) required to uniquely
describe the target object. We then apply a set of simple generation templates to generate the
surface form of referring expressions based on the hypothesis graph.

3.3 Empirical Evaluations

3.3.1 Evaluation Setup

3.3.1.1 Mismatch Views

To simulate the Mismatched perception basis between human and an artificial agent, we applied
standard computer vision algorithms to process the original image and generate the impover-
ished representation of the same scene. This procedure is illustrated in Figure 3.4. To create
the impoverished image, we first used the OTSU algorithm [86] to separate foreground objects
from the background. Then each segmented object was fed into a feature extraction routine that
computed a set of region-based and contour-based shape features of the object [85]. The fea-
ture vector of the object was then compared with all the known objects in a knowledge base.
The object was recognized as the class of its nearest neighbor in the knowledge base. After this
segmentation → featureextraction → recognition pipeline, the final outcome was then dis-
played as an abstract illustration in the impoverished image. For instance, if an object was rec-
ognized as a pear, an abstract illustration of pear was displayed in the impoverished image at the
position of the original object. The color of the illustration was set to the average color of the
pixels of the original object, and the height and width were set according to the bounding box of
the original object.
Figure 3.4: The procedure of generating the impoverished image from the original image

3.3.1.2 Experiment Setup using Amazon Mechanical Turk

To evaluate the performance of this hypergraph-based approach to REG, we conducted a comparative study using crowd-sourcing. More specifically, we created 48 different scenes similar to that in Figure 3.1(a). Each scene has 13 objects on average and there are 621 objects in total. For each of these scenes, we applied the above impoverished scene generation procedure and generated scene hypergraphs as described in Section 3.2.2. We then use different generation strategies (varied in terms of graph representations and cost functions, to be explained in Section 3.3.2) to automatically generate referring expressions to refer to each object.

To evaluate the quality of these generated referring expressions, we applied Amazon Mechanical Turk to solicit feedback from the crowd. Through an interface as shown in Figure 3.5, we

---

3To control the quality of crowdsourcing, we recruited participants based on the following criteria: Participants’ locations are limited to the United States. Approval rate for each participant’s previous work is greater than or equal to 95%, and the number of each participant’s previous
Figure 3.5: a web-based interface shown to the Mechanical turk workers displayed an original scene and generated referring expressions (from different generation strategies) in a random order. We asked each turkers to select the object in the scene that he/she believed was the one referred to by the shown referring expression (i.e., reference identification task). Each referring expression received three votes from the crowd. In total, 217 turkers participated in our experiment.

3.3.2 Generation Strategies

We applied a set of different strategies to generate referring expressions for each object. The variations lie in two dimensions:

- Graph representations: using a hypergraph to represent the perceived scene as described in Section 3.2.2 versus using a regular graph as introduced in [23]; approved work is greater than or equal to 1000.
• Cost functions for attributes and relations: cost functions that have been used in previous works [87, 88] and cost functions that incorporate uncertainties of perception as described in Section 3.2.4.2.

Cost functions play an important role in graph-based approaches [23]. Previous works have examined different types of cost functions [87, 88, 89]. We adopted some commonly used cost functions from previous work together with the cost functions defined here. In particular, we experimented with the following different cost functions:

• **Simple Cost**: The costs for all hyperarcs are set to 1. With this cost function, the graph-based algorithm resembles the Full Brevity algorithm of Dale [90] in that a shortest distinguishing description is preferred.

• **Absolute Preferred**: The costs for hyperarcs representing absolute attributes (e.g., type, color, and position) are set to 1. The costs for relative attributes (e.g., size) and relations are set to 2. This cost function mimics human’s preference for absolute attributes over relative ones [18].

• **Relative Preferred**: The costs for hyperarcs representing absolute attributes are set to 2 and for relative attributes and relations are set to 1. This cost function has been applied previously to emphasize the importance of spatial relations in REG [7].

• **Uncertainty Based**: The costs for all hyperarcs are defined by incorporating uncertainties from perception as described in Section 3.2.4.2.

• **Uncertainty Relative Preferred**: To emphasize the importance of spatial relations as demonstrated in situated interaction [83, 35], the costs for hyperarcs representing relative attributes
and relations are divided by 3. This cost function will allow the algorithm to prefer spatial relations through the reduced cost.

Note that we only tested a few (not all) commonly used cost functions proposed by previous work [23, 87, 88, 89]. For example, we did not include the stochastic cost function which is defined based on the frequencies of attribute selection from the training data [23]. On the one hand, we did not have a large set of human descriptions of the impoverished scene to learn the stochastic cost. On the other hand, it is not clear whether human strategies of describing the impoverished scene should be used to represent optimal strategies for the robot. Nevertheless, the above different cost functions will allow us to evaluate whether incorporating perceptual uncertainties will make a difference in the REG performance.

3.3.3 Evaluation Results

As mentioned earlier, each generated referring expression received three independent votes regarding its referent from the crowd. The referent with the most votes is taken as the predicted referent and is used for evaluation. If all three votes are different, then by default, it is deemed that the referent is not correctly identified for that expression. We use the accuracy of the referential identification task (i.e., the percentage of generated referring expressions where the referents are correctly identified) as the metric to evaluate different generation strategies illustrated in Section 3.3.2.

3.3.3.1 The Role of Cost Functions

Table 3.1 shows the results based on different cost functions and different graph representations. There are several observations.

- First, when the agent does not have perfect knowledge of the environment and has to auto-
automatically infer the environment as in our setting here, cost functions based on uncertainties of perception lead to better results. This occurs for both regular graphs and hypergraphs. This result is not surprising and indicates that cost functions should be tied to the agent’s ability to perceive and infer the environment. The uncertainty based cost functions allow the agent to prefer reliable attributes or relations.

- Second, consistent with previous work [7], we observed the importance of spatial relations. Especially when the perceived world is full of uncertainties, spatial relations tend to be more reliable. In particular, as shown in Table 3.1, using hypergraphs enables generating group-based relations and results in significantly better performance (45.2%) compared to regular graphs (36.7%) ($p = 0.002$).

Note that our current cost function only includes uncertainties of the agent’s own perception in a simplistic form. When humans and agents have mismatched perceptual basis, the human’s model of comprehension and tolerance of inaccurate description could play a role in REG. Incorporating human models in the cost function will require in-depth empirical studies and we will leave that to our future work.

<table>
<thead>
<tr>
<th>Cost Function</th>
<th>Regular Graph</th>
<th>Hypergraph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Costs</td>
<td>33.2%</td>
<td>33.3%</td>
</tr>
<tr>
<td>Absolute Preferred</td>
<td>30.1%</td>
<td>30.3%</td>
</tr>
<tr>
<td>Relative Preferred</td>
<td>31.1%</td>
<td>35.4%</td>
</tr>
<tr>
<td>Uncertainty Based</td>
<td>35.7%</td>
<td>37.5%</td>
</tr>
<tr>
<td>Uncertainty Rel. Prefer.</td>
<td>36.7%</td>
<td>45.2%</td>
</tr>
</tbody>
</table>

Table 3.1: Results with different cost functions
3.3.3.2 The Role of Imperfect Perception

To further understand the role of hypergraphs in mediating mismatched perceptions between humans and agents, we created a perfect scene regular graph and a perfect scene hypergraph (representing the agent’s perfect knowledge of the environment) for each of the 48 scenes used in the experiments. In each of these scene graphs, the attribute and relation descriptors are manually provided. We further applied the Absolute Preferred cost function (which has shown competitive performance in previous work) to generate referring expressions for each object. Again, each referring expression received three votes from the crowd.

Table 3.2 shows the results comparing two conditions:

- (1) REs generated (by the Absolute Preferred cost function) based on the perfect graphs which represent the agent’s perfect knowledge and perception of the environment; and

- (2) REs generated based on automatically created graphs (by the Uncertainty Relative Preferred cost function) which represent the agent’s imperfect knowledge of the environment as a result of automated recognition and inference.

The result shows that given perfect knowledge of the environment, hypergraphs only perform marginally better than the regular graphs ($p = 0.07$). Given imperfect knowledge of the environment, hypergraphs significantly outperforms the regular graphs by taking advantage of spatial grouping information ($p = 0.002$). It is worthwhile to mention that currently we use spatial proximity to identify groups. However, the hypergraph based approach is not restricted to spatial grouping. In theory, it can represent any type of group based on different similarity criteria.

Furthermore, our result shows that the graph-based approaches perform quite competitively under the condition of perfect knowledge and perception. Although evaluated on different data...
<table>
<thead>
<tr>
<th>Environment</th>
<th>Regular Graph</th>
<th>Hypergraph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfect Perception</td>
<td>80.4%</td>
<td>84.2%</td>
</tr>
<tr>
<td>Imperfect Perception</td>
<td>36.7%</td>
<td>45.2%</td>
</tr>
</tbody>
</table>

Table 3.2: Results of comparing perfect perception and imperfect perception of the shared world.

sets, this result is consistent with results from previous work [24, 25]. However, what is more interesting here is that while graph-based approaches perform well when the agent has perfect knowledge of the environment, as its human partner, these approaches literally fall apart with close to 40% performance degradation when applied to the situation where the agent’s representation of the shared world is problematic and full of mistakes.

These results indicate that REG for automatically perceived scenes can be extremely challenging. Many errors result from automated perception and reasoning that will affect the internal representation of the world and thus the generated REs. In our experiments here, we applied a very basic CV algorithm which resulted in rather poor performance in our data: overall, 60.3% of objects in the original scene are mis-recognized, and 10.5% of objects are mis-segmented. We think this poor CV performance represents a more challenging problem.

Some errors such as recognition errors can be by-passed using our current approach based on hypergraphs. For example, in Figure 3.1 target object 9 (a stapler) and 13 (a key) are mis-recognized as a cup and a pen. Using our hypergraph-based approach, for the target object 9, instead of generating “a small cup” (as in the case of using regular graphs), “a gray object on the top within a cluster of four objects” is generated. For the target object 13, instead of “a pen” as generated by regular graphs, “a small object on the right within a cluster of 4” is generated. Even with recognition errors, these group-based descriptions will allow the listener to identify target objects in their representation correctly.
<table>
<thead>
<tr>
<th></th>
<th>Minimum Effort</th>
<th>Extra Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfect Perception</td>
<td>84.2%</td>
<td>88.1%</td>
</tr>
<tr>
<td>Imperfect Perception</td>
<td>45.2%</td>
<td>51.5%</td>
</tr>
</tbody>
</table>

Table 3.3: Results of comparing minimum effort and extra effort using hypergraphs

Nevertheless, many processing errors cannot be handled by our current approach. For example, an object can be mistakenly segmented into multiple parts or several objects can be mistakenly grouped into one object.

In addition, our current semantic grounding functions are simple. Sometimes they do not provide correct descriptors for the extracted visual features. More sophisticated functions that better reflect human’s visual perception [91, 92, 93] should be pursued in the future.

3.3.3.3 The Role of Extra Effort

While REG systems have a tendency to produce minimal descriptions, recent psycholinguistic studies have shown that speakers do not necessarily follow the Grice’s maxim of quantity, and they tend to provide redundant properties in their descriptions [94, 95, 96]. With this in mind, we conducted a very simple evaluation on the role of extra effort. Once a set of descriptors are selected based on the minimum cost, one additional descriptor (with the least cost among the remaining attributes or relations) is added to the referential description. We once again solicited the crowd feedback to this set of expressions generated by extra effort. Each expression again received three votes from the crowd. Evaluation is again based on the identification accuracy.

Table 3.3 shows the results by comparing minimum effort with extra effort when using hypergraphs to generate REs. As indicated here, extra effort (by adding one additional descriptor) leads to more comprehensible REs with 3.9% improvement under perfect perception and 6.3% improve-
ment under imperfect perception (both are significant, $p < 0.05$). The improvement is larger under imperfect perception. This seems to indicate that exploring extra effort in REG could help mediate mismatched perceptions in situated dialogue. This finding is in line with some Psycholinguistic study showing that over-specification can speed up the object identification process [96]. However, more understanding on how to engage in such extra effort will be required in the future.

### 3.4 Conclusion

In situated dialogue, humans and agents have mismatched perceptions of the shared environment. To facilitate successful referential communication between a human and an agent, the agent needs to take such discrepancies into consideration and generate referential descriptions that can be understood by its human partner. With this in mind, we re-visited the problem of referring expression generation in the context of mismatched perceptions between humans and agents. In particular, we applied and extended the state of the art graph-based approach [23] in this new setting.

Our empirical results have shown that,

- To address the agent’s limited perceptual capability, REG algorithms will need to take into account the uncertainties in perception and reasoning. More specifically, the cost functions should be tied to the agents ability to perceive and infer the environment.

- Group-based information appears more reliable explicitly model group-based information could circumvent some perceptual errors occur in perceiving the environment. Thus group-based information should be modeled by an approach that deals with automated perception of spatially rich scenes.

- While graph-based approaches have shown effective for the situation where the agent has
complete knowledge of the environment, as its human partner, these approaches are often inadequate when humans and agents have mismatched representations of the shared world. This implies that, traditional paradigm of REG approach might not be applicable in situated dialogue.

Our empirical results here call for new solutions to address the mismatched perceptual basis. Previous work indicated that referential communication is a collaborative process [16, 38]. Conversation partners make extra effort to collaborate with each other. For the situation with mismatched perceptual basis, a potential solution thus should go beyond the objective of generating a minimum description, and towards a collaborative model which incorporates immediate feedback from the conversation partner [39].
Chapter 4

Two Collaborative Models for REG

This chapter focuses on collaborative models for REG 1

4.1 Introduction

Recently, there is an increasing interest in REG for situated dialog [60, 35, 44, 97, 51]. While traditional approaches work well in situated dialogue in a virtual world, they are not adequate to support situated dialogue in a physical world because of the mismatched perceptual basis between the human and the agent.

To investigate the problem of REG under the mismatched perceptual basis, In chapter 3, We have conducted a study using the artificial scenes (an example is shown in Figure 3.1(a)). The original scene is what a human sees, and the corresponding impoverished scene is a re-rendering of the robot’s internal representation of the same scene, created by processing the original scene with a computer vision algorithm [85] and a perceptual grouping algorithm [79]. Each object/group has an ID associated for the identification purpose. This example demonstrates the perceptual differences between the human and the agent. For example, in the impoverished scene, some objects (e.g., object 19, 20) are missing or mis-recognized (e.g., object 2, 10), and perceptual groupings are

1The majority of this chapter was published in the following paper: Rui Fang, Malcolm Doering, Joyce Y. Chai. Collaborative Models for Referring Expression Generation in Situated Dialogue. Proceedings of the Twenty-Eighth National Conference on Artificial Intelligence (AAAI), Quebec, Canada, 2014.
also different (e.g., group 18 contains only 2 objects). Using these scenes, we applied a hypergraph-based approach to REG. Our experimental results have shown that, although the hypergraph-based approach performs better than the leading approach based on regular graphs [23], our approach only achieved 45% accuracy (in referential identification) under the mismatched perceptual basis. Our results indicate that, in situations where the shared perceptual basis is missing, if the agent applies traditional approaches to generate referring expressions, the intended objects often cannot be correctly identified by the human. Inspired by these findings, we have developed collaborative models to REG to particularly address the mismatched perceptual basis. We use the same scenes and target objects as used in Chapter 3 in our evaluation in order to have a valid comparison.

Next, we will first introduce the observations from collaborative behaviors in human’s referential communication. Based on these observations, we describe our two collaborative models for REG, which do not follow the traditional paradigm of REG algorithm. We then present the evaluation and comparison with the hypergraph based approach (introduced in Chapter 3) under the mis-matched perception basis. We will also discuss the importance of human feedback in collaborative referring process and possible future directions.

4.2 Collaborative Behaviors in Human-Human Referential Communication.

Previous psycholinguistic studies have indicated that referential communication is a collaborative process [16]. To minimize the collaborative effort, partners tend to go beyond issuing an elementary referring expression (i.e., a single noun phrase), but rather using other different types of expressions such as episodic, installment, self expansion, self correction, etc. [16, 98] have identified nine different types of referring behaviors listed below:
As mentioned in the introduction, Clark and Bartenger[2] have identified nine types of noun phrases in referring expressions. The first and second type have only one-way communication: It represents the approach most current and past research on REG has taken. In all other types there is collaboration between the director and the matcher to identify the right figure.

- **Elementary noun phrases**: These are the only non-iterated phrases, such as: The triangle rabbit.

- **Self-corrected noun phrases**: Phrases where the director corrects himself, such as: The one with the large square, ehh, triangle in the middle.

- **Other-corrected noun phrases**: Noun phrases that are corrected by the matcher, such as:
  Director: ... with the triangle on top.
  Matcher: Don’t you mean the square?
  Director: Yes, you’re right.

- **Expanded noun phrases**: The speaker at first produces a complete and distinguishing description and only continues describing because the hearer did not understand it. The difference with the installment noun phrase is that there the speaker does not produce a matching description. An example:
  Director: The next one is the rabbit.
  Matcher: Uhh....
  Director: That’s asleep. It looks like it’s got pointy ears.

- **Episodic noun phrases**, noun phrases produced in two or more easily distinguished intonation units, enabling the matcher to think and interact, such as: The next one is a rabbit, <small pause> with ears and a head pointing down.
• **Installment noun phrases**: The speaker starts with the first part of a distinguishing description, waits for acknowledgment and after that, continues. It looks much like the episodic noun phrases, the only real difference is the absence of a response from the matcher in the episodic noun phrases. An example:

Director: The next one is the one who’s sitting,
Matcher: Okay.
Director: With his knees bended.

• **Trial noun phrases**: Almost the same as four. Instead of just waiting, the director asks a yes-no question, like: You see the ice-skater? Depending on the answer, the director will continue differently.

• Holder noun phrases: Noun phrases that are used as a stand-in when a speaker does not know how to refer to a certain object such as whatchamacallit in: The eh-hmm - whatchamacallit, the, eh-mm... the rabbit.

• Invited noun phrases With these, directors invite the matcher to complete the expression, for example:

Director: and the last figure is ehh...
Matcher: the rabbit.

Among these nine types of referring expressions, *episodic descriptions* and *installment descriptions* are particularly collaborative. Unlike a single elementary description to describe a target object, an episodic description is produced in two or more easily distinguished episodes or intonation units. Here is an example of an episodic description from [75].

A: below the orange, next to the apple, it’s the red bulb.
An installment behavior is similar to the episodic behavior in the sense that it also breaks down generation into smaller episodes. The difference is that an explicit feedback from the addressee is solicited before the speaker moves to the next episode. Here is an example of an installment from [75].

A: under the pepper we just talked about.
B: yes.
A: there is a group of three objects.
B: OK.
A: there is an a yellow object on the right within the group.

The generation of episodic or installment descriptions is not to minimize the speaker’s own effort, but rather to minimize the collaborative effort so that the addressee can quickly identify the referent. These collaborative behaviors from human-human referential communication have motivated our collaborative models.

4.3 Two Collaborative Models for REG

Inspired by the episodic noun phrases and installment noun phrases used in human-human dialogue, we developed two collaborative computational models for REG:

- **Episodic Model.** The episodic model generates referring expressions (REs) in an “episodic” fashion: it generates a RE in a sequence of smaller noun phrases which lead to the target object.

- **Installment Model.** The installment model generates REs in an “installment” fashion: it generates one small noun phrase, waits for partner’s response, and then generates another small noun phrase. This process iterates until the target object is reached.
For both models the goal is to construct a sequence of objects to describe, where the target object is the final object to be described in the sequence. Both models can choose to directly describe the target object (as the traditional REG methods do) if such choice is deemed to have the lowest overall cost. But in general these models often find an object that is “easier” to describe and then gradually lead to the target object.

For example, in Figure 4.1 suppose the robot wants to describe the target object 7. Note that it is hard to describe this object directly based on its features. So both models will search for a sequence of objects that lead to the target. In the episodic model, a sequence could be

\[ object 5 \rightarrow group 15 \rightarrow object 7 \]
which could be linguistically realized as: “a red pepper, below that a group of 3 objects, and a yellow object on the right within the group”. In the installment model, once an object (e.g., object 5) is chosen, it will ask the human to provide feedback. Then based on the feedback, it will search for another object and iterate this process until the target object is reached.

4.3.1 Episodic Model

The episodic model is based on the Branch-and-Bound search method [99]. It searches for the path to a target object with the overall least cost. We represent our problem space as a directed graph $G = (N, A)$, in which

$$N = \{n_i\}$$

$$A = \{a_j = \langle t_j, h_j \rangle \mid t_j, h_j \in N\}$$

$N$ is a set of nodes, and $A$ is a set of arcs. Each node $n \in N$ represents a perceived object or a perceptual grouping of objects by the agent, together with one of all possible concatenations of linguistic descriptors describing attributes (e.g., type, color, etc.). Each descriptor comes with a cost, which indicates how accurately the linguistic descriptor matches the underlying physical features. We use the ‘Uncertainty Relative Preferred’ cost functions from [9], where the costs for hyperarcs representing relative attributes and relations are divided by 3. This cost function will allow the algorithm to prefer spatial relations through the reduced cost.

A path in graph $G$ is a sequence of nodes $\langle n_0, \ldots, n_k \rangle$ such that $\langle n_{i-1}, n_i \rangle \in A$. The cost of a path is the sum of the cost of all the nodes and arcs on the path:

$$\text{cost}(\langle n_0, \ldots, n_k \rangle) = \sum_{i=0}^{k} \text{cost}(n_i) + \sum_{i=1}^{k} \text{cost}(\langle n_{i-1}, n_i \rangle)$$

Figure 4.2 details our episodic model for generating episodic expressions. The algorithm starts
with an initial state in which the path contains only the root node $n_0$, the best path is none $\bot$, and the bound is set to infinity. The algorithm then recursively extends the path $\langle n_0 \rangle$ until the minimum cost path is found. The Boolean function $isDiscriminating()$ measures whether a path can uniquely identify object $n_k$ in the path. The Boolean function $isGoal()$ tests whether $n_k$ is the target node.

Figure 3.3

### 4.3.2 Installment Model

In the Installment model, we treat the collaborative referring expression generation task as a sequential decision making problem and formalize it under the reinforcement learning framework. Our goal is to learn a good strategy that can construct a series of objects in the scene which will lead to the successful identification of a target object.

Our model is inspired by [100, 101], where reinforcement learning is applied to interpret navigational natural language instructions. Here we apply reinforcement learning to optimize the generation strategy that can generate most appropriate referring expressions to successfully identify the target object. We will first introduce the basic reinforcement learning concepts and show how we formalize our installment model under the reinforcement learning framework.

#### 4.3.2.1 Reinforcement Learning Basis

Reinforcement Learning is based on the Markov Decision Process (MDP). The main ingredients of a reinforcement learning approach contains:

- a set of states $S = \{s_1, s_2, ..., s_N\}$ that represents the agents knowledge of the previous and current context. In REG task, the state should contain all relevant information that are important for the agent to generate referring expressions. States need to meet the Markov
property which requires that the current state contains all the information in such a way that, an agent’s decisions making based on the current state is just as good as if it made decisions based on the entire history of states.

- a set of actions $A = \{a_1, a_2, ..., a_M\}$ that represents the agents potential behavior under a certain state. For REG task, an action need to decide which object to describe, how to describe the object (e.g., what attributes to include in the description, what relation need to be mention with respective to other objects).

- a state transition function $T$ that specifies the next state of the agent for taking an action in a state. The transition could be stochastic or deterministic. For REG task, after an agent generated a particular referring expression in the current state, how does this referring expression change the environment? (E.g., what is the user’s response? How many objects are still available to describe?)

- a reward function $R$ that assigns a numeric value to the agent for taking an action in a state. A reward represents the goal that we want the learning agent to achieve. It is critical that the rewards we set up truly indicate what we want accomplished. In REG task, the assignment of reward could be based on different criterion: e.g.:

  1. how well the description can uniquely identify the target object and rule out all other distractors.

  2. how complex the description is (e.g., the length of the description).

  3. what kind of the descriptors are included in the description (e.g., is color mentioned in the description?).

  4. how confident is to use a description for a given target object, etc.
• a policy $\pi$ is any mapping from states to actions. A policy specifies what actions to take in a state. In our REG task, a policy will decide what objects to describe and how to describe it given certain situation.

The dynamics of an MDP can be described as follows. At the beginning of an interaction between the agent and the environment, when the time step $t = 0$, the agent receives a representation of the environment, called the state $s_t \in S$. It needs to perform an action $a_t \in A$ that is the set of actions available in state $s_t$. As a result, the agent will receive a reward $r_{t+1} \in R$ and observe the next state $s_{t+1} \in S$, which is the updated environment state. This process can be seen as a finite sequence of states, actions and rewards $\{s_0, a_0, r_1, s_1, a_1, r_2, ..., s_t\}$ and is illustrated in Figure 4.3. The agent's goal is to learn an optimal policy $\pi^*$, a mapping from every state to an action $a$ that will yield the highest expected return.

In particular, we wish to develop an agent that learns an optimal generation policy, i.e. a mapping from any generation context $s$ to a generation action $a$ that will maximize the expected success of a generated utterance.

Remember that an agent does not learn a policy to optimize immediate rewards. Rather, the agent's goal is to optimize the cumulative reward that can be obtained for an entire sequence of actions. Thus, to identify those actions that yield the best expected long-term return, the agent needs to apply a trial and error strategy trying individual actions many times in different states in order to estimate their real reward.

We refer to [10] for a comprehensive overview of the reinforcement learning approaches. Now, it is time for us to present the detailed formalization of the installment model under reinforcement learning framework.
4.3.2.2 State

The state set is denoted as $S$, $s \in S$. Let $O$ be the set of all perceived objects and perceptual groups in the scene. A state $s = <t, lm, re, W>$ of our sequential decision making problem contains the target object $t \in O$, the landmark $lm \in O$ and its generation strategy $re \subset RE$ (will explain in the Action section), which is confirmed by user (e.g., when the user accepts the description of the landmark object), together with a set of objects $W$ in the scene which contains the objects that have not been used but can potentially be used as landmark objects: $W = \{w_1, \ldots w_n\}, w_i \in O$. In the initial state $lm$ is none, and $W = O$. When a user accepts the description of an object or a group it becomes the landmark object $lm$. When a user rejects the description of an object or a group, it is removed from $RE$. In the terminal state, the target object is reached and becomes the current landmark object $t = lm$.

4.3.2.3 Action

An action in our problem basically describes an object in relation to a landmark object. The action set is denoted as $A$, $a \in A$. $a = \langle o, re, sp \rangle$ is composed of an object to be described $o \in W$, its generation strategy $re \subset RE$, and a spatial relation $sp \in SP$ to the landmark object $lm$. Currently, $RE$ represents the strategies that can be used to describe an object. The space of $RE$ consists of all possible concatenations of the following dimensions:

- **desType**: describes the type of the object such as “apple”, “pen”
- **desColor**: describes the color of the object such as “red”, “yellow”
- **desSize**: describes the size such as “big” or “small”.
- **desSpaLoc**: describes the spatial location with respect to the whole scene, e.g., “on the left”,


“on the right”, etc.

- desNumGroup: describes the number of objects within a group.

SP represents the strategies that can be used to describe a spatial relation between two objects, between two perceptual groups, between one object and a perceptual group or between one object and the group it belongs to. The space of SP can be one of the following dimensions:

- desBetween: describes the spatial relation between two objects/groups, e.g., “below”, “above”, “to the left of”, and “to the right of”.

- desInGroup: describes the spatial relation between one object and the group it belongs to, e.g., “on the right within”, “on the left within”, “on the bottom within”, “on the top within”.

For example, an action sequence could be

\[
< \text{object5}, [\text{desColor}, \text{desType}], [\text{none}] > \\
\downarrow \\
< \text{group15}, [\text{desNumGroup}], [\text{desBetween}] > \\
\downarrow \\
< \text{object7}, [\text{desColor}], [\text{desInGroup}] >
\]

These actions capture the decisions on how to generate an expression to describe an intended object. Given an action, the surface form of an expression can be generated through some templates.

4.3.2.4 Transition

\(T(s, a)\) is the transition function. It takes an action \(a = \langle o, re, sp \rangle\) in current state \(s = \langle \text{tar}, \text{lm}, re, W \rangle\) and gives the next state \(s' = \langle \text{tar}, \text{lm}', re', W' \rangle\). Note that, the action does not change the target \text{tar}. 

72
Rather the landmark object and $W'$ are affected. Given the expression generated by a particular action, the human’s response will lead to the update of the landmark object $lm'$ and its description $re'$, as well as the remaining objects that can be used as landmark objects in the future $W'$.

4.3.2.5 Reward

The goal of the reinforcement learning algorithm is to identify a policy to maximize the expected utility

$$U_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k},$$

where $0 \leq \gamma \leq 1$.

The reward function $r(s, a)$:

$$r(s, a) : S \times A \rightarrow \mathbb{R}$$

It specifies a numeric reward that an agent receives for taking action $a$ in state $s$. It is defined as follows:

$$r = \begin{cases} 
100 & \text{if the terminal state is reached and the target object is correctly identified.} \\
10 & \text{if the terminal state is not reached and the current object is correctly identified.} \\
-1 & \text{otherwise.}
\end{cases}$$

4.3.2.6 Policy

A policy $\pi : S \rightarrow A$ is a mapping from states to actions. We can determine the expected return of a policy by estimating its action-value function. The action-value function $Q^\pi(s, a)$ of a policy $\pi$
is the expected return for starting in state $s$, taking action $a$ and then following $\pi$. It is defined as

$$E_\pi[U_t|s_t = s, a_t = a]$$

The optimal action-value function that yields the best expected return for each action $a$ taken in state $s$ is defined as

$$Q(s, a)^* = \max_\pi Q^\pi(s, a)$$

The optimal policy $\pi^*$ is therefore the policy that chooses the action $a$ that maximizes $Q(s,a)$ for all $s \in S$,

$$\pi^*(s) = \arg\max_a Q(s, a)$$

Basic reinforcement learning approaches use a table to represent the action-values. However, as the number of states in an environment increases, it becomes infeasible for an agent to visit all possible states enough times to find the optimal actions for those states. Furthermore, it becomes important to be able to generalize the learning experiences in a particular state to the other states in the environment. One common way to handle the large state space and generalization issue is through function approximation [10]. Existing function approximation methods for value prediction includes artificial neural networks, decision trees, and linear functions. We apply the linear functions approximation for our work here.

We define a mapping $\phi$ that assigns a feature vector to each state-action pair. Then, the action value $Q(a,s)$ of a state-action pair $(s,a)$ is obtained by linearly combining the components of $\phi(s, a)$ with the weights $\theta$:

$$Q(s, a) = \sum_i^d \theta_i \phi(s, a)$$

Here we represent the $Q$-function as a linear combination of the features, the agent extracts
<table>
<thead>
<tr>
<th>#</th>
<th>Feature Value Description</th>
<th>Learned Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>normalized sum of vision confidence</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>of all descriptors in re</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>is spatial location descriptor in re?</td>
<td>0.54</td>
</tr>
<tr>
<td>3</td>
<td>visual confidence of spatial relation descriptor</td>
<td>0.52</td>
</tr>
<tr>
<td>4</td>
<td>vision confidence of type descriptor</td>
<td>0.51</td>
</tr>
<tr>
<td>5</td>
<td>vision confidence of spatial location descriptor</td>
<td>0.48</td>
</tr>
<tr>
<td>6</td>
<td>is type descriptor in re?</td>
<td>0.21</td>
</tr>
<tr>
<td>7</td>
<td>number of descriptors in re</td>
<td>0.19</td>
</tr>
<tr>
<td>8</td>
<td>vision confidence of size descriptor</td>
<td>0.13</td>
</tr>
<tr>
<td>9</td>
<td>is size descriptor in re?</td>
<td>0.13</td>
</tr>
<tr>
<td>10</td>
<td>is color descriptor in re?</td>
<td>0.10</td>
</tr>
<tr>
<td>11</td>
<td>vision confidence of color descriptor</td>
<td>0.09</td>
</tr>
<tr>
<td>12</td>
<td>can re together with sp to lm uniquely identify an object?</td>
<td>0.88</td>
</tr>
<tr>
<td>13</td>
<td>is there a sp between o and tar?</td>
<td>0.51</td>
</tr>
<tr>
<td>14</td>
<td>number of spatial links from lm</td>
<td>0.23</td>
</tr>
<tr>
<td>15</td>
<td>number of spatial links to o (in degree)</td>
<td>0.01</td>
</tr>
<tr>
<td>16</td>
<td>number of spatial links from o (out degree)</td>
<td>0.01</td>
</tr>
<tr>
<td>17</td>
<td>is o and lm in the same group?</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>is o a group?</td>
<td>0.97</td>
</tr>
<tr>
<td>19</td>
<td>is lm a group and o in lm?</td>
<td>0.96</td>
</tr>
<tr>
<td>20</td>
<td>is o a group and tar in o?</td>
<td>0.90</td>
</tr>
<tr>
<td>21</td>
<td>is o and tar in the same group?</td>
<td>0.51</td>
</tr>
<tr>
<td>22</td>
<td>is o a group and lm in o?</td>
<td>0.11</td>
</tr>
</tbody>
</table>

Table 4.1: Features used in the installment model. We use the following abbreviations, o: an object to describe, re: generation strategy for an object, lm: landmark object, tar: the target object, sp: spatial relation.

the relevant information it needs from the state through feature extraction, and uses the resulting feature vector to calculate the approximate value it gets from being in that state. This is done by doing a dot product between the feature vector and a parameter vector. In this way, similar state features will also have similar values and learn weights which most closely approximate the true expected reward. The goal of reinforcement learning with function approximation is then to learn the best values for this parameter vector.
4.3.2.7 Features

We use features from both the state and the action to model the \( Q \)-function as summarized in Table 4.1. Features 1-11 are inspired by [9] which demonstrates that encoding visual confidence of a symbolic descriptor into the cost function improves performance. Feature 12 models the discriminating power of a referring expression, which is the key requirement for traditional REG approaches. Features 13-16 measure the spatial relations between objects. Features 17-22 are inspired by previous work indicating that group descriptions are important to REG [13, 9].

4.3.2.8 Learning

To learn these weights \( \theta \) we follow the method in [101], which uses SARSA [10] with linear function approximation. The learning model is shown in Figure 4.4. We use \( \Pr(a_0|s_0; \theta) = \frac{\exp(\theta^T \phi(s_t,a_t))}{\sum_{a'} \exp(\theta^T \phi(s_t,a'))} \) to choose the best action based on the current estimation of \( \theta \), with \( \epsilon \)-greedy (\( \epsilon = 0.2 \)) for the exploration (That means 20% of the time, we randomly choose an action). The learning rate \( \alpha_t \) is set to \( \frac{30}{30+t} \) and we stop training when the magnitude of updates \( \|\theta_{t+1} - \theta_t\| \) is smaller than 0.0001. The parameters are tuned based on our own empirical interaction of the installment model.

4.4 Empirical Evaluations

4.4.1 Experimental Setup

To evaluate the performance of both the episodic model and the installment model to REG, we conducted an empirical study using crowd-sourcing from the Amazon Mechanical Turk. Through a web interface (e.g., Figure 4.5 and 4.6) we displayed an original scene and a description generated
to refer to an object in the scene. We asked each turkers to choose the object he/she believed was referred to by the description. They can also choose if none of the objects or multiple objects were considered to be referred to.

Similar to [9], each description received three votes regarding its referent from the crowd. The referent with a majority voting was taken as the identified referent and was used to calculate the performance metric: the accuracy of referent identification (i.e., the percentage of generated referring expressions where the target object are correctly identified). If all three votes were different, then the referent was evaluated as not correctly identified for that expression.

4.4.2 Training of the Installment Model

In order to learn the weights for features in the installment model, we first created 32 different training scenes similar to the scenes used in [9], then used the Amazon Mechanical Turk to solicit feedback from the crowd. The training was divided into sessions where each session was used to identify only one target object. More specifically, in each session, the system applied Figure 4.4 to pick an action. Then a referring expression was generated based on this action and shown to the user. The user was then asked to identify the referent based on this expression. Based on the user’s feedback, the internal state would be updated and the system would pick the next action accordingly. This process iterates until one of the following three conditions was met:

- The system reached the target object and describes it.
- There was no more action available for the system to take.
- The number of interactions exceeds 13, which is the average number of objects in a scene.

\(^2\)To have a fair comparison, we use the same quality control of crowdsourcing as used in [9].
When one training session ended, a new training session for a new object would start. We stopped training when the algorithm converged. We had a total number of 1860 training sessions and a total number of 6775 interaction with the user, which resulted in an average of 3.6 interactions per session.

Figure 4.7 shows the number of objects correctly identified during training at an interval of 200 sessions. From Figure 4.7, we observe a trend that the number of correctly identified objects gradually increases, indicating that the system is learning in the right direction towards the true underlying parameter $\theta$. Note that we stopped the training when the algorithm converged. However, as shown in Figure 4.7, the performance seems to pick up during the last 200 sessions. This suggests that further training may improve the performance.

The weights learned for each feature are shown in the third column of Table 4.1. We can observe that, group-based features (e.g., features 17 - 20) are assigned relatively higher weights. Through the interaction with the user, the system learns that a strategy of describing an object with reference to groups has a higher probability of correctly identifying the target object. This learning result is consistent with previous work showing that group descriptions are important. However, in contrast to previous work, our system can learn the importance of the group descriptions through interaction and gradually assign higher weights to them. Visual confidence (feature 1) and the discriminating power (feature 12) of a referring expression also receive relatively high weights.

Given these features with learned weights, we can modify the episodic model in two ways

1. using the trained installment model, but assume that turker’s feedback is the intended object described by the installment model (We name it Epsodic Model II)

2. using the features with learned weights to guide the Branch and Bound search in the episodic model (We name it Episodic model III)
Table 4.2: Comparison of accuracy for referent identification based on expressions generated from three models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-collaborative model [9]</td>
<td>47.2%</td>
</tr>
<tr>
<td>The episodic model</td>
<td>53.6%</td>
</tr>
<tr>
<td>The episodic model II</td>
<td>56.9%</td>
</tr>
<tr>
<td>The episodic model III</td>
<td>57.8%</td>
</tr>
<tr>
<td>The installment model</td>
<td>68.9%</td>
</tr>
</tbody>
</table>

4.4.3 Evaluation Results

In the testing phase, we applied the learned parameters to the testing scenes and evaluated how well the turkers \(^3\) can identify target objects based on the expressions generated by the learned policy. We used the same 48 scenes used in [9] for evaluation. Each testing scene has 13 objects on average and there are 583 objects in total \(^4\). The turkers went through all the testing scenes and we recorded the number of responses the turkers made while identifying the target objects, as well as the total number of target objects that were correctly identified. Each expression received three votes from the crowd and the referent with the majority voting was used in calculating the accuracy.

Table 4.2 shows the results of our two collaborative models compared with the non-collaborative model [9].

- Two collaborative models significantly outperform the non-collaborative approach with an absolute performance gain of over 6% and 21% respectively.

- The installment model further significantly outperforms the episodic model with an absolute

\(^3\)We randomly recruit the turkers from Amazon Mechanical turkers using the same quality control criteria. The turkers are presented with the same interface as in the training phase.

\(^4\)Here we remove 38 split objects from the target object set, so the adjusted result shown in Table 4.2 is a little different from the reported results in [9].
performance gain of 15% (Pearson’s Chi-square test, $\chi^2 = 20.11$, $p < 0.001$ with Tukey’s Post Hoc test). The average number of turns in the installment model is 2.35 (Standard Deviation = 1.07).

4.4.3.1 Comparison of the Episodic Model with the Non-Collaborative Model.

The episodic model outperforms the non-collaborative model. This is mainly due to two reasons:

- First, instead of directly describing the target object, the episodic model provides more options to describe the target object, and these options have less cost than the direct description used in the non-collaborative model. In the episodic model 72% of the target objects are not described directly (i.e., one long noun phrase to describe the target), which indicates that, in these cases describing other objects first and then gradually approaching the description of the target object has a lower overall cost.

- The second reason is due to the complexity of interpreting a description. The average number of descriptors in a referring expression for the non-collaborative model (Mean=4.3,SD=1.23) is larger than that in each small noun phrase in the episodic model (Mean=2.2,SD=0.78) (t-test, $p < 0.001$). In the non-collaborative model, the target object can be described in relation to other objects. Although as a whole the overall description should be able to distinguish the target from the rest of objects, the expressions to describe each of the objects (in relation to the target) do not need to be distinguishing. In contrast, the smaller descriptions generated by the episodic model can already distinguish the intended object from the rest. We believe these behaviors contribute to a less complexity in interpreting episodic expressions.

4.4.3.2 Comparison of the Installment Model with the Episodic Model.

There are two main reasons that the installment model outperforms the episodic model.
• First, the installment model incorporates many more features than the episodic model, and the weight of those features are learned through on-line interaction with users. The episodic model only relies on two features when searching for the best path to a target object: vision uncertainty and the discriminating power of a path. These two features roughly corresponds to feature 1 and 12 in Table 4.1. Realizing this reason and to have a fair comparison, we build episodic model III, which applies the same features as used in the installment. The results show that, the installment model still outperform the episodic model III with absolute performance gain of 11%.

• Second, in the episodic model there is no intermediate feedback from humans, while in the installment model the system selects the next object to describe based on the intermediate feedback from the human partner. To explore the role of intermediate feedback, we build Epsodic Model II, which uses the trained installment model, but assume that turker’s feedback is the always the intended object described by the installment model. The results show that with intermediate feedback, the installment model outperforms the episodic model II with an absolute performance gain of 12%.

Figure 4.8 shows among all failed cases in the episodic model, where do these fail case occur. Among all the successful sessions (where the target object is correctly identified), 18.7% of them in fact encountered some problems. (The turkers could not identify the object at the intermediate step.) However, the system was able to get around the problem and chose to describe another object. It is the intermediate feedback that guides the installment model to generate referring expressions that lead to the correct identification of target objects.

Here is an example comparison between the episodic model and installment model. The target object is object 10 in Figure 4.1. For the episodic model the sequence of objects to describe is
group18 → object10. It is linguistically realized as: ‘a group of 2’ → ‘an object on the top within the group of 2’. While, for installment model:

System: ‘a group of 2’
User: None of the above
System: ‘a group of 4’
User: group 16
System: ‘an orange object below the group of 4’
User: object 10

4.5 Conclusion

In situated dialogue, humans and robots have mismatched perceptions of the shared world. To facilitate successful referential communication between a human and a robot, the robot needs to take into account such discrepancies and generate referring expressions that can be understood by its human partner. Motivated by collaborative behaviors in human-human referential communication, we developed two collaborative models for REG. In contrast to previous non-collaborative models which tend to generate a single long description to describe a target object, our models generate multiple short expressions describing easier-to-identify landmark objects that eventually lead to the target object. The goal of these models is to minimize the collaborative effort between the human and the robot. Our empirical results have shown that the two collaborative models significantly outperform the non-collaborative model.

Although our current models, especially the installment model, have yielded encouraging results, several problems need to be addressed in the future.
First, the cost function we used for each descriptor is predefined. As the environment changes, the agent may have different confidence in capturing different dimensions of perception. Thus, one future direction is to automatically learn and adjust these cost functions in new environments.

Second, our current evaluation is only based on the simplified scenes without addressing complexities in true human-robot dialogue. Even for our installment model, although follows the definition from (Clark and Wilkes-Gibbs 1986) does not fully resemble human-human dialogue, where the addressee can take a more active role in referential grounding. Our future work will extend the current work to real human-robot dialogue and incorporate more interaction strategies and non-verbal modalities (e.g., gaze direction from the human) as intermediate feedback for generating referring expressions.

Third, based on reinforcement learning framework, in order for the installment model to converge, it needs thousands of interactions between user and our system. This posts the question of how to faster the learn phase of the installment modal in real human-robot interaction.

Fourth, although in the installment model we can learn a policy to generate description, the model does not explicitly learn where the mismatched perception occurs (e.g., after learning, the scene hypergraph is still full of error: objects are still mis-recognized/missing, etc). So one direct extension is to directly mediate the mis-alignment in perception. The effect is that, after learning, the robot’s visual perception system will be improved and the recognition performance is better.
Algorithm `generateEpisodic()`

\[
G \leftarrow (N, A) \\
bestPath \leftarrow \perp \\
n_0 \leftarrow \text{root node} \\
bound \leftarrow \infty \\
\text{Return } \text{Search}(G, bestPath, bound, \langle n_0 \rangle)
\]

Procedure `Search(G, bestPath, bound, \langle n_0, ..., n_k \rangle)`

\[
\text{if } \langle n_0, ..., n_k \rangle \text{ only contains } n_0 \text{ then} \\
\quad \text{foreach } \langle n_0, n_1 \rangle \in A \text{ do} \\
\quad \quad \text{Search}(G, bestPath, bound, \langle n_0, n_1 \rangle) \\
\quad \text{end} \\
\text{end} \\
\text{if } \text{cost}(\langle n_0, ..., n_k \rangle) < \text{bound } \text{And isDiscriminating}(\langle n_0, ..., n_k \rangle) \text{ then} \\
\quad \text{if isGoal}(n_k) \text{ then} \\
\quad \quad \text{bestPath} \leftarrow \langle n_0, ..., n_k \rangle \\
\quad \quad \text{bound} \leftarrow \text{cost}(\langle n_0, ..., n_k \rangle) \\
\quad \text{else} \\
\quad \quad \text{foreach } \langle n_k, n_{k+1} \rangle \in A \text{ do} \\
\quad \quad \quad \text{Search}(G, bestPath, bound, \langle n_0, ..., n_k, n_{k+1} \rangle) \\
\quad \text{end} \\
\text{end} \\
\text{Return } bestPath
\]

Figure 4.2: The Episodic Model for REG
Figure 4.3: The Agent-Environment Interface in MDP [10]

\[
\begin{align*}
\text{Input} & : \text{Object identification task set } I \\
& \quad \text{Feature function } \phi \\
& \quad \text{Transition function } T \\
& \quad \text{Reward function } r(s, a) \\
& \quad \text{Learning rate } \alpha_t \\

\text{Initialize } \theta \text{ to small random values} \\
\text{while } \theta \text{ not converge do} \\
\quad \text{foreach task } i \in I \text{ do} \\
\quad \quad \text{Initialize } s \text{ to initial state} \\
\quad \quad \text{Select } a \sim \Pr(a|s; \theta) \text{ using } \epsilon \text{ greedy} \\
\quad \quad \text{while } s \text{ not terminal do} \\
\quad \quad \quad s' = T(s, a) \\
\quad \quad \quad \text{Select } a' \sim \Pr(a'|s'; \theta) \text{ using } \epsilon \text{ greedy} \\
\quad \quad \quad \Delta \leftarrow r(s, a) + \theta^T \phi(s', a') - \theta^T \phi(s, a) \\
\quad \quad \quad \theta \leftarrow \theta + \alpha_t \phi(s, a) \Delta \\
\quad \quad \end{align*}
\]

\text{Output: Estimate of feature weights } \theta

Figure 4.4: The Installment Model for REG

85
Figure 4.5: a web-based interface shown to the Mechanical Turk workers for the episodic model

A simple object identification game

Most recent response:
We asked: “a group of 4 on the right of the screen”
You chose: "15"
You have answered 1 times. There are at most 11 questions remaining.

"a small red object on the right of the screen and on the right within [15]".

Which object/group is this referring to?
(Keep in mind that the machine may not see the same thing as you.)

Figure 4.6: a web-based interface shown to the Mechanical Turk workers for the installment model
Figure 4.7: The number of objects correctly identified at an interval of every 200 training sessions

Figure 4.8: The distribution of failure cases at the 1st, 2nd, 3rd and after the 3rd expression in the episodic model
Chapter 5

Embodied Collaborative Models for REG

This chapter focuses on embodied collaborative models for REG in situated dialogue \(^1\). In situated dialogue, robots and humans have different representations of the shared environment because of their mismatched perceptual capabilities. The robot’s representation of the shared environment is often incomplete and error-prone. When a shared perceptual basis is missing, referential communication between partners becomes difficult\(^1\). Specifically for the task of REG, \([9]\) has shown that when the human and the agent have a mismatched perceptual basis traditional approaches to REG tend to break down. A competitive algorithm that achieves over 80\% accuracy (in referent identification) when the human and the agent have the same representation of the world (i.e., the traditional setting) only obtains over 40\% accuracy under a mismatched perceptual basis \([9]\). Based on this observation, \([1]\) developed two computational collaborative models for REG: an episodic model and an installment model. Their collaborative models, motivated by collaborative referential behaviors in human-human communication \([16]\), instead of generating a single referring expression, generate multiple small noun phrases that gradually lead to the target object with a goal to minimize the collaborative effort. In a web-based study where 2D images were used as shared environment between an agent and a human \([1]\), empirical results have shown the collaborative models can better handle mismatched representations between the agent (which only has an im-

\(^1\)The majority of this chapter was published in the following paper: Rui Fang, Malcolm Doering, Joyce Y. Chai. Embodied Collaborative Model for Referring Expression Generation in Situated Human-Robot Interaction. Proceedings of the 10th Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI), Portland, USA, 2015.
poverished representation of the image due to limited object recognition capability) and the human (who has a rich representation of the shared image) and significantly outperform the traditional approaches.

While our previous results on collaborative models are encouraging, the use of 2D images as a shared environment is a great simplification of the shared physical world as in human-robot interaction. Collaborative models have not been applied to real-time human-robot interaction in a physical environment. In situated human-robot interaction, a key characteristic is embodiment: agents (i.e., robots) and humans both have physical bodies and they can use non-verbal modalities (e.g., gesture and eye gaze) to make references to the shared world and to provide immediate feedback. It is not clear to what degree the collaborative models can take into account of embodiment and apply to physical interaction. To address this issue, we have developed embodied collaborative models for REG. These models seamlessly incorporate robot gesture behaviors (i.e., pointing to an object) and human’s gaze feedback (i.e., looking at a particular object) into the collaborative model for REG. We examined different strategies incorporating robot gestures and human feedback. Our empirical results have shown that when robot gestures and human verbal feedback is incorporated, the new collaborative model achieves over 28% absolute gains compared to the baseline collaborative model. This chapter discusses the opportunities and challenges brought by modeling embodiment in collaborative referential communication in human-robot interaction.

5.1 Embodied Collaborative Models for REG

Inspired by the collaborative behaviors in referential communication, [1] developed two computational collaborative models for REG: an episodic model and an installment model. Their collaborative models, instead of generating a single referring expression to describe a target object, generate
multiple small noun phrases that gradually lead to the target object with a goal to minimize the collaborative effort. In particular, the installment model treats the collaborative referring expression generation task as a sequential decision making problem and formalize it under the reinforcement learning framework. The goal is to learn a good policy to generate installments under different situations to eventually lead to successful identification of a target object.

Through a web-based crowd-sourcing interface and interacting with users based on 2D images, the weights for a set of features are learned to represent the \( Q(s, a) \), the utility for a particular state \( s \) (which captures the agent representation of the shared environment) and a particular generation strategy \( a \) (which represents a generation strategy/action e.g., which attributes to choose to describe the object) as in:

\[
Q(s, a) = \theta^T \phi(s, a)
\]

Where \( \phi \) assigns a feature vector to each state-action pair, and \( \theta \) is the weight vector of the features. The utility of a state-action pair \( Q(a,s) \) is defined as a linear combination of the features.

Once the weights for features are learned, they are applied to calculate \( Q(s, a) \) for any state-action pairs. During real-time generation, the agent will pick the generation action \( a^* \) to maximize the \( Q(s, a) \).

\[
a^* = \arg \max_{a \in A} Q(s, a)
\]

The above collaborative model was only applied and evaluated in a simplified environment with 2D images on the computer screen [1]. Human-robot interaction is much more complex. To extend the collaborative model to real-time human-robot interaction and seamlessly incorporate embodiment into the model, we pursued two directions: (1) incorporating the robot’s gesture (i.e., pointing) into the collaborative model to generate multimodal referring expressions; and (2) incorporating the human real-time feedback (either verbal feedback, or gaze feedback, or both) into the
collaborative model. Figure 5.1(a) shows a typical setup where the embodied collaborative model is examined.

**Incorporation of robot’s gesture in referring acts.**

Instead of generating verbal descriptions only, non-verbal description such as pointing gesture can also be included. Here, we treat cost of gesture generation as a feature and incorporate it with other features learned from [1] into the collaborative models.

<table>
<thead>
<tr>
<th>#</th>
<th>Feature Value Description</th>
<th>Learned Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>normalized sum of vision confidence of all descriptors in re</td>
<td>0.92</td>
</tr>
<tr>
<td>2</td>
<td>is spatial location descriptor in re?</td>
<td>0.54</td>
</tr>
<tr>
<td>3</td>
<td>visual confidence of spatial relation descriptor</td>
<td>0.52</td>
</tr>
<tr>
<td>4</td>
<td>vision confidence of type descriptor</td>
<td>0.51</td>
</tr>
<tr>
<td>5</td>
<td>vision confidence of spatial location descriptor</td>
<td>0.48</td>
</tr>
<tr>
<td>6</td>
<td>is type descriptor in re?</td>
<td>0.21</td>
</tr>
<tr>
<td>7</td>
<td>number of descriptors in re</td>
<td>0.19</td>
</tr>
<tr>
<td>8</td>
<td>vision confidence of size descriptor</td>
<td>0.13</td>
</tr>
<tr>
<td>9</td>
<td>is size descriptor in re?</td>
<td>0.13</td>
</tr>
<tr>
<td>10</td>
<td>is color descriptor in re?</td>
<td>0.10</td>
</tr>
<tr>
<td>11</td>
<td>vision confidence of color descriptor</td>
<td>0.09</td>
</tr>
<tr>
<td>12</td>
<td>can re together with sp to lm uniquely identify an object?</td>
<td>0.88</td>
</tr>
<tr>
<td>13</td>
<td>is there a sp between o and tar?</td>
<td>0.51</td>
</tr>
<tr>
<td>14</td>
<td>number of spatial links from lm</td>
<td>0.23</td>
</tr>
<tr>
<td>15</td>
<td>number of spatial links to o (in degree)</td>
<td>0.01</td>
</tr>
<tr>
<td>16</td>
<td>number of spatial links from o (out degree)</td>
<td>0.01</td>
</tr>
<tr>
<td>17</td>
<td>is o and lm in the same group?</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>is o a group?</td>
<td>0.97</td>
</tr>
<tr>
<td>19</td>
<td>is lm is a group and o in lm?</td>
<td>0.96</td>
</tr>
<tr>
<td>20</td>
<td>is o a group and tar in o?</td>
<td>0.90</td>
</tr>
<tr>
<td>21</td>
<td>is o and tar in the same group?</td>
<td>0.51</td>
</tr>
<tr>
<td>22</td>
<td>is o a group and lm in o?</td>
<td>0.11</td>
</tr>
<tr>
<td>23</td>
<td>the cost of producing a pointing gesture</td>
<td><strong>0.50</strong></td>
</tr>
</tbody>
</table>

Table 5.1: Features used in the installment model [1].

Table 5.1 shows the features and their associated weights. We directly adopt features and the learned weights for the first 22 features. The 23rd feature represents the cost of the gesture.
generation which we empirically set it as 0.5 (a middle value since we have no evidence to either promote or discount it) ²

The cost of a pointing gesture depends on several factors: the distance from the robot to the target object, the size of the target object, adjacency of other objects to the target object, etc. Inspired by previous work on the costs of pointing [102] and [103], we define the cost for a pointing gesture to an object as follows:

$$cost = \begin{cases} 
\log_2\left(\frac{1.5 \times \text{Distance}}{\text{Size}} + 1\right) & \text{If the object to be described is in a group} \\
\log_2\left(\frac{\text{Distance}}{\text{Size}} + 1\right) & \text{Otherwise}
\end{cases}$$

Given a situation $s$, the robot will apply the features and their associated weights in Table 5.1 to calculate $Q(s, a)$ and then choose generation action $a^*$ that maximize $Q(s, a)$. For example, In Figure 5.1(b), $R_1$ includes gesture as part of the referring expression, while $R_2$ does not.

| R1: | “Do you see a group of two on the left?” (Robot points to the group) |
| H1: | “Yes” (The group becomes the landmark) |
| R2: | “That group I was just talking about, do you see an object in the back that is on the right within that group?” (Describe the intended object in relation to the landmark) |
| H2: | “No” (Human rejects the description) |
| R3: | “That group I was just talking about, do you see an object on the right that is to the right of that group?” (Robot chooses another description strategy) |
| H3: | “Yes” |

(a) An Example of Situated Setup  (b) An Example of Referring Expressions in Installments

Figure 5.1: An example of situated setup and referential communication

**Incorporation of eye gaze as feedback.**

²Since it is challenging to conduct studies to undergo a large number of interactions with the robot to learn the weight through reinforcement learning, we decided to empirically set this weight in our initial investigation.
Previous psycholinguistic studies have shown that human eye gaze directly links with language comprehension. Immediately after hearing a referring expression, the hearer’s eyes move to the objects being referred to. Motivated by this finding we incorporate the user’s real-time gaze and verbal information as intermediate feedback in the installment model.

In order to use the human’s gaze information as feedback, we first must find how the gaze is distributed among each of the objects in the scene over the time immediately following an RE such as in Figure 5.1(b). Specifically, starting from the onset of an RE uttered by the robot, we measure at least 200 gaze readings (taking on average 7.7 sec) from the participant, telling which object in the scene their gaze is focused on. From these 200 gaze readings we calculate a distribution showing which objects are focused on the most. Based on this distribution we make two decisions: 1) which object receives most of the focus? And, 2) is the distribution on this object high enough to assume the human believes it to be the referent of the robot’s RE? We answer the first question by taking the object with the highest gaze distribution, and the second question by checking whether the object with the highest distribution is significantly (at least two times) greater than the object with the second highest distribution. Thus, our gaze processing algorithm returns an (objectId, YES/NO) tuple that can be incorporated as feedback to the installment model in various ways.

The gaze feedback can be incorporated into the installment model in three specific ways.

- **Gaze Only Feedback** In this method of incorporating the gaze feedback the robot receives the tuple output by the gaze processing algorithm. If YES, the human’s gaze is mostly focused on a single object, then the installment is successful and objectId becomes the next landmark for the next RE. Else, if NO, the human’s gaze is not focused on a single object, then the robot will try a different RE.

- **Gaze and Verbal Feedback Combined** To incorporate both the gaze and verbal information from the human as intermediate feedback, we must handle all cases where the gaze feedback
and verbal feedback conflict, such as when the human responds “yes” but the gaze indicates that no unique object was identified. We manage these cases in the following ways: when the user responds “No” then the gaze information is disregarded and the robot follows the verbal feedback. When the user says “Yes” then there are two cases: (1) if the gaze processing algorithm returns NO then the robot assumes the human understands and the robot’s intended object becomes the next landmark. (2) if the gaze processing algorithm returns YES then the robot assumes that the human believes it is talking about the object with the highest gaze distribution and it becomes the next landmark.

- **No Gaze Feedback** In this method gaze is not used as feedback to modify the behavior of the installment model. However, the gaze can be recorded and processed with the gaze processing algorithm in order to later analyze the human’s gaze behavior.

Figure 5.1 shows an example of how (verbal) feedback from the user determines which object/group becomes the landmark in the next RE. The robot starts by asking, “Do you see a group of two on the left?” After the user responds in the affirmative, that group becomes the landmark in the next utterance. The robot describes the next object in relation to this landmark, “... do you see an object in the back that is within that group?”

Note that, instead of generating a single minimum description to describe a target object (potentially with pointing gesture), our model actually plans for incrementally generating smaller episodes that lead to the target object. The planning is based on what the robot perceives from the environment and what feedback is provided by the human. Note that, when the gesture option is available, it does not mean the referring expression generated will have to use gesture. Whether the gesture is used or not depends on how it weighs against other features extracted from the environment. Although the current model is built upon the results (22 visual features) from our previous study from a 2D image, this work demonstrates that the cost of gesture can be integrated seam-
lessly with the visual features to apply to truly situated environments. As far as we know, this is the first model for referring expression generation in situated human robot dialogue that incorporates embodiment in the planning for generating episodes towards the target object.

5.2 Empirical Studies

Using the embodied collaborative models for REG described above, we conducted an empirical study. The goal was to examine the performance of embodied collaborative models in situated human-robot interaction and understand the role of embodiment in language generation.

5.2.1 Design Factors

Our experiments were designed to take into consideration the following two main issues relating to the embodiment of the REG models in a robot:

Robot’s embodied generation strategy. We want to explore to what degree gesture may help referential communication. In our experiments we specifically modeled two methods of generation.

- **Verbal Only (R-)** refers to the condition where only verbal referring expressions are generated, using the installment model. In this condition the robot will first refer to the landmark object saying “That object/group I was just talking about/you were just looking at...” Then after a short pause the robot will refer to the current object, “Do you see a ...” in some relation to the landmark.

- **Verbal and Gesture (R+)** refers to the condition where both verbal referring expressions and pointing gestures are generated, using the installment model. In this condition the robot will first verbally refer to the landmark object as in R- and may simultaneously point towards the landmark. Subsequently, the robot will verbally refer to the current object in the same
way and may also point towards that object.

Incorporating human’s feedback. We want to examine how different ways of incorporating the human’s feedback affects the performance of the installment model for REG.

- **No Feedback (N)** refers to the condition where no feedback is used. This is similar to the Episodic behavior [1], which generates an RE in a sequence of smaller noun phrases that lead to the target object. The user was not asked to provide any intermediate verbal feedback; instead the system assumes that user can correctly follow an installment before generates the next installment. Gaze is tracked for later analysis.

- **Verbal Only (V)** refers to the condition where only verbal information is used as the feedback from human. In this condition the human is allowed to respond “yes”, “no”, or “repeat” if they did not hear clearly. If the response is “yes” the robot assumes that the human understands the generated RE and the current intended object becomes the landmark for the next step. If the response is “no” the robot assumes the human does not understand the intended object and will try another RE. Lastly, if the response is “repeat” the robot repeats the last generated RE.

- **Gaze Only (G)** refers to the condition where only gaze is used as the feedback from human. In this condition the gaze of the human is incorporated as described in Section 3. The gaze is tracked and used as feedback to the installment model.

- **Gaze and Verbal (VG)** refers to the condition where both gaze and verbal information are used as the feedback from human. The gaze and verbal feedback are combined as described in Section 3.

These design factors lead to $4 \times 2 = 8$ experimental conditions.
5.2.2 Experimental Tasks

We designed an object identification game to evaluate the effects of the feedback and gesture conditions on referential communication. Figure 5.1(a) shows an example of the situated setup for the experiment. A human subject and our NAO robot were positioned in front of a table which held ten everyday objects. The subject was wearing an ASL mobile eye tracker. The NAO robot would generate referring expressions in an installment fashion to describe these objects using different strategies (depending on the experimental setting). The human subject’s goal was to identify which object the NAO robot was talking about. We compare the object identified by the human with the intended target object to compute the accuracy of a particular generation strategy.

5.2.3 Procedures

We designed four scenes with ten objects of various types in different configurations, six of which were randomly chosen as target objects for the robot. These four scenes were processed by a computer vision algorithm [85] and a perceptual grouping algorithm [79] and the results were given to the robot as its internal representations of the scenes. Here, we applied a very basic CV algorithm which resulted in rather poor performance in our data: overall, about 73% of objects in the original scene are mis-recognized or mis-segmented. We think this poor CV performance represents a more challenging problem. These four scenes were independent from each other (e.g., an apple in two different scenes may have two different recognition results). To each subject a set

---

3 Computer vision algorithms have advanced significantly in recent years. However, they are far from being perfect. It is very likely, in a new situation or changing environment, that the robot which is equipped with modern CV algorithms can only correctly recognize a few objects (not all objects) in the shared environment with the human. This is the situation our work intends to address. Therefore, we intentionally applied a very simple CV algorithm so that the misrecognition rate would be high. This is a different setting than previous work on referring expression generation.
of games was assigned, where each game had a different combination of scene, feedback condition, gesture condition, and target object.

Different from [63], Our user study was designed not for studying the role of gestures in referring communication. We designed the studies to evaluate our computational model and address how to automatically generate verbal and gesture expressions on the fly given human’s feedback and different visual configurations of the environment.

We recruited a total of 24 human subjects to participate in our study. These subjects were not aware of any part of the system or the study design. During the experiments, each participant played 24 object identification games - 6 games on each of four separate scenes. Each scene was paired with one of the four feedback conditions such that the participant interacted with the robot under each condition. Moreover, in each of the 6 games on a scene, the target object was changed. Half of these 6 games were played under the Verbal and Gesture condition (R+) and the other half under the Verbal Only condition (R-). For each participant the scenes were presented in one of the 24 possible permutations so that the order would not have an effect on the experiment. Furthermore, the orders of the targets were randomized. Based on this design each participant played three games in each of the 8 feedback-gesture conditions, resulting in 72 games total played in each condition.

5.2.4 Empirical Results

Our experiments resulted in a total of 576 object identification games based on various scenes and target objects. To help interpret the results, we reiterate some key terms that will be used throughout this section: The target object is the object the robot wants the human to identify in the end. An intended object at each installment is an object the robot refers to on the path to the target. A landmark object is the object on the path (i.e. an intended object) that has been confirmed.
5.2.5 Performance

Table 5.2 shows the identification accuracy of the target object under the eight conditions. There is a significant difference among the settings ($\chi^2 = 56.97, p < 0.0001$). A Tukey post-hoc test reveals that this difference is significant (at $p < 0.05$) between VR+ and NR-, VR+ and VR-, VR+ and VGR-, VR+ and GR-, VR- and NR+, VGR+ and GR-, VGR- and GR+, VGR- and NR+, GR+ and GR-, GR- and NR+.

The gesture with verbal feedback condition performs significantly better than most of the other models with an accuracy of 0.88. Furthermore, each model that incorporates gesture into its generation strategy performs significantly better than its corresponding model without gesture.

These results indicate the importance of embodiment in RE generation in the situated environment.

<table>
<thead>
<tr>
<th>Feedback Condition</th>
<th>Generation Strategy</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gesture Only</td>
<td>0.53</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>Gesture + Verbal</td>
<td>0.88</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Table 5.2: Comparison of accuracy for the eight experiment conditions.

5.2.5.1 The role of gesture (on accuracy)

First, we examine the role of the robot’s gesture as part of REG.

Table 5.3 shows the identification rate of the target object under the Verbal Only condition (R-) and Verbal and Gesture condition (R+). When gesture is incorporated into the REG model, the identification accuracy is much higher at 0.76 than when gesture is not used at all, 0.50. There is a significant difference between the two settings ($\chi^2 = 43.11, p < 0.0001$) and gesture was used at a
rate of 0.89 in REs in the R+ condition. Our observation is that in general, in the conditions where gesture is involved, the object identification accuracy is higher. In the R+ condition participants were able to find the intended object based on additional information to the verbal RE provided by the robot’s pointing behavior.

An example of the usefulness of gesture is apparent in the gaze feedback only condition (G). Observing the experiment under this condition, where only gaze feedback is used to decide the next landmark, several participants appeared to have difficulty understanding which object was the landmark because they were not aware of which object they had been looking at previously (The landmark is reference, “That object you were just looking at...”). In this condition gesture was useful for maintaining the common ground between the dialog participants by indicating the landmark object to the human. Overall, Gesture is a useful modality for improving the effectiveness of REs generated in the situated environment.

<table>
<thead>
<tr>
<th></th>
<th>Verbal only</th>
<th>Verbal and gesture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num. Games</td>
<td>288</td>
<td>288</td>
</tr>
<tr>
<td>Num. Success</td>
<td>144</td>
<td>220</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.50</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Table 5.3: Accuracy of object identification game, when gesture is present (R+) or absent (R-)

5.2.5.2 The role of feedback

Second we examined the effect of the different types of feedback on object identification accuracy.

We analyzed the effects of integrating the human’s real time eye gaze and verbal feedback into the installment model. Table 5.4 shows the object identification rate under the four feedback conditions.

For object identifications accuracy, there is a marginal difference among the settings ($\chi^2 = 10.81, p = 0.013$).
Table 5.4: Rate of successful object identification game, with different feedback conditions

<table>
<thead>
<tr>
<th></th>
<th>Without-Feedback</th>
<th>With-Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N</td>
<td>V</td>
</tr>
<tr>
<td>Num. Games</td>
<td>144</td>
<td>144</td>
</tr>
<tr>
<td>Num. Success</td>
<td>100</td>
<td>101</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.69</td>
<td>0.70</td>
</tr>
</tbody>
</table>

Our observation is that, when the intermediate feedback includes eye gaze (G and VG), the performance in terms of referential identification accuracy is lower than without gaze (N and V). Although we were hoping that the incorporation of eye gaze as intermediate feedback may improve the performance of the installment model, the empirical results however have shown otherwise. This is likely due to the inaccuracy of the eye gaze feedback. As a consequence of our method of measuring the subjects’ eye gaze, accidental movements of the subjects’ heads during the experiment may cause the coordinates of gaze readings to shift to objects nearby the actual focus of gaze.

The empirical results indicate our current way of incorporating gaze is not successful. For a more in-depth analysis based on our data, we separate out the cases where the feedback indicated by gaze coincides with the intended landmark object and examine whether this would be a strong signal of understanding and whether it improves the accuracy of identifying the target object in the end. Specifically, on the G and VG feedback condition, we calculate how often gaze feedback actually provides a fixed object that is used as a landmark object and how often this fixed object in fact is the intended object. Table 5.5 shows the above results.

Table 5.5 shows results from G and VG conditionbs that succeeded and failed (meaning the participant did not identify the correct referent when you asked them at the end of the game).

For G condition, the data shows that for both successful and unsuccessful games the landmark was established by the gaze feedback with about the same rate (88.6% vs. 85.1%, $\chi^2 = 1.124$, and...
<table>
<thead>
<tr>
<th></th>
<th>Gaze Only</th>
<th>Gaze + Verbal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Success</td>
<td>Fail</td>
</tr>
<tr>
<td>Num. Installment</td>
<td>185</td>
<td>235</td>
</tr>
<tr>
<td>Num. LM estd. by gaze</td>
<td>164</td>
<td>200</td>
</tr>
<tr>
<td>Rate</td>
<td>0.886</td>
<td>0.851</td>
</tr>
<tr>
<td>Num. LM estd. by gaze</td>
<td>39</td>
<td>21</td>
</tr>
<tr>
<td>and match intended obj. Rate</td>
<td>0.238</td>
<td>0.105</td>
</tr>
</tbody>
</table>

Table 5.5: Statistics for landmark and intended object in Gaze feedback only condition (G), Gaze and Verbal feedback condition (VG)

More interestingly, the rate of landmarks established by gaze which actually matched the intended object (the participant gazed at the intended object and that object became the next landmark) is twice as high in successful games as it is in failed games ($23.8\%$ vs.$10.5\%$, $\chi^2 = 11.545$, $p = 0.000679$). This shows that when the human is able to correctly identify the current intended object and the gaze feedback works properly, it leads to successful target identification.

For VG condition, we see similar statistics as with the G condition. One difference is that here the rate of landmarks established by gaze feedback (for the VG condition this only happens when the verbal response is “yes”) is much higher for successful games than failed games ($68.6\%$ vs. $52.0\%$, $\chi^2 = 13.201$, $p = 0.000279$).

One of the reasons this is lower compared to the G condition is because of the verbal feedback integration. But we are not sure why there is such a large difference for this statistic between the successful and unsuccessful games in the VG condition compared to the G condition. (There was not significant difference in the G condition $88.6\%$ vs. $85.1\%$.) We wonder if it has to do with how gaze feedback and verbal feedback is integrated.

Furthermore, as with the G condition the rate of landmarks established by gaze which actually matched the intended object (the participant gazed at the intended object and that object became
the next landmark) is greater in successful games than it is in failed games (19.3% vs. 7.9%, 
\( \chi^2 = 7.795, p = 0.00524 \)). The difference in rate is even higher here than for the G condition. Similarly, this shows that when the human is able to correctly identify the current intended object and the gaze feedback works properly, it leads to successful target identification.

### 5.2.5.3 Effects on number of installments

There is a significant difference among the feedback conditions for the average number of installment per game \( (F(3, 572) = 7.583, p < 0.0001) \). Figure 5.2 shows the average number of installments per game in each of these feedback conditions. Games in the verbal feedback condition (V) lasted slightly longer, with an average of 2.36 installments per game, than games in the no feedback condition (N), with 2.07 installments per game. Games in both the gaze only (G) and verbal-gaze combined (VG) conditions lasted significantly longer.

![Figure 5.2: The average number of installments per game in each of the feedback conditions](image)

On the other hand, statistical tests have shown that the presence of gesture does not have a significant effect on the number of installments per session.

### 5.2.5.4 Gesture on gaze indicated focus

We have also examined the role of robot’s gesture on human’s focus indicated by gaze. More specifically, we compare the proportion of different eye gaze patterns in the conditions where
gesture is present and absent. The gaze responses after each RE were classified into 3 non-mutually exclusive classes. The conditions for being classified into each of the 3 classes are listed below.

- **IntendedObject** The object with the highest distribution of gaze readings is the intended object or contained in the intended group.

- **LandmarkObject** The object with the highest distribution is (contained in) the landmark.

- **OtherObject** The object with the highest distribution is neither (contained in) the intended object/group nor the landmark, i.e., focus indicated by gaze is on objects other than the target or landmark objects.

In order to classify the gaze response, the object of the highest distribution was calculated in the same way as the gaze feedback in the G and VG conditions, except object/group with the highest distribution was not compared to the object with the second highest distribution. By non-mutually exclusive we mean, for example, it is possible for a gaze response to be classified into both IntendedObject class and LandmarkObject class if the target is contained in a landmark group.

<table>
<thead>
<tr>
<th></th>
<th>Verbal Only</th>
<th>Verbal and Gesture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Num</td>
<td>656</td>
<td>739</td>
</tr>
<tr>
<td>Num. IntendedObject</td>
<td>206</td>
<td>327</td>
</tr>
<tr>
<td>Rate</td>
<td>0.31</td>
<td>0.44</td>
</tr>
<tr>
<td>Num. LandmarkObject</td>
<td>212</td>
<td>252</td>
</tr>
<tr>
<td>Rate</td>
<td>0.32</td>
<td>0.34</td>
</tr>
<tr>
<td>Num. OtherObject</td>
<td>298</td>
<td>248</td>
</tr>
<tr>
<td>Rate</td>
<td>0.45</td>
<td>0.34</td>
</tr>
</tbody>
</table>

Table 5.6: Rate of different gaze patterns, under Verbal Only condition and Verbal and Gesture Condition

Table 5.6 shows the rate of different gaze patterns, when gesture is present or absent. There is a significant difference between the settings for IntendedObject class response ($\chi^2 = 24.29$, $p < 0.0001$) and for OtherObject class response ($\chi^2 = 20.55$, $p < 0.0001$). We can see that the use of gesture towards the intended object is associated with a higher percentage of IntendedObject
class, with a rate of 0.44 under R+ and 0.31 under R-. Moreover, the use of gesture towards the intended object is associated with a lower percentage of OtherObject class. These results show that the presence of gesture helps the human visually locate the intended object.

5.2.5.5 Gesture and gaze fixation time

Gesture also has an effect on how long it takes for the human to find the robot’s intended target. Figure 5.3 shows the effects of the presence and absence of gesture on the length of time it takes for the participant’s gaze to fixate on the intended object.

Gesture can direct the human’s gaze quickly to the intended object. We use fixation time to demonstrate this effect. Intuitively, the fixation time of a gaze response is the length of time it takes for the gaze to settle on a single object after the robot begins uttering the RE to the current object. Fixation times for successful gaze responses (those in IntendedObject class, where the intended object received the highest distribution of gaze readings) were computed by finding the distributions of the eye gaze over the objects in the scene after each gaze reading is taken. We find the length of time after the first gaze reading is taken at which the intended object obtains the highest distribution and after which no other object obtains a higher distribution than the intended object. We call this length the fixation time.

There is a significant difference between the two settings ($t = 2.43$, $p < 0.015$). The gaze fixation data indicates that the presence of gesture during the RE (R+) helps the participant’s gaze to fixate on the intended object in less time (3.73 s) than when gesture is absent (R-, 4.17 s). This is likely because the robot points in the direction of the intended object, immediately reducing the search space to a subset of objects in the scene and allowing the human to identify the object being referred to more quickly.
5.3 Discussion

One of the important characteristics of situated interaction is the embodiment. Humans and robots have physical bodies in the shared environment. Humans can refer to objects by generating a referential expression, or we can simply point to the object, or a combination of both. The decision of how to describe objects and when to incorporate gesture into referring act is complex. As an initial step in our investigation, we empirically set the weight for the gesture feature to take into account of embodiment and directly adopted the weights learned for other features [1] to address the robot’s and the human’s mismatched representations of the shared world. Undoubtedly the 2D setting in the crowd-sourcing is different from the 3D physical environment where we conducted our study. How the gesture interacts with other features to generate referring expression need to be learned in physical-world settings. We leave this to our future work.

Our empirical results have demonstrated that incorporation of gesture into the collaborative model significantly outperforms models without gestures. Our observations have shown that gestures are more useful when spatial descriptions such as “on the left/right” or “in the front/back” are used. Gestures remind the user which frame of reference is being used (human’s perspective vs robot’s perspective). Furthermore, in a scene where everything is technically in the “front” of
the user, gesture disambiguates unclear spatial relations.

Gesture is also very useful in the gaze only condition because it is difficult for the human to remember what object they were looking at previously, i.e. which is the landmark object. For example, when hearing, “That object you just looked at...”), users are not necessarily aware of the previous object that they looked at unless they catch on to the way the robot is using their gaze feedback.

Eye gaze data obtained from Mobile Eye tracking system might not be suitable to be integrated for making fine-grained decisions about which object the human is looking at. It still comes with much noise. In the conditions with eye gaze as feedback (G, VG), the rate of successful games, where the target is identifies, is lower than the conditions without eye gaze (e.g., N condition). This result seems conflicting with previous studies (e.g., [68]). However the setting in [68] is human-computer interaction mediated through a computer screen, which is very different from our situated interaction. The former represents a simplified setting where monitor-mounted gaze tracker tends to be more reliable. Better use of mobile eye track or other devices (e.g., Kinect) in physical environment needs to be explored as well as the granularity of representing gaze feedback (e.g., focused object versus focused region).

In this study we used an empirical heuristic to decide how to process the gaze of the human and use it as feedback. Often, however, the verbal response from the human does not match the gaze response. E.g., the human says, “Yes” but the gaze feedback does not indicate that the gaze has fixated on a unique object. We have recorded a large amount of gaze data from participants simultaneously with yes/no verbal feedback. In the future we could use this data in a supervised learning approach to learn how to better classify human gaze patterns.
5.4 Conclusions

In this section we develop an embodied collaborative model for referring expression generation which makes use of non-verbal modalities during interaction. We have conducted an experiment to evaluate the new embodied collaborative model through situated, physical human robot interaction and understand the role of embodiment in language generation. Our empirical results have shown that the incorporation of non-verbal modalities such as gesture into the embodied collaborative model performs better. However, the incorporation of eye gaze as the intermediate feedback for the model does not perform as well as we expected.
Chapter 6

Future Directions

6.1 Dialogue Context

Currently in our embodied collaborative model, a human’s verbal feedback is limited to simple YES/NO response. If the response is YES the robot assumes that the human understands the generated RE and the current intended object becomes the landmark for the next step. If the response is NO the robot assumes the human does not understand the intended object and will try another RE. In real natural human robot interaction, human could have different types of response, for example, human can express clarification, or they can describe what they can perceive. How to dynamically incorporate rich types of user response and dialogue context into the embodied model will be critical to REG system for practical use.

6.2 Joint Optimization

The idea of jointly optimizing a set of distinct but related sub-tasks is likely to improve the performance of systems. Specifically for the NLG system, designers have traditionally used a pipeline architecture to divide the generation process into the distinct stages of content selection, utterance planning and surface realization to choose the semantics, organization and realization of an utterance. However, this sequential model does not account for the inter-dependencies that exist among these stages. Some joint models (e.g., [104]) that consider decisions at all NLG states in
interdependence with each other could possibly obtain overall better system performance.

6.3 Online Learning

Our installment model is based on reinforcement learning, the model was trained by learning from interactions with real users on crowd-sourcing. Usually for reinforcement learning agent, it needs several thousands of interactions to learn a good policy for a domain, which makes learning from real human-robot interactions impractical. The approaches that can learn generation strategies in a fast and efficient way worth more exploration.
REFERENCES
REFERENCES


