# SIMULTANEOUSLY ACQUIRING THE SYNTAX AND SEMANTICS OF SPATIAL REFERRING EXPRESSIONS

by

Jeremy Bryan Wright



A Dissertation Submitted to the Faculty of the

DEPARTMENT OF COMPUTER SCIENCE

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2014

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Jeremy Bryan Wright, titled Simultaneously Acquiring the Syntax and Semantics of Spatial Referring Expressions and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Mihai Surdeanu — Date: 29 April 2014

Kobus Barnard — Date: 29 April 2014

Carole Beal — Date: 29 April 2014

Date: 29 April 2014

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Dissertation Director: Paul Cohen — Date: 29 April 2014

# ACKNOWLEDGEMENTS

I would like to thank Paul Cohen for taking a chance on me, and providing support, motivation, and insight throughout my graduate studies. Thanks to my graduate committee, Carole Beal, Kobus Barnard, and especially Mihai Surdeanu for their helpful comments and suggestions in preparing this dissertation. Many thanks to Ian Fasel, Clay Morrison, Tom Walsh, and Wes Kerr for providing me with opportunities to expand my research experience, as well as supplying help, encouragement and friendship during those endeavors. Thanks also to Ed Gatzke and Vincent Van Brunt for inspiring and emboldening me to pursue a doctorate in an unfamiliar field.

I thank Mathias Gibbens, Jen Dempsey, Wallace Chipidza, and Enrique Noriega, for their tireless efforts in the graduate student council, and for always being willing to help organize and attend the fun times. Thanks to Bryan Helm, Lisa Wang, and the rest of the Science Salon crew for many evenings of stimulating discussion and entertainment. Thanks to Nassim Mafi, Kevin Coogan, and the rest of the hiking club for creating many great memories of Tucson's majestic outdoors. Most of all I want to thank Derek Green, Antons & Julia Rebguns, Daniel Hewlett, Anh Tran, Naka Pilantanakitti, Dan DeBlasio, Mark Tokutomi, Sam Martin, Andrew Predoehl, Peter & Vanessa Bailey, Sankar Veeramoni, Ernesto Brau, Kyle Simek, Jinyan Guan, Kate Kharitonova, Matt DePorter, and the rest of the computer science graduate students, staff, and faculty, and their families for years of friendship, meals, drinks, conversations, movie nights, game nights and poker nights, and for being an all-around lovely bunch of people that I will miss dearly.

# DEDICATION

*Dedicated to William Gibson, Neal Stephenson, Greg Egan, and all of my favorite science-fiction authors for inspiring me to work on creating thinking machines.*

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

# LIST OF FIGURES

LIST OF FIGURES – *Continued*

# LIST OF TABLES

# ABSTRACT

To be useful for communication language must be grounded in perceptions of the world, but acquiring such grounded language is a challenging task that increases in difficulty as the length and syntactic complexity of utterances grow. Several state of the art methods exist to learn complex grounded language from unannotated utterances, however each requires that the semantic system of the language be completely defined ahead of time. This expectation is problematic as it assumes not only that agents must have complete semantic understanding before starting to learn language, but also that the human designers of these systems can accurately transcribe the semantics of human languages in great detail. This paper presents Reagent, a construction grammar framework for concurrently learning the syntax and semantics of complex English referring expressions, with an emphasis on spatial referring expressions. Rather than requiring fully predefined semantic representations, Reagent only requires access to a set of semantic primitives from which it can build appropriate representations. The results presented here demonstrate that Reagent can acquire constructions that are missing from its starting grammar by observing the contextual utterances of a fully fluent agent, can approach fluent accuracy at inferring the referent of such expressions, and learns meanings that are qualitatively similar to the constructions of the agent from which it is learning. We propose that this approach could be expanded to other types of expressions and languages, and forms a solid foundation for general natural language acquisition.

# CHAPTER 1

# INTRODUCTION

## 1.1 Overview

Historically, attempts at teaching language to machines have fallen into two camps: natural language processing (NLP), and grounded semantics approaches. NLP primarily concerns finding patterns within and between languages, such as grammars, co-reference, or translation, which may be completely separated from the world they describe and only relate to other linguistic elements. Grounded semantics, conversely, is about identifying objects, events or relations in the world, and assigning them proper linguistic labels, usually simple words or phrases.

Recently several efforts have been made to teach machines language that is both complex and grounded, such as instructions for navigating an indoor environment (Artzi and Zettlemoyer, 2013). Most of these methods require fully annotated training data, where each utterance is paired with its equivalent semantic representation, but even state-of-the-art methods that need less supervision make several problematic assumptions about the nature of the semantics behind language. First, these systems assume that an agent has access to a completely defined set of semantics from the outset of learning. However, it is quite possible that some semantics are not known ahead of time but acquired during learning to fit the usage of fluent speakers. This assumption also requires the human designers of these learning systems to have conscious access to detailed semantics for the purposes of transcribing them, when this is unlikely to be the case (exactly how far is "far"?). Secondly these systems assume that semantics can be described in terms of logical forms. However, long-standing work in linguistics show that humans do not make purely binary judgments on the truth or falsehood of statements as logical forms semantics assume, but assign them varying degrees of applicability depending on the context.

This dissertation presents a framework for learning complex grounded language for referring expressions, called REAGENT (for **R**eferring **E**xpression **Agent**). REAGENT does not use a logical form representation, instead it assumes that semantics can be represented as applicability functions which determine to what degree a word or phrase applies to a given situation. Additionally, REAGENT does not require the semantic basis of a language to be predefined, but rather it can build applicability functions from a small set of predefined semantic primitives during learning . Finally REAGENT does not require semantic annotations for training, but trains online from paired examples of referring expressions and the entities to which they refer.

Referring expressions are utterances designed to "pick out" some specific entity in the world, as in "the cup on the table". REAGENT is designed to handle multiple types of referring expressions, however the experimental evaluation focuses on spatial referring expressions. These types of expressions use spatial prepositions and other spatial language to describe the location of the referent relative to some other entity or entities, as in the example above. Spatial referring expressions exhibit enough syntactic and semantic complexity to provide a useful testbed for REAGENT's learning methods, as well as being a highly used subset of English.

### 1.1.1 Learning Approach

REAGENT's approach to learning assumes that a pre-linguistic agent must know certain things both about the purposes of language and how language works. Take, for example, a mother who when playing with her pre-linguistic child regularly picks up a wooden block and says "block". At some point in the language acquisition process the child must understand that her mother's utterance is a reference to the block in her hand. Such an understanding has several prerequisites. The child must know what objects are and be able to perceive that the block is a separate entity from her mother's hand. The child must realize that the sounds coming out of her mother's mouth might mean something, and that one of those meanings could be to "pick out" or describe the object in her hand.

If, from her previous interactions with the world, this child has already learned to differentiate blocks as their own category of objects, then she must only learn to map the word "block" to the category. However it is by no means clear that the concept corresponding to a given word (or other syntactic unit)[1] must exist in a person's mind before they can learn the word for it. Proponents of linguistic relativity assert that the opposite is true: that linguistic categories determine cognitive categories. In such cases the child must learn a meaning for the word "block" based on her observations of its usage. To do so she must have knowledge of some set of semantic primitives, as well as methods for building word meanings from them.

Later in the child's development when she is beginning to understand slightly longer phrases such as "the block" and "the red block" she similarly must realize that putting "block" together with other words changes the meaning of the utterance, and it does so in predictable ways. In other words, she must have some inkling of the compositionality of language, and what types of composition are possible. And because words are rarely encountered alone, she must be able to learn the meaning of a word even if it is composed with other words.

While Reagent is not intended to be a cognitive model of how children learn, its goal is to accomplish the same thing: learn the syntax and semantics of language from complex utterances encountered in context. Therefor Reagent assumes the same pre-requisites for learning that are posited above, namely: the ability to perceive and distinguish objects, a model of how syntax maps meanings to words as well as allows composition (of words and meaning), an understanding or what reference is for and how it works, and access to semantic primitives and methods for building word meanings out of them.

The remainder of this dissertation is organized as follows: The rest of this chapter gives background on research leading up to the development of Reagent, including the immediate predecessor to this framework described in [Dawson et al]. Chapter 2 provides an overview of literature related to applicability functions and spatial

[1]Here "word" is used as shorthand for any piece of a language's syntax that has non-decomposable meaning. This could be a morpheme, a word, or an idiomatic phrase.

semantics. Chapter 3 describes the construction-based model of communication REAGENT uses, and chapter 4 describes how this model is exploited for acquiring new constructions from examples. Chapter 5 experimentally demonstrates the effectiveness of REAGENT for both unimpeded communication and language acquisition. Finally chapter 6 discusses the implications of this work, and plans for the future.

## 1.2 Background

This section contains a brief comparison of the history of computational linguistics with the language development of children, leading up to the development of REAGENT. As the only known complete model of language learning, the human approach is worth examining to see what may be missing from computational approaches.

**Developmental Linguistics** According to (Rice et al., 2010) a child's linguistic acuity can be measured by Mean Length of Utterance Morphemes ($MLU_m$), the average number of morphemes in a child's utterances at a given age. In English, children start small, typically beginning with an $MLU_m$ of 1 around age one, uttering single morpheme words (e.g. "ma", "car", "more"). By age two they usually produce some longer words, as well as two word phrases (e.g. "car go", "apple", "more juice"), and around age three begin consistently producing three word phrases and sentences, with an $MLU_m$ of 3.5. Throughout these early stages children primarily learn language grounded in the world around them, language which refer to people, objects or activities they interact with or participate in. But around three years of age, children begin also learning concrete words by relating them to other words they know–learning by definition–without necessarily having a reference at hand.

From the age three upward, $MLU_m$ increases linearly with age, but at a lower rate than previously, reaching an $MLU_m$ of 5 by age nine. This slowdown can reasonably be attributed to the exponential increase in grammatical complexity required to increase utterance length at this stage. The more morphemes in a sentence, the more ways they could potentially be put together. Learning the correct way is no

easy task.

**Computational Linguistics** The history of machine language learning has not proceeded in exactly the same manner as children do. Early computers could handle text with relative ease, compared to richer sources of information that would enable semantic grounding such as video, audio, tactile and kinetic input. This led early natural language processing (NLP) to be based primarily in text-to-text transformations, such as the Georgetown experiment in 1954 which automatically translated 60 Russian sentences into English (Dostert, 1955). During this early period, if language was grounded at all, it was in formal logic(Kellogg, 1967, 1968; Schwarcz et al., 1970; Simmons, 1965; Simmons et al., 1968; Simmons, 1970; Thompson, 1966; Woods, 1968) regarding simulated "blocks worlds" (Winograd, 1972) and relational databases (Woods and Kaplan, 1977; Waltz, 1978), or used hand-built conceptual ontologies to interpret or generate stories (Schank et al., 1973; Cullingford, 1978; Wilensky, 1978; Meehan, 1977; Lehnert, 1977; Carbonell, 1979; Lehnert, 1981). By the 1990s, it was apparent that hand-building rules for formally manipulating text was a long and difficult process and NLP transitioned to using machine learning and statistical models to find such rules using large corpora. With these methods NLP could achieve high performance on tasks such as word-sense disambiguation (Krovetz and Croft, 1989; Veronis and Ide, 1990), parsing (Church, 1988; Brill et al., 1990; Magerman and Marcus, 1990), and speech recognition (Bahl et al., 1989). However, without corpora that included rich information about the context in which language was used, such methods could not easily induce accompanying semantics.

Around this time the increasing power and convenience of computers also allowed mobile robotics and computer vision to come into their own as fields, setting the field for work in semantics grounded in experience. Since then, the study of grounded language and semantics has followed a similar path to very young children. Facial recognition and labeling were some of the earliest applications of computer vision (Turk and Pentland, 1991; Viola and Jones, 2004). There have also been steady

advancements in recognizing and labeling manmade objects and images (Lowe, 1987, 1999; Bay et al., 2006; Rublee et al., 2011), natural objects and textures (Duygulu et al., 2002; Barnard et al., 2003), and human actions and activities (Niu et al., 2004; Robertson and Reid, 2006) in images and video. Similarly there have been efforts to recognize and label human activities from sensed (Maurer et al., 2006) or annotated acceleration data (Bao and Intille, 2004), and in robots via motor events and internal state (Peters et al., 2013).

The most common form of this vocabulary building differs significantly from that which infants do, however. In building a classifier to label instances of objects, properties or activities, discrete training examples are often presented as an example of "X" or "not X", so that the distinction is easily made. While parents do produce an abundance of speech which can be used as training examples by their child, these examples are frequently ambiguous. Pointing at a car and saying "car" does not unambiguously separate the vehicle from its parts or its surroundings, and moreover parents rarely point out examples of "not car". Additionally most child-directed utterances are more complex than a single word, and the rich source of examples that is overheard language from competent users is more complex and ambiguous still.

Several recent systems have been developed to learn complex language that is grounded in database or logical-form semantics, including Cocktail (Tang and Mooney, 2001), Scissor (Ge and Mooney, 2005), Silt (Kate et al., 2005), Krisp (Kate and Mooney, 2006), Wasp (Wong and Mooney, 2006, 2007), and several more unnamed systems of increasing sophistication (Lu et al., 2008; Zettlemoyer and Collins, 2005, 2007; Kwiatkowski and Zettlemoyer, 2010, 2011). All of these systems require fully annotated training corpora, where each utterance is accompanied by a complete semantic translation. Such systems perform well, but their training annotations require a great deal of effort on the part of humans to create, and therefor do not seem feasible in the long term due to the constantly changing nature of language. Such annotations are also subject to human's ignorance of their own semantic representations, as well as transcription errors.

A few systems have been put forward to learn complex language with much less supervision in order to avoid these issues. (Artzi and Zettlemoyer, 2013) develops a semantic parsing approach to learn a joint model of meaning and context for interpreting and executing natural language instructions. This semantic parser trains on instruction utterances for navigating an indoor environment, paired with the starting and final states that correct execution of the instructions should include. It learns a mapping from sub-expressions of an utterance to logical forms that correspond to actions, entities, relations and properties. This approach must be seeded with a set of logical forms that can be evaluated in the environment, and trains by learning which logical forms co-occur with which instruction sub-expressions. Training and validating on standard instruction set corpora, this system learns to correctly execute up to 82% of instruction utterances, and nearly 60% of instructions sequences.

(Liang et al., 2013) introduces a semantic parsing system for answering questions from a database of facts, trained only on question-answer pairs and thus not requiring semantic annotations. Unlike (Artzi and Zettlemoyer, 2013), these queries are not situated in specific environments but instead may reference any part of the given database. Therefor instead of using context to narrow the logical forms that may correspond to a given utterance, predicates are "triggered" by words with which they share a name. Semantic trees are then constructed from these triggered predicates, and as with (Artzi and Zettlemoyer, 2013) learning proceeds by observing co-occurrence between semantic forms and phrase elements from each utterance. Once trained, this system is capable of correctly answering 91% of questions in a corpus and database with a geographical domain, and up to 95% of questions in a job queries domain.

Lastly, the predecessor to Reagent, introduced in (Dawson et al., 2013), learns how to parse and interpret spatial referring expressions from only paired, situated examples of expressions and their referents. Like the instruction utterances in (Artzi and Zettlemoyer, 2013), these spatial referring expressions are encountered in a specific environment which limits the set of possible semantics that an utterance

could pertain to. For each training utterance, this system samples a semantic form from the set of all valid semantics in the scene and tracks co-occurence with syntactic structures in the utterance. Unlike the previous two systems however, (Dawson et al., 2013) treats semantics as having varying degrees of applicability, where more applicable semantics are more likely to be sampled. The applicability of a given semantic form is also used as a weight on the example during training. On a novel corpus of situated utterances and referent pairs, this system learned to recover the correct referent for slightly over 50% of utterances, approaching human performance at the same task.

All of these current approaches to learning complex grounded language make certain assumptions about the underlying semantic representations. First, they each require that the semantics of the domain be fully defined before learning can start. This amounts to assuming that an agent already fully understands how the world works before learning how to speak. While this may be true in tasks such as querying databases or executing instructions in which the underlying machinery is only capable of processing certain types of representations, in a more diverse environment an agent may not even know a given concept exists until he encounters a word or phrase that refers to it. In such cases, it is necessary for an agent to not only be able to map syntactic elements to known semantics, but to develop new semantics when called for.

A semantic learning agent would have the added benefit that his semantics would not rely on definitions provided by humans, which are unlikely to be accurate due to their lack of direct access to the specifics of their own representation. For instance it would be very difficult for a person to manually define the meaning of “far” in a way that would not only allow judgments based on distance measurements, but also fully accords with human usage.

The second assumption that each of these systems make, except for (Dawson et al., 2013), is that semantics can be properly represented using logical forms. Logical forms only support binary truth values, but research into prototype theory has shown that people commonly judge words to apply to situations with various

degrees. For instance, given multiple examples of an object being "over" a table, some examples are rated to be more appropriate usages of the word than others. This suggests that a proper representation of semantics should include some way of calculating the applicability of a word, phrase, or sentence to a given scenario.

REAGENT addresses both of these issues by providing a framework to learn complex, grounded language without assuming predefined semantics. Instead, REAGENT only assumes that a small set primitive elements are predefined from which semantic representations can be built. Furthermore, the semantic representations that REAGENT learns are allowed (but not required) to have varying degrees of applicability, which significantly complicates the learning process.

The next chapter gives a more in-depth look at literature related to this work, including prototype theory and discussions of the semantic forms and primitives that linguists speculate underlie natural language.

# CHAPTER 2

# RELATED WORK

This chapter gives a brief overview of previous research in cognitive and computational linguistics that motivates the use of applicability functions in REAGENT's model of communication. The focus here is on spatial relations, as they not only serve as the primary domain on which REAGENT's learning techniques are evaluated, but have also historically been one of the motivating domains for developing non-binary semantic representations in linguistic research. Many linguists have chosen spatial prepositions as an ideal testbed for establishing semantic representations because they are frequently used in speech, while also forming a closed set that could potentially be fully defined.

For a more complete overview of spatial relations and the semantics of referring expressions in general, please refer to (Waller and Nadel, 2012),(Geeraerts and Cuyckens, 2007), (Coventry and Garrod, 2004), or (Olivier and Gapp, 1998)

### 2.0.1 Spatial Cognition and Cognitive Linguistics

**Geometric Definitions** Early attempts at formally identify the meaning of spatial prepositions typically resulted in simple definitions based on logic and Euclidean geometry. For instance, in (Cooper, 1968), the following definition is proposed:

**IN:** X in Y: X is located internal to Y, with the constraint that X is smaller than Y–where X is the located object, and Y is the reference object

A similar definition appeared in (Leech, 1969). For each such definition, researchers soon found counter-examples and refined the definition. For example, in

(Miller and Johnson-Laird, 1976) the authors realized that for 'in', the located object need not be smaller than the containing object, nor fully contained by it. They proposed the following definition:

**IN(X,Y):** A referent X is "in" a relatum Y if:
(i) [PART (X,Z) & INCL (Z,Y)]

Where INCL means "included spatially in". Over time it seemed that singular definitions were always either too general or overly strict. Soon, linguists proposed that no single definition would suffice for any spatial preposition, and there must be either contextual changes to the definition (Herskovitz, 1986; Brugman and Lakoff, 1988; Taylor, 1986), or multiple definitions for each preposition (Lakoff, 1987; Lakoff and Johnson, 1999; Langacker, 1987).

Although stand-alone definitions proved to be inadequate to define most spatial prepositions, they did show the value of euclidean geometry concepts such as containment for such formal definitions. Most proposed semantics of these propositions, including this work, still make use of geometrical concepts.

**Radial Categories** Lakoff referred to the idea that some parts of language have many related but separate definitions as radial categories. Radial categories are separate in that, while analysts may be able to explain how each definition is related to or extended from another, it is generally impossible to predict a priori which extensions will be acceptable or commonly used. This property of human linguistic categories indicates that whatever learning methods are used in this work must be able to fit multiple models to the data.

**Prototypicality** Lakoff was also a proponent of research (begun by Roth), showing that categories in the human conceptual system are rarely black and white, as predicted by the conventional theory of categories as sets. Most categories, including spatial relations, exhibit prototypicality, meaning some members of a category are better exemplars than others. Thus, in order to conform to human usage, definitions for spatial prepositions must not only predict when a given preposition is

appropriate, but *how* appropriate it is. This observation suggests that classification functions are an insufficient basis for most spatial relations, and regression should be used to make judgments of partial applicability as well.

**Effects of Function** In a more recent development, (Coventry and Garrod, 2004) show that many seemingly disparate definitions for the same spatial preposition are often due to the functions of the objects involved. For instance, test subjects rate a couch and television as being closer to each other when the objects are facing each other as when they are facing away. In a separate series of studies, subjects rated umbrellas as better examples of 'above' when they were blocking rain from hitting a person, even if this meant not being directly over the person's head.

This research implies that encyclopedic knowledge of how humans perceive and use objects would be necessary to properly judge the applicability of spatial relations. However, gathering such knowledge is a difficult problem unto itself, and constitutes the field known as Affordance Learning. This work chooses to leave studying the effects of function on spatial relations for future work, and avoid the issue by focusing primarily on geometrically primitive objects. For unavoidable functional aspects, such as support, such knowledge will be encoded in the world representation, by using tags such as "supported by: object X".

**Cognitive Maps** One theory for the origin of spatial relations in human's conceptual systems is presented in (O'keefe and Nadel, 1978) and (O'Keefe, 1996). Studies on the hippocampus of rats show that these animals build a topological map of their environment composed of spatial relations as a compact form of representation. Such a map allows animals to reason about navigating the environment without remembering every detail of it. Thus, spatial relations could be useful not just for speaking about the world, but for thinking about it as well.

A question posed by many of those studying spatial language is: do we have concepts of spatial relations before learning language, or are they learned *through* language? The research on cognitive mapping suggests we developed spatial concepts

before language. However, some languages use non-maplike relations suggesting otherwise. In reality, we probably use some combination of pre- and post-linguistic spatial relations (depending on the language), and this work proposes to show how such a mix is possible.

### 2.0.2 Computational Linguistics

There have been several attempts to develop computational approaches to learning spatial prepositions and other spatial language.

One of the earliest computational models spatial prepositions was Regier's landmark work, (Regier, 1996). This work not only developed a method for machines to learn and understand spatial prepositions, but attempted to be a cognitively plausible account of how people do the same. This was accomplished by using a connectionist approach, similar to neural networks, which also included special nodes to compute perceptual features. The features used were inspired by cognitive research, and included: proximal and center-of-mass distance and orientation, direction of potential motion, implicit paths, and topographical features such as contact and inclusion. One of the important aspects of Regier's model, was that it did not require negative examples for learning, instead assuming that scenes with different labels are negative examples of each other.

This model was quite successful at labeling scenes, however it did not address some issues that are important for learning more complex language. Training labels were given explicitly, rather than being drawn from whole utterances, significantly reducing ambiguity. Similarly, trajectors and landmarks were clearly marked in scenes, also reducing the model's ambiguity and prohibiting the model from learning to discern object roles. In (Regier, 1996), combination weights for features were learned by the network during training. However, Regier later did a more in depth study of how two of these features, proximal and center-of-mass orientation, could be used to evaluate projective relations such as 'above', in (Regier et al., 2001). This new system of combining orientation features was called the attention-vector sum, and made use of a model of human attention called the attentional vector to

determine the weights on these two features.

(Gorniak and Roy, 2004) developed a system for robots to understand spatial language using features such as object grouping, surface areas and directional extrema, as well as instrinsic features such as color. This system used a hand-built construction-like grammar to interpret complex utterances with an impressive success rate.

A system that makes use of extra-geometric features such as dynamic-kinematic routines, and conceptual knowledge was put forth in (Coventry et al., 2005). Here, scenarios such water being poured from a teapot toward a cup were analyzed by the system to predict what will happen (whether the tea will go in the cup), and that knowledge was combined that with geometric features to make judgments of the applicability of 'over', 'under', 'above' or 'below'.

In (Spranger, 2012), not only were experiments performed for learning a grammar using a known semantic system, but also learning spatial relation semantics to fit language use, as well as co-evolving a grammar and semantic system together. In each of these experiments, small humanoid robots interacted in environments consisting of yellow blocks for trajectors, a single inanimate landmark with its relative directions marked (i.e. front, left and so on), and sometimes a landmark marking absolute directions (i.e. north). The robots used each other as landmarks as well. This work makes use of many of the same mechanisms and assumptions used in Reagent, such as representing spatial relations as applicability functions and choosing relations which uniquely, contrastively specify the chosen trajector.

This work used three types of spatial relations: proximal (e.g. near, far), relative projective (e.g. front, left), and absolute projective (e.g. north, west). The applicability functions for these relations were based on distance for proximal relations, angle for the others, and used an exponential decay function to determine applicability of a given distance or angle. These functions were fit to data using the mean and standard deviation of recorded examples of the given relation.

Unlike with Reagent, agents in this system did not choose a landmark when generating or interpreting an utterance, rather landmarks were set for any given task,

leading to a smaller sample space for agents to explore. Nevertheless, Spranger found that agents had difficulty learning both types of projective relations at the same time without some built in preference for one type or the other. While Spranger's work on learning spatial relations is quite similar to the representation learning methods implemented in REAGENT, incorporating a choice of landmarks, and solving the problem of relations that can alias each other required the development of new techniques. Additionally, Spranger only makes use of one type of applicability function (exponential decay) and does not address fitting functions to human language use, which could take on a much different form.

# CHAPTER 3

# COMMUNICATION MODEL

## 3.1 Overview

This chapter describes the model of communication underlying REAGENT's language acquisition system. Note that while the focus of this work is on spatial referring expressions, the illustrative examples in this chapter and the next are primarily simpler referring expressions for the sake of brevity and clarity.

## 3.2 Scene Model

Referring expressions occur in a context containing one or more entities, denoted the **scene**. Entities are anything that can be selected by a referring expression, for instance objects or parts of objects. REAGENT does not assume any specific model for scenes and entities, the details are left up to the representational system accompanying a given grammar. REAGENT's only requirement of the scene model is that it provides access to the set of entities contained in the scene, which is necessary for both generating and evaluating referring expressions. In the remainder of this document a scene is denoted using set notation (as a set of entities), or as $\widehat{S}$, and entities are denoted with lowercase $e$.

## 3.3 Semantic Model

### 3.3.1 Applicability Functions

The purpose of a referring expression is to select a referent from among all of the entities present in the scene. REAGENT models the semantics of referring expressions as a function that maps every entity in the scene to a value in the range $[0, 1]$. This value represents to what degree the expression applies to a given entity. Such

functions are called **applicability functions**, or **AFs**. For instance, the referring expression "the sphere" might apply with degree 1.0 to a perfectly spherical object, and to some very small degree (such as 0.0) to a cubical object. Using AF semantics, the goal of a speaking agent is to generate a referring expression that applies to the intended referent more than any other entity in the scene, and from among such uniquely applicable expressions to choose the most applicable one. For easy identification, applicability functions are denoted with a tilde, as in $\widetilde{F}$.

REAGENT's model of communication is agnostic to the inner workings of applicability functions; the details are left up to each specific grammar and its representational system. Calculating the applicability of "the sphere" to a given entity could involve something as complex as comparing a three dimensional model of the entity to a prototypical sphere and returning a value based on its conformity to that prototype, or as simple as returning a constant value regardless of the entity. However, REAGENT's methodology for learning applicability functions as described in chapter 4 does depend on certain conditions being true of a grammar's applicability functions, namely that they are calculated using a known set of features and conform to a known set of general applicability function shapes.

Using these semantics, REAGENT determines the referent of an expression by extracting the expression's applicability function, evaluating its applicability to each entity in the scene, and selecting the entity with the maximum applicability. If there are multiple entities with the maximum applicability, the agent must use some tie-breaking procedure such as choosing randomly, or asking a clarifying question. Currently REAGENT uses random selection to break ties.

### 3.3.2 AF Sets

Applicability functions may be grouped together into **AF sets**. AF sets are defined as in traditional set notation using brackets, or denoted as a bold letter with tilde, as in $\widetilde{\boldsymbol{S}}$. The applicability of an AF set to an entity is the product of the constituent

functions' applicabilities for that entity:

$$\widetilde{\boldsymbol{S}}(e) = \prod_{\widetilde{F}_i}^{\widetilde{\boldsymbol{S}}} \widetilde{F}_i(e) \tag{3.1}$$

Because the product of any number of values in the range $[0, 1]$ is also in that range, an AF set itself qualifies as an applicability function. Thus AF sets can be be evaluated as functions (as in the above example), as well as serve as the semantics of referring expressions. The collection of functions is treated as a set because the ordering of functions do not matter, and each function must be unique. The union of two AF sets is denoted with the usual $\cup$ notation. Additionally, in order to allow easy union of applicability functions and AF sets, all single applicability functions will be considered to be AF sets of size one.

### 3.3.3 Relations

Relational referring expressions, including spatial expressions, return an applicability based on a given entity's relation to another entity known as the landmark. Such expressions may include a subordinate referring expression specifying the landmark, as in the phrase "the object near to *the sphere*". Computing the applicability of this phrase requires a pair of entities, the potential referent entity and a landmark entity, and depends on the referent's proximity to the landmark.

A straightforward method of computing this applicability would be to first evaluate the landmark expression "the sphere" as described above: by computing its applicability to each object, and then choosing the maximally applicable entity as the landmark. However in the case that multiple entities have equally high applicability given the landmark expression, more care must be given to tie-breaking. If the agent breaks the tie by random selection, he may find that no entities are "near to" the chosen landmark when evaluating the primary expression. Asking a clarifying question is more likely to be successful, but also requires significantly more time and effort on the part of both the interpreting agent and the original speaker.

To avoid this dilemma, REAGENT delays making a decision on the landmark by

evaluating the *expected* applicability of a potential referent over all possible landmarks:

$$\widetilde{F}_{\widetilde{G}}(e) = \sum_{e_l}^{\widehat{S}} \widetilde{F}(e, e_l) \cdot \widetilde{G}(e_l) \tag{3.2}$$

Where $\widetilde{G}$ is the function corresponding to the landmark expression, and the subscript notation $\widetilde{F}_{\widetilde{G}}$ denotes that $\widetilde{G}$ is the subordinate applicability function to be used if $\widetilde{F}$ requires a landmark.

### 3.3.4 Compositional Semantics

As has been seen, applicability functions may be combined in two ways: by grouping them together into AF sets (either by set creation, or the union of two existing sets), or by subordinating one function to another. Linguistic structures which serve to perform these combinations have functional semantics, which are denoted using lambda calculus. For instance a structure with the function of subordinating one applicability function to another might have the semantics:

$$\lambda\, \widetilde{F}\, \lambda\, \widetilde{G}\, .\, \widetilde{F}_{\widetilde{G}}$$

Another structure might return the union of two AF sets:

$$\lambda\, \widetilde{\boldsymbol{F}}\, \lambda\, \widetilde{\boldsymbol{G}}\, .\, \widetilde{\boldsymbol{F}} \cup \widetilde{\boldsymbol{G}}$$

## 3.4 Syntactic Model

Reagent models the syntax of grammar using a construction grammar framework called Resyn. Resyn constructions have four parts, denoted as follows:

$$\text{NAME} \mid \text{class} \rightarrow \text{pattern} : \textit{sempole}$$

The sample grammar in Grammar 3.1 defines five constructions in this format.

### 3.4.1 Name

The first field, *name*, is used only for ease of reference and does not enter into any computations.

THE | det → "the" : {}
RED | Adj → "red" : $\{\widetilde{F}_{red}\}$
BLUE | Adj → "blue" : $\{\widetilde{F}_{blue}\}$
SPHERE | R → "sphere" : $\{\widetilde{F}_{sphere}\}$
CUBE | N → "cube" : $\{\widetilde{F}_{cube}\}$
SIMPLEREFEX | Refex → det N : $\lambda dn.d \cup n$
ADJREFEX | Refex → det Adj N : $\lambda dan.d \cup a \cup n$

Grammar 3.1: A simplistic grammar for referring expressions.

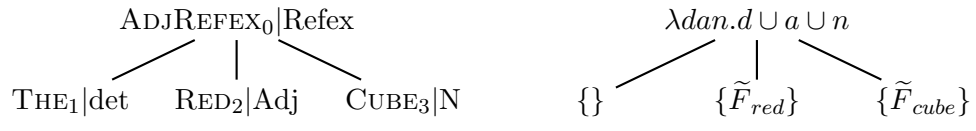
### 3.4.2 Syntactic Pole

*Class* and *pattern* constitute the syntactic pole of the construction, and are equivalent to the left-hand side and right-hand side of rules in a formal grammar. A construction's class must be a single symbol, while the pattern can consist of a text string, a sequence of class symbols or a mixture of the two, making RESYN grammars syntactically equivalent to context-free grammars. Those constructions with one or more strings in their pattern are termed lexical, while construction patterns with one or more classes are compositional; a construction may be a mix of the two.

In order to instantiate a compositional construction, each class symbol in its pattern must be matched with a construction instance of that type. These matched instances are the construction's constituents or subordinates. An instantiated construction forms a construction tree, where each constituent is a branch which may have branches of its own. Purely lexical constructions require no constituents, and thus form the leaves of such a tree. Figure 3.1a depicts an instantiated compositional construction from Grammar 3.1, which forms a two layer tree.

Taken from left to right[1], the patterns of the lexical constructions in a tree produce a valid utterance. RESYN provides a SYN() function that, when called on any

[1]Including constructions embedded in compositional-lexical constructions.




(a) Construction tree (b) Semantic tree

Figure 3.1: Construction and semantic trees for the referring expression "the red cube" using Grammar 3.1. Constructions are numbered for reference.

construction tree or subtree, returns the corresponding utterance string.

Being context free, Resyn grammar utterances can be parsed using standard context-free parsing algorithms, such as the CYK parser. However, the Resyn framework provides its own parser with the additional function of producing hypothetical parses, as described in Section 4.1.1. Both standard and hypothetical parsing take an utterance and a grammar as input, and output zero or more valid construction trees, where a valid construction tree is any tree with a single root that would generate the same utterance.

### 3.4.3 Semantic Pole

The *sempole* of a construction contains its semantic representation. Resyn places only one restriction on a construction's sempole: if a construction is compositional its sempole must be a function that can accept its subordinate constructions' sempoles as arguments. Purely lexical constructions have no subordinates, and thus may have unrestricted semantics. Resyn provides a Sem() function which takes an instantiated construction as input and returns its composite semantics. For compositional constructions, Sem() recursively calls itself on each of the construction's constituents, and then passes the returned values to the construction's sempole function. For purely lexical constructions Sem() simple returns the construction's sempole unaltered. Figure 3.1b shows the sempoles of each construction in the tree in Figure 3.1a. Figure 3.2 depicts the process of calling Sem() on this construction instance, and the value that is returned.

$\textsc{Sem}(\textsc{AdjNP}_0)$
$(\lambda dan.d \cup a \cup n)[d := \textsc{Sem}(\textsc{The}_1), a := \textsc{Sem}(\textsc{Red}_2), n := \textsc{Sem}(\textsc{Cube}_3)]$
$(\lambda dan.d \cup a \cup n)[d := \{\}, a := \{\widetilde{F}_{red}\}, n := \{\widetilde{F}_{cube}\}]$
$\{\} \cup \{\widetilde{F}_{red}\} \cup \{\widetilde{F}_{cube}\}$
$\{\widetilde{F}_{red}, \widetilde{F}_{cube}\}$

Figure 3.2: Evaluating the construction tree in Figure 3.1.

### 3.4.4 Generating utterances

All valid construction trees in a RESYN grammar can be generated by considering each of the grammar's constructions as a root node, finding all possible matches for the root's constituents, and then repeating the process with each subordinate construction recursively.[2] A speaking agent could generate a reference to a specific entity by evaluating all possible construction trees from the available grammar and choosing one which selects the intended referent. Because more than one tree in a grammar may produce the same utterance, to be unambiguous an agent must check that no other tree produces the same utterance with a different meaning. For large grammars that may generate an enormous number of valid constructions, this process is often intractable. One solution to this problem is to generate a smaller set of candidate constructions, and choose the best from among that set.

As mentioned in Section 3.3.1, the goal of a speaking agent is to choose a referring expression that applies to the intended referent more than any other entity in the scene, with the additional constraint that it should also apply to the referent as much as possible. Without this second constraint, agents might generate an expression that has very little applicability to the intended referent as long as it applies even less to all other entities. Therefore REAGENT agents choose the expression with the highest Unique Applicability (UA) for the intended referent, where the Unique

[2]If the grammar allows a construction to be a descendant of itself a limit must be set to prevent this process from recursing infinitely

| | Expression | Semantics | Object 1 | Object 2 | Object 3 | Object 1 UA |
|---|---|---|---|---|---|---|
| 1 | "the sphere" | $\{\widetilde{F}_{sphere}\}$ | 0.10 | 0.90 | 0.10 | 0.009 |
| 2 | "the cube" | $\{\widetilde{F}_{cube}\}$ | 0.90 | 0.10 | 0.90 | 0.426 |
| 3 | "the red sphere" | $\{\widetilde{F}_{red}, \widetilde{F}_{sphere}\}$ | 0.09 | 0.81 | 0.01 | 0.009 |
| 4 | "the red cube" | $\{\widetilde{F}_{red}, \widetilde{F}_{cube}\}$ | 0.81 | 0.09 | 0.09 | 0.663 |
| 5 | "the blue sphere" | $\{\widetilde{F}_{blue}, \widetilde{F}_{sphere}\}$ | 0.01 | 0.09 | 0.09 | 0.001 |
| 6 | "the blue cube" | $\{\widetilde{F}_{blue}, \widetilde{F}_{cube}\}$ | 0.09 | 0.01 | 0.81 | 0.009 |

Table 3.1: Applicability values for all Grammar 3.1 referring expressions to the objects in the scene

Applicability is defined as:

$$UA(\widetilde{F}, e, \widehat{S}) = \frac{\widetilde{F}(e)^2}{\sum_{e_i \in \widehat{S}} \widetilde{F}(e_i)} \tag{3.3}$$

## 3.5 Example Communication

This section illustrates how REAGENT agents generate and interpret referring expressions. Suppose two agents, A and B, are viewing a scene containing three objects on a table, as depicted in Figure 3.3. Both agents know grammar 3.1, and Agent A wants to refer to object 1. Usually such a reference would be part of a larger expression such as a command, but here we consider only the referring expression. Agent A starts by generating all possible constructions trees of root type Refex under grammar 3.1. This consists of the expressions shown in Table 3.1.

Agent A extracts the AF set for each expression and computes the applicabilities for each object shown, also in Table 3.1.[3] After calculating the Unique Applicability of each expression to object 1, Agent A chooses and utters "the red cube" as the most appropriate referring expression.

When Agent B hears A's utterance he attempts to parse it using Grammar 3.1. Parsing returns all construction trees in the given grammar that could produce

[3]The applicability functions in this example have not been defined, however the values in Table 3.1 are based $\widetilde{F}_{red}$ and $\widetilde{F}_{blue}$ having applicabilities of 0.9 to objects of their respective colors, and 0.1 applicability to other colors. The values for $\widetilde{F}_{sphere}$ and $\widetilde{F}_{cube}$ are similarly defined and can be seen in lines 1 and 2 of the table.

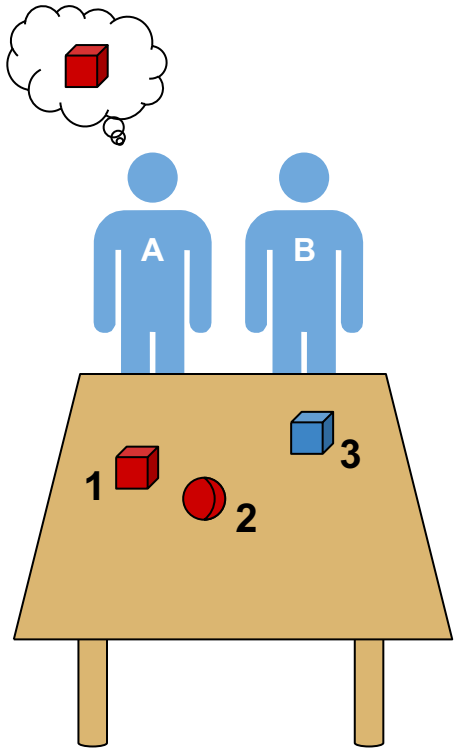



Figure 3.3: Two agents viewing a scene with three objects.

the utterance, in this case only the tree shown in Figure 3.1a. Agent B then extracts the semantics of the phrase as depicted in Figure 3.2, resulting in the AF set $\{\widetilde{F}_{red}, \widetilde{F}_{cube}\}$. He evaluates this AF set to get the applicabilities for each object, shown in line 4 of Table 3.1. The semantics of this expression apply more to object 1 than any other entity, and so Agent B interprets object 1 as the referent of the utterance.

# CHAPTER 4

# ACQUISITION

The previous chapter introduced REAGENT's model of communication, and illustrated in section 3.5 how two agents could use this model to successfully generate and interpret referring expressions if they both know the grammar in use. This chapter continues that example in order to examine the question of how this communication might fail if the interpreting agent did not have a complete copy of the grammar, and how the agent could recover from such a failure.

Suppose that Agent B is missing the RED construction from his copy of Grammar 3.1. When he hears Agent A utter "the red cube" and attempts to parse it, he discovers that he cannot find a valid parse given his grammar. Because Agent B cannot parse the utterance, he does not even know if it is a referring expression because that would require knowing the root construction, as depicted in Figure 4.1.

Agent B could attempt to recover from this failure by assuming "red" is semantically empty and ignoring it. However, with only CUBE to judge the referent by he will be unable to differentiate between objects 1 and 3.

## 4.1 Failure Modes

In general there are four ways interpreting an utterance could fail:

1. Failure to parse due to:
   (a) Malformation by the speaker
   (b) Incomplete grammar of the listener
2. Failure of expression's semantics to select intended referent due to:
   (a) Poor choice of expression by speaker

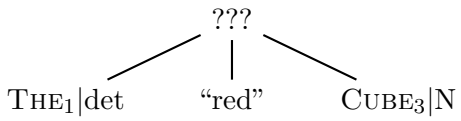



Figure 4.1: An incomplete construction tree for the utterance "the red cube".

(b) Incomplete grammar OR incomplete semantics of the listener

REAGENT does not currently try to recover from failure modes 2a or 2b, although potential methods to do so are discussed in chapter 6.

However REAGENT does try to recover from the failure to parse an utterance by learning a new construction that would make it parseable. REAGENT does not try to distinguish between malformations and missing constructions, because either a malformation is infrequent and attempting to learn it will cost little, or it happens frequently and is thus equivalent to a missing construction.

REAGENT addresses missing constructions in two stages: hypothesizing the syntax of a new construction that would yield a parse for the given utterance, and trying to learn an AF set for the proposed construction's sempole.

### 4.1.1 Hypothetical Parsing

Hypothetical parsing is the process by which the syntax of new constructions are proposed that would yield a valid parse for an otherwise unparseable utterance.

During standard operation the RESYN parser builds construction trees iteratively from the the leaves upward, starting with the utterance string, matching lexical constructions to the string and then compositional constructions to the sequence of lexical constructions. It halts when either no constructions match the current sequence, or the sequence is reduced to a single construction. During hypothetical parsing, rather than discard sequences for which no more matches can be found the RESYN parser searches for constructions that partially match the sequence. Specifically, it accepts only partial matches for which all but one element of the construction's pattern is matched, and at least one element of the current sequence exists in that gap. If there were a construction with the same type as the

unmatched pattern element and a pattern equivalent to the unmatched elements of the subsequence then parsing could continue. The parser therefore inserts a hypothetical construction with this syntax into the sequence, and continues. If this sequence can successfully finish parsing (without requiring additional hypothetical constructions), the proposed construction may be valid.

In the example of "the red cube", because Agent B is missing the RED construction from his grammar, the standard parser would be unable to find any matches for the string "red", and would stall out at the sequence:

$$\text{THE|det "red" CUBE|N}$$

However, ADJREFEX partially matches this sequence:

$$\text{ADJREFEX} \mid \text{Refex} \rightarrow \textbf{det Adj N} : \lambda dan.d \cup a \cup n$$

Where only the Adj part of its pattern does not have a matching construction in the parse sequence and at least one part of the sequence (the string "red") is in that gap. Therefore the parser hypothesizes a construction with the signature:

$$\text{HYP1} \mid \text{Adj} \rightarrow \text{"red"} : \{\dots\}$$

This is in fact the signature of the missing construction. However, hypothetical parsing would propose other constructions as well, such as:

$$\text{HYP2} \mid \text{N} \rightarrow \text{"red cube"} : \{\dots\}$$

Which could be composed into a valid tree using SIMPLEREFEX.

### 4.1.2 Ranking Hypothetical Parses

The shorter a hypothetical construction's pattern is, the more of the utterance must be covered by the other constructions in the tree, which are known to be valid. Thus, for any set of hypothetical parses returned by the parser, the length of each hypothetical's pattern can be used as a proxy for judging parse validity. This hypothetical construction evaluation criterion is called the **hyp-width**, and is used

by REAGENT for ranking hypothetical parses. In the case of our example, HYP1 has a shorter pattern than HYP2 and so the parse containing HYP1 would be ranked higher.

Such a ranking is useful if REAGENT has limited time or resources. The semantic learning process described in section 4.2 is often time consuming, and REAGENT can prioritize learning meanings for those hypothetical constructions expected to be more valuable. Likewise if an interpreting REAGENT agent needs to act on the referring expression, for instance by picking up the referent, he must choose which parse or parses to use for interpretation. The fastest method is for the agent to choose one parse to interpret, such as the one with the highest priority (lowest hyp-width). However an agent could also interpret multiple parses and choose the entity with the highest average applicability over their semantics, where the average could also be weighted according to the (inverse) hyp-width in order to give more heed to the more likely parses.

It should be noted that hyp-width ranking is heuristic, and not guaranteed to prioritize the the correct construction. For instance, "red panda" does not have the same meaning as the composition of "red" and "panda", and "red cube" could be similarly non-decomposable. In such cases the shortest hyp-width construction may be the wrong choice.

Hypothetical parses fall into three categories depending on what is missing from the grammar: purely lexical, purely compositional, or a hybrid of the two.

In Figure 4.2a a lexical construction for the pattern "bar" is hypothesized to be missing, so a lexical construction is proposed to allow constructions C and A to complete the parse. A purely lexical construction must have an AF set as its sempole, and so this case requires AF learning.

In Figure 4.2b a compositional construction that would compose C and D together is hypothesized to be missing from the grammar, and so a compositional construction is proposed to complete the parse. This construction should have a functional sempole that combines the AF sets of its subordinates.

In Figure 4.2c the grammar is either missing a both a lexical construction for

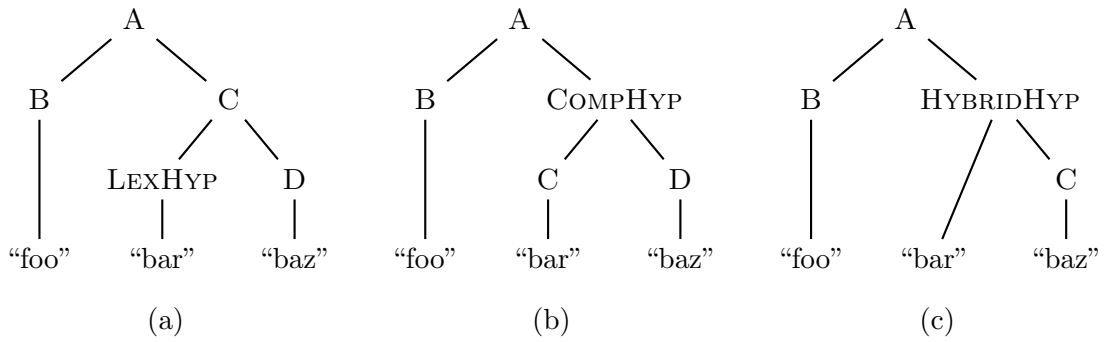



Figure 4.2: The three types of hypothetical constructions

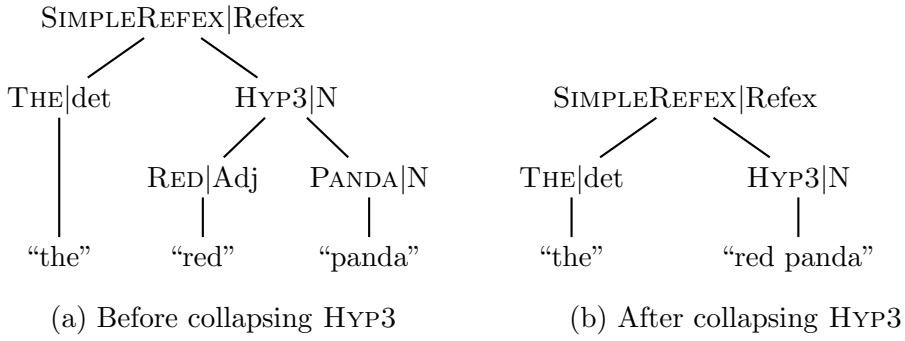



Figure 4.3: Example of collapsing a hypothetical construction.

"bar" and a compositional construction to combine it with C, or it is missing a hybrid construction that produces a different meaning for "baz" when it is preceded by "bar", as in the "red panda" case.

Reagent currently only supports lexical learning, that is learning an AF set to complete a proposed lexical construction, and so can only handle lexical hypothetical parses. However, both the compositional and hybrid cases can be converted into lexical constructions by collapsing their subtree: replacing their proposed patterns with the concatenated patterns of their subordinates. This process is depicted in Figure 4.3. This allows Reagent to handle all three types of hypothetical parses in the same manner.

## 4.2 Learning AF Sets

### 4.2.1 Lexical Learning

Given a parse containing a hypothetical lexical construction, REAGENT attempts to learn an AF set to serve as its sempole. REAGENT's complete algorithm for interpreting utterances and learning new constructions is shown in Figures 4.4, 4.6, and 4.7. Learning is only possible under certain conditions. First, the learning agent must have some way of knowing the intended referent. This could occur for many reasons, such as the listener following the speaker's eye gaze, or the speaker gesturing to the referent.

Because the purpose of a referring expression is to apply maximally to the intended referent, REAGENT assumes that each construction in the utterance both applies to the referent, and "rules out" some subset of the other entities in the scene by applying less to them. For instance in the example at the beginning of this chapter, in order to refer to object 1, Agent A utters "the red cube" which has the composite semantics $\{\widetilde{F}_{red}, \widetilde{F}_{cube}\}$. $\widetilde{F}_{red}$ applies equally well to objects 1 and 2, but has very low applicability to object 3. Similarly $\widetilde{F}_{cube}$ applies equally well to objects 1 and 3, but not to object 2. Each function is required for the composite AF set to apply maximally to object 1.

However, in this example Agent B is missing the RED construction from his grammar. He has already hypothesized that a construction of the form:

$$\text{HYP1} \mid \text{Adj} \rightarrow \text{“red”} : \{\dots\}$$

would allow him to parse the utterance, and is now attempting to learn an AF set that corresponds to this construction. If Agent B starts by assuming this construction is semantically empty, he can evaluate the parse to find it has the composite semantics $\{\widetilde{F}_{cube}\}$. After calculating the applicability of this AF set to the objects in the scene, he finds that it applies equally well to objects 1 and 3. Thus Agent B must either guess that one is the referent and point to it, or indicate that he does not understand. If he points to the correct referent, he can expect confirmation

that he is correct from Agent A. If he chooses the wrong object or indicates misunderstanding, he can expect Agent A to point out the correct referent. Either way, Agent B is likely to find out that the true referent is object 1.

Given the true referent, Agent B can logically assume that the missing construction is more applicable to object 1 than object 3. Thus object 1 could be considered a positive example of RED, and object 3 a negative example. Object 2 could also be a negative example of RED as far as Agent B knows. In general, REAGENT always treats the true referent as a positive example and all other entities as negative examples, but gives each example a weight equal to the inverse applicability of the incomplete utterance, $1 - \widetilde{\boldsymbol{U}}(e)$ (where $\widetilde{\boldsymbol{U}}$ is the utterance semantics sans the construction in question, and $e$ is the example under consideration). Thus Agent B's first encounter with RED yields two datapoints which may or may not be enough to learn an accurate representation of its semantics. However, future encounters with this construction will provide more data.

The second requirement REAGENT has for learning AF sets is that the grammar's applicability functions must be computable using a known set of feature functions and a set of general, parameterized functions that map feature values to the range $[0, 1]$. For example, a relation such as "near to" may be based on the distance between the referent and a landmark. In this case the function that measures this distance is a feature function, and distance could be mapped to $[0, 1]$ using a sigmoid function that returns high values for small distances and low values for larger distances (the slope and location of the curve being controlled by some set of parameters).

Formally REAGENT's AF set learning requires a set of feature functions, $\mathbf{E}$, a set of general mapping functions, $\mathbf{M}$, and two weighted sets of positive example entities, $\mathbf{P}$, and negative examples, $\mathbf{N}$. This algorithm searches for an applicability function that assigns high applicability to the feature values of the positive examples, and low probability to the negative ones. The search involves taking every pair of feature function and mapping function, and finding the parameters for the mapping function that maximize its applicability to the positive example features and minimizes its

applicability to the negative example features, by minimizing the weighted average binomial error (ABE):

$$ABE(< E, M_p >, \mathbf{P}, \mathbf{N}) = \frac{\sum_{e \in \mathbf{P}} M_p(E(e)) + \sum_{e \in \mathbf{N}} 1 - M_p(E(e))}{|P| + |N|} \quad (4.1)$$

Where $M_p$ is a mapping function with parameters $p$, and $E$ is a feature function. Once the optimal parameters for each feature/mapping function pair $< E, M_p >$ has been determined, the pair with the lowest ABE is chosen as the first applicability function in the set.

An optimal AF set may include more than one applicability function. After learning one applicability function, a secondary function can be learned by reweighting the example data according to the inverse of the utterance's applicabilities (including the new applicability function), and repeating the process. However, repeated indefinitely this will result in overfitting to the available data by adding a large number of applicability functions. To counter this, REAGENT performs a k-folds cross-validation on each extension of the AF set to determine what number of functions to include to provide the optimal applicability while still generalizing to unseen examples.

Each utterance and referent pair only provides one positive example and some small number of negative examples of a given hypothetical construction, and is thus unlikely to provide enough data to learn an accurate semantic representation. Therefore each time an unparseable utterance and its associated referent is known, REAGENT performs this analyses for the top $n$ hypothetical constructions (where, in practice $n$ is 5), and stores the weighted, (positive or negative) labeled feature vectors of those constructions in a database. If one of these hypothetical constructions is encountered again, REAGENT can then fetch all associated datapoints and use them as training examples.

**input** : $u$, a referring expression utterance
$\widehat{\mathbf{S}}$, the set of entities in the scene
$D$, the database of previous observations
$\mathbf{E}$, the set of feature extraction functions
$\mathbf{M}$, the set of parameterized applicability functions
$n$, # of hypothetical parses to analyze

**output**: $a$, the inferred referent of $u$

$p \leftarrow$`Parse(`$u$`)`

**if** $p$ **then**
  $a \leftarrow \underset{e \in \widehat{S}}{\operatorname{argmax}} \textsc{Sem}(p)(e, \widehat{S})$

**else**
  $\mathbf{H} \leftarrow$`HypParse(`$u$`)`
  **if not** $H$ **then return** *Failure*
  sort $\mathbf{H}$ by hyp-width
  $H_0 \leftarrow$`CompleteSemantics(`$H_0$,$\mathbf{E}$,$\mathbf{M}$`)`
  $a \leftarrow \underset{e \in \widehat{S}}{\operatorname{argmax}} \textsc{Sem}(H_0)(e, \widehat{S})$

point to $a$ as interpreted referent

$r \leftarrow$true referent (either $a$ or another entity)

**if H then** `/*` $u$ `was unparseable so store examples for learning */`
  **for** $i \leftarrow 1$ **to** $n$ **do**
    $D \leftarrow$`Observe(`$D, H_i, \widehat{\mathbf{S}}, r, \mathbf{E}$`)`

Figure 4.4: `Interpret` This algorithm interprets a referring expression to determine the referent. If the utterance is not parseable with the standard parser, it attempts to learn one or more new constructions and then gather data for future learning.

**input** : $h$, the hypothetical parse to learn
$D$, the database of previous observations
**E**, the set of feature extraction functions
**M**, the set of parameterized applicability functions
$N$, maximum size AF set to build

**output**: an AF set that optimizes observations of $h$

$c \leftarrow$ the hypothetical construction in $h$
$\mathbf{O} \leftarrow$ `Fetch(`$c$,$D$`)`; `fetch all observations for c`
$\widetilde{F}_0 \leftarrow \{\}$
**for** $i \leftarrow 1$ **to** $N$ **do**
  $best \leftarrow \infty$
  $\widetilde{F}_{best} \leftarrow null$
  **for** $f \in$ **E do**
    **for** $m \in$ **M do**
      $weights \leftarrow O_{weights} \cdot (1 - \widetilde{F}_{i-1}(\mathbf{O})$
      $p \leftarrow$ parameters to minimize `ABE(`$m_p(\mathbf{O}_f), O_{labels}, O_{weights}$`)`
      $err \leftarrow$`ABE(`$m_p(\mathbf{O}_f), O_{labels}, O_{weights}$`)`; `Average Binomial Error`
      **if** $err < best$ **then**
        $best \leftarrow err$
        $\widetilde{F}_{best} \leftarrow m_{p,f}$
  $\widetilde{F}_i \leftarrow \widetilde{F}_{i-1} \cup \{F_{best}\}$
choose best $\widetilde{F}_i$ using k-fold cross-validation
`SetSempole(`$c$, *best* $\widetilde{F}_i$`)`
**return** $h$ *with modified* $c$

Figure 4.6: `CompleteSemantics`

**input** : $D$, the database of previous observations
$h$, hypothetical parse of utterance
$\widehat{\mathbf{S}}$, the set of entities in the scene
$r$, the true referent
$\mathbf{E}$, the set of feature extraction functions
**output**: $D$, the database including the new observations

$c \leftarrow$ the hypothetical construction in $h$
$\widetilde{\boldsymbol{F}}' \leftarrow \textsc{Sem}(h)$ ; `extract AF set from` $h$ `while` $c$ `semantically empty`
$\widetilde{\boldsymbol{G}} \leftarrow$ subordinate AF set of $\widetilde{\boldsymbol{F}}'$

**for** $e \in \widehat{\mathbf{S}}$ **do**
  **if** $\widetilde{\boldsymbol{G}}$ **then**
    **for** $l \in \widehat{\mathbf{S}}$ **do**
      $\mathbf{o} \leftarrow$ new observation vector
      $\mathbf{o}_{construction} \leftarrow c$
      $\mathbf{o}_{label} \leftarrow e = r$
      $w \leftarrow \widetilde{\boldsymbol{F}}'(e) \cdot \widetilde{\boldsymbol{G}}(l)$
      $\mathbf{o}_{weight} \leftarrow$**if** $e = r$ **then** $w$ **else** $1 - w$
      **for** $f \in \mathbf{E}$ **do**
        $\mathbf{o}_f \leftarrow f(e, l)$
      $D \leftarrow$`Store(`$\mathbf{o}$`,`$D$`)`
  **else**
    $\mathbf{o} \leftarrow$ new observation vector
    $\mathbf{o}_{construction} \leftarrow c$
    $\mathbf{o}_{label} \leftarrow e = r$
    $w \leftarrow \widetilde{\boldsymbol{F}}'(e)$
    $\mathbf{o}_{weight} \leftarrow$**if** $e = r$ **then** $w$ **else** $1 - w$
    **for** $f \in \mathbf{E}$ **do**
      $\mathbf{o}_f \leftarrow f(e)$
    $D \leftarrow$`Store(`$\mathbf{o}$`,`$D$`)`

Figure 4.7: `Observe`. This algorithm gathers the positive and negative example feature vectors for a given hypothetical construction and stores them in a database for future learning.

# CHAPTER 5

# EXPERIMENTAL MEASUREMENTS

This chapter describes several experiments designed to test the methods described in the previous chapters, as well as compare them to a simple baseline method for interpreting referring expressions (described in section 5.2). These experiments all take place in the context of a game played between two agents where one agent, the Speaker, generates a referring expression and the other agent, the Listener, tries to interpret that expression and non-verbally indicate the proper referent. If the Listener indicates the wrong entity, the Speaker corrects him by non-verbally indicating the correct referent. In a physical setting, non-verbal indication may be accomplished by pointing, however as these are software agents they simply transmit the scene model ID of the indicated entity instead. This game is called the Referring Game.

This chapter contains two series of experiments: one in which the Speaker is an instance of REAGENT, and one in which descriptions come from human speakers. The first series is divided into four sets, where the experiments within a set share the same grammar. In each set the given grammar is first tested to determine how successfully the Listener can interpret referring expressions with a complete copy of the grammar, and in the remaining experiments some number of constructions are removed from the grammar and the Listener must learn replacements for them. In each experiment the interpreting agent's accuracy at choosing the correct referent is used as a quantitative measure of success. In learning experiments, the constructions learned by the Listener are also qualitatively compared to the constructions that were removed from his grammar. Because REAGENT is used on both sides of the conversation, the Speaker's grammar and semantic system is known, making it possible to show whether the Listener learns semantic representations similar to those that were used for generating the descriptions.

In the second series of experiments, rather than referring expressions being computer-generated they are gathered from human speakers via Amazon's Mechanical Turk. In these experiments the Listener will use the same grammars as in the first series, and again will have certain constructions removed for each experiment. This is designed to test whether REAGENT can learn useful meanings from observing human expressions.

As described in the previous chapters, REAGENT's communication model does not specify the details of how scenes and applicability functions are modeled and implemented, which allows it to accommodate a variety of representations. While grammars may generally each have their own associated representation, the grammars in this chapter all share the scene model and applicability function representation described in the next section.

## 5.1 Semantic Model

### 5.1.1 Scene

For the purposes of these experiments, the scene is modeled in two dimensions as viewed from overhead. Each scene is defined by a pair of coordinate axes and a set of objects.

Each object is modeled as an axis-oriented bounding box and has two fields that declare the object's shape (e.g. 'sphere') and color (e.g. 'red'). Objects may also have parts such as edges, corners, and a middle. An edge is represented as a line-segment (the actual edge) with an attached rectangular area representing the edge surface (within which objects might be considered 'on' the edge). Corners and middles are represented as points also with attached rectangular areas, where the point is located at one of the rectangle's vertices or at its center, for corners and middles respectively. The scene objects and their parts constitute the set of entities to which an agent may refer. Each scene additionally contains a vector representing the viewpoint of all agents observing the scene.

In the experiments below, all scenes contain one rectangular table (a large object

of shape 'table' with no color), which has four edges, four corners and a middle, some number of smaller objects of various shapes and colors, and a viewpoint near the table, centered on one of its edges and pointing toward its center. These scenes are constructed so that the scene's axes are aligned with the table's axes, producing a bounding box for the table of the highest fidelity. The scenes in these experiments are all randomly generated and constructed directly in two dimensions, however in general scenes may be constructed from a physical model as observed by visual sensors, or constructed from a three-dimensional computer model.

### 5.1.2 Applicability Functions

As required by the communication model, semantics are modeled using applicability functions and AF sets. As required by the learning model, these applicability functions are implemented using a set of feature extraction functions and a set of general, parameterized applicability functions that can map feature values to the range $[0, 1]$.

#### Features

The model includes six features with which to build applicability functions. There are three unary features, meaning they can be computed from a single entity. The feature functions that extract these return discrete values denoted by strings.

**shape** - 'block', 'sphere', 'cylinder', or 'table'
**color** - 'red', 'green', 'blue', 'black' or 'white'
**rep** - 'box', 'edge', 'corner', or 'middle'

There are three binary features that can be computed between any two entities, A and B. These features return a real number unless otherwise stated:

**part-of** - whether A is part of B, boolean
**distance** - the edge-to-edge distance from A to B
**overlap** - the fraction of A's area that overlaps B

There is one trinary feature, computed from the center of two entities A and B, and the agent viewpoint V:

**angle** - the angle in degrees from vector $V - B$ to $A - B$

However, because there is only one agent viewpoint per scene, only two entities must be specified to compute this feature, making it effectively binary.

### Applicability Function Types

This representation has three types of general, parameterizable applicability functions: discrete, sigmoid, and centroid. The feature used by an applicability function is denoted with a subscript, shown below using a generic $x$. A discrete applicability function is an explicit mapping between feature values and numbers in the range [0,1], denoted as a list of pairs:

$$[\text{`val1'} : \textit{app1},\ \text{`val2'} : \textit{app2},\ \cdots\ \text{`valN'} : \textit{appN}]_x$$

Discrete AFs default to 0 for any value not in the map.

A sigmoid AF is a function of real values defined by a logistic curve, which asymptotically approaches 0 at one end of its domain, and 1 at the other end:

$$Sig_x(loc, shape) = \frac{1}{1 + e^{-\pi \cdot \frac{x - loc}{shape}}}$$

The *loc* parameter determines the x-value of a sigmoid's inflection point, the same value for which it outputs 0.5 applicability. The *shape* parameter determines both the direction and grade of the sigmoid's slope.

A centroid AF is a function of real values with the shape of a logistic distribution, but scaled to a height of 1:

$$\text{Cen}_x(loc, shape) = e^{-\frac{1}{2} \cdot \left(\frac{x - loc}{shape}\right)^2}$$

A centroid function's *loc* parameter determines its centerpoint. The *shape* parameter determines the width of the bell, and must be positive.

### Parameter Fitting for Applicability Functions

REAGENT's learning algorithm requires methods to be provided for choosing parameters for these general applicability functions to optimize the applicability they assign to a set of positive and negative feature-vectors. Each of the fitting algorithms described below makes use of the average binomial error function given in equation 4.1.

Discrete applicability functions are fit to example data in the same manner as learning decision stumps. For a given feature, the parameter fitting algorithm considers every proper subset of feature values as a possible "in" set, and calculates the error of labeling examples with a feature value in that subset with an applicability of 1.0, and examples with other values as having an applicability of 0.0. The subset that generates the least error is kept.

Sigmoid and Centroid applicability functions can be fit with standard numeric regression techniques using the given error function. However, such regression techniques generally require an estimate of the parameters from which to start. These parameters are estimated for centroid fitting by taking the mean ($\mu$) and standard deviation ($\sigma$) of the positive datapoints, where *loc* is equal to $\mu$ and *shape* to $\sigma \cdot \sqrt{3}$. For sigmoids, because the function is the integral of a centroid function, these parameters are estimated by taking the mean and standard deviation of the midpoints between neighboring positive and negative datapoints (as an approximate derivative) and again *loc* is equal to $\mu$ and *shape* to $\sigma \cdot \sqrt{3}$. These techniques may also be used without regression as they produce similar results to regression fitting and are much faster to compute.

## 5.2 Baseline: Pointwise Mutual Information

In order to accurately judge REAGENT's performance in these experiments it must compared to other methods for choosing referents. However, while there are several powerful semantic parsing systems capable of discovering new syntax and learning their semantics by mapping them to a composition of basic boolean functions, none

of them can build those functions from real-valued features at the same time. So to serve as a baseline, a simpler method for this task was designed using Pointwise Mutual Information (PMI), a statistical method for determining how likely a pair of outcomes are to occur together. PMI is commonly used for natural language tasks such as collocation extraction, (Bouma, 2009), sentiment or affect analysis (Recchia and Jones, 2009; Read, 2004), and other types of implicit feature identification (Su et al., 2006). In this case, PMI could be used to determine which potential referent's features are most likely to co-occur with the words of a given utterance, using previously observed pairs of feature vectors and utterances as evidence. The entity with the highest mutual information with the utterance can then be chosen as the interpreted referent.

In general the pointwise mutual information between two occurrences $A$ and $B$ is:

$$PMI(A;B) = \log \frac{p(A|B)}{p(A)} = \log \frac{p(B|A)}{p(B)} \tag{5.1}$$

Because referring expressions pick out an entity based on its properties and relations to the scene, the utterance should have high mutual information with the referent. The PMI of an utterance $u$ (treated as a bag of words, $w$) and a given entity $e$ and landmark $l$ (which can be null) is calculated as in Equation 5.6 below:

$$\begin{aligned}
PMI(u;<e,l>) &= \log \frac{p(<e,l>|u)}{p(<e,l>)} && (5.2)\\
&= \log p(<e,l>|u) - \log p(<e,l>) && (5.3)\\
&\approx \log \left( \prod_{w \in u} \prod_{f \in \mathbf{E}(<e,l>)} p(f|w) \right) - \log \left( \prod_{f \in \mathbf{E}(<e,l>)} p(f) \right) && (5.4)\\
&= \left( \sum_{f \in \mathbf{E}(<e,l>)} \sum_{w \in u} \log p(f|w) \right) - \left( \sum_{f \in \mathbf{E}(<e,l>)} \log p(f) \right) && (5.5)\\
&= \sum_{f \in \mathbf{E}(<e,l>)} - \log p(f) + \sum_{w \in u} \log p(f|w) && (5.6)
\end{aligned}$$

Where $\mathbf{E}(< e, l >)$ is the set of features extracted from a given entity-landmark pair. The marginal ($p(f)$) and conditional ($p(f|w)$) probabilities of a feature $f$ can be estimated using the frequency of $f$ occurring or co-occuring with a word in the database of observations $D$. However for continuous features it is very likely that a particular value will not exist in the database at all, or in too few samples to accurately approximate probability. For continuous features this probability is instead estimated using kernel density estimation (KDE) over all of those feature values that do occur or co-occur in $D$. In order to get good estimates for potentially multi-modal data kernel density estimation by diffusion as described in (Botev et al., 2010) is used, but because this method fails for very small numbers of datapoints, in such degenerate cases the more common method as described in (Samiuddin and El-Sayyad, 1990) is used.

### 5.2.1 Two-pass PMI

In order to calculate PMI a database of observed pairs of feature-vectors and utterances must be stored. In the experiments in this section, these feature vectors are calculated from the true referent of each utterance as it is encountered in training. However for spatial referring expressions, such as "the object near the red cube" the proper landmark for the utterance must also be known in order to calculate the binary features for the observation, but the landmark is not revealed during training. Naively, this means that the Listener must store a feature vector for every possible landmark, as it is unknown which entity-landmark vector should actually be paired with the utterance. Because of this PMI is unlikely to be successful at interpreting spatial referring expressions.

In order to better prepare PMI as a baseline method for interpreting referring expressions a two pass version was developed. In this formulation, a second database of observations, $L$, is available containing only observations (including feature vectors) of landmark referring expressions such as "the red cube" or "the middle of the table". Given a spatial referring expression, such as "the object near the red cube", $L$ could be used to calculate the mutual information between this utterance

and each potential landmark in the scene, and then normalized in order to get a probability of each entity being the correct landmark. A second pass of PMI can then be calculated using the database of spatial referring expression observations begin built during training, $D$, to determine which entity-landmark pair has the highest mutual information weighted by the prior probability of its landmark. And of equal importance, once the true referent is known all of the observations which include that referent can be stored in $D$ along with the probabilities of their landmark, and these weights will be used in future calculations of PMI. The equation for PMI remains as defined in Equation 5.6, however this means the calculation of the marginal and conditional probabilities will be estimated from weighted frequencies in the database, and for continuous features, weighted kernel density estimation.

## 5.3 Experiments

In each of the experiments in this section a Speaker agent generates referring expressions using a given grammar, and the Listener tries to interpret the expressions and choose the correct referent. After the Listener indicates which entity he believes is the referent, the Speaker either tells the Listener that he was correct, or he indicates the true referent of the utterance. This allows every example to serve first as test point for the Listener's comprehension, and after receiving the correct answer it is used again as a training example.

### 5.3.1 Experiment Set 1: Intrinsic Referring Expressions

The first set of experiments test REAGENT's ability to both communicate and learn constructions in a grammar of simple intrinsic referring expressions. This type of expression accomplishes reference using intrinsic properties of the referent, specifically their shape or color, as in "the sphere" or "the blue cylinder". The representation of color and shape features in this grammar is very simplistic (each entity has a field giving its shape and color as strings) and so the applicability functions for the associated words are simplistic as well.

A more accurate treatment of color and shape semantics would require a much more involved representational system, including color spaces, lighting, three dimensional models and a more elaborate set of features to go with that representation, ground that has already been covered by many studies in machine perception and classification. However, some form of intrinsic reference is required for the formation of spatial referring expressions which are the focus of this evaluation. This discrete-featured version of intrinsic reference is ideal, then, because it not only provides an unambiguous basis on which to build spatial referring expressions, but it also provides a simple first-pass test case for examining REAGENT's functionality for both communication and learning.

### Intrinsic Refex Grammar

Grammar 5.1 has three types of lexical constructions: determiners, color adjectives, and shape nouns. There is a single determiner, THE, which is semantically empty. Nouns can form a noun phrase (NP) either alone, as in the SIMPLENP compositional construction, or with an adjective phrase (AdjP) to make using the ADJNP construction. An adjective phrase can be a single adjective alone, or an adjective together with another adjective phrase. Finally the IREFEX construction combines a determiner and a noun phrase to make an intrinsic referring expression.

THE | Det → "the" :

RED | Adj → "red" : {['red' : 1.0]$_{color}$}
GREEN | Adj → "blue" : {['green' : 1.0]$_{color}$}
BLUE | Adj → "green" : {['blue' : 1.0]$_{color}$}
WHITE | Adj → "white" : {['white' : 1.0]$_{color}$}
BLACK | Adj → "black" : {['black' : 1.0]$_{color}$}

BLOCK | N → "block" : {['block' : 1.0]$_{shape}$}
SPHERE | N → "sphere" : {['sphere' : 1.0]$_{shape}$}
CYLINDER | N → "cylinder" : {['cylinder' : 1.0]$_{shape}$}

NP | NP → N : $\lambda n.n$
ADJNP | NP → AdjP N : $\lambda a.\lambda n.a \cup n$
ADJP | AdjP → Adj: $\lambda a.a$
TWOAP | AdjP → Adj Adj: $\lambda a_1.\lambda a_2.a_1 \cup a_2$

IREFEX|RefEx → Det NP : $\lambda d.\lambda n : d \cup n$

Grammar 5.1

**Experiment 1.1-3: Upper limits**

We ran three experiments to establish how successfully agents can play the referring game using REAGENT's communication model when both agents have complete versions of Grammar 5.1. These experiments do not require any learning. Each of these experiments involved a Speaker and Listener playing 100 episodes of the referring game, each episode consisting of one randomly generated scene with a table and some number of objects on the table. In each episode the Speaker expresses a reference to each object on the table in turn, and the Listener tries to determine the referent of the expression. Because the Listener is not learning in these experiments, its performance is not compared to the PMI method, as there is no way to test the

upper limit of PMI's performance without access to the "true" probabilities of co-occurrence, which are not available.

**Experiment 1.1** The first experiment tested the noun constructions under ideal circumstances. Each scene contained one object of each shape, all of the same color and randomly distributed on the table. In this setup, it should have been possible for the Speaker to generate an unambiguous referring expression for each object every time. This experiment was run once for 100 episodes, with 3 utterances each episode for a total of 300 utterances.

As expected, averaged over 100 episodes, the Listener had 100% accuracy at selecting the correct referent in this experiment.

**Experiment 1.2** The second experiment, similarly, tested the adjective constructions under ideal circumstances. The setup was the same, except each scene contained one object of each color, all of the same shape and randomly distributed on the table. This experiment was run for 100 episodes with 5 utterances per episode for a total of 500 utterances.

Again, as expected, the Listener had 100% accuracy at selecting the correct referent.

**Experiment 1.3** The third experiment for Grammar 5.1 tested the effectiveness of the grammar in circumstances that may not be ideal. In this case, each scene contained five objects of randomly selected shape, color and location. Because the intrinsic properties are randomly selected, there is no guarantee that a uniquely applicable referring expression would be available for each object, as there may be more than one object with the same properties.

Under these circumstances, averaged over 100 episodes the Listener attained 94% accuracy at selecting the correct referent.

### Experiment 1.4: Learning Nouns

This experiment tested the ability of REAGENT to acquire noun constructions. The setup was the same as Experiment 1.1, with each episode containing one object of each shape. However in this experiment, all of the noun constructions were removed from the Listener's copy of Grammar 5.1, forcing it to learn replacement constructions. This experiment was run for 60 episodes, with 3 utterances per episode for a total of 180 training/testing utterances.

In 3 out of 3 runs, both REAGENT and the PMI baseline attained an accuracy of 100% on the final 100 utterances, which equals upper limit performance on this task. The results of training are shown in Figure 5.1 averaged over all 3 runs as well as a moving window of 50 utterances.

At the end of training, REAGENT's database of examples contained the syntactic signatures of three hypothetical constructions, one for each missing construction. The AF sets built from the examples associated with each construction exactly match up with those of the missing constructions in all 3 test runs, as would be expected with 100% accuracy:

HYP1 | N $\rightarrow$ "block" : $\{[\text{'block'} : 1.0]_{shape}\}$
HYP2 | N $\rightarrow$ "sphere" : $\{[\text{'sphere'} : 1.0]_{shape}\}$
HYP3 | N $\rightarrow$ "cylinder" : $\{[\text{'cylinder'} : 1.0]_{shape}\}$

### Experiment 1.5: Learning Adjectives

This learning experiment was set up identically to Experiment 1.3, except that all adjective constructions were removed from the Listeners grammar. As this setup does not guarantee unique descriptions for each object, the Listener is not expected to achieve 100% accuracy. This experiment was run for 200 episodes, with 5 utterances per episode (one for each object) for a total of 1000 training/testing utterances. This experiment was also run 3 times.

Average over the 3 runs, in the last 100 utterances of training REAGENT and PMI had attained accuracies of 85% and 87%, respectively, close to maximum per-

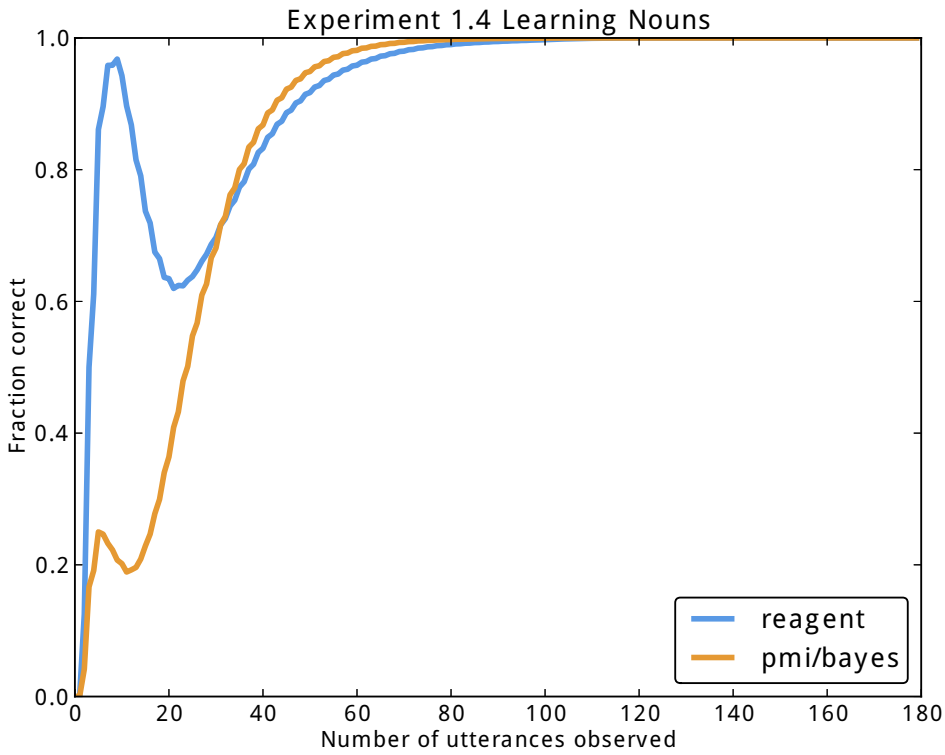



Figure 5.1: REAGENT and PMI accuracy over number of utterances encountered for Experiment 1.4, averaged over 3 runs, and with a running average over a window of 50 utterances

formance as seen in table 5.1. The full results of this experiment (again averaged over 3 runs and a running window of 100 episodes), are shown in Figure 5.2. As with experiment 1.4, REAGENT approaches the upper limit of performance after only 100 training utterances on average.

Again, after each run of the experiment the training database was used to build hypothetical constructions for each missing construction. In each of the 3 runs this resulted in exact matches for the missing constructions:

HYP1 | Adj $\rightarrow$ "red" : $\{[\text{'red'} : 1.0]_{color}\}$
HYP2 | Adj $\rightarrow$ "blue" : $\{[\text{'green'} : 1.0]_{color}\}$
HYP3 | Adj $\rightarrow$ "green" : $\{[\text{'blue'} : 1.0]_{color}\}$
HYP4 | Adj $\rightarrow$ "white" : $\{[\text{'white'} : 1.0]_{color}\}$

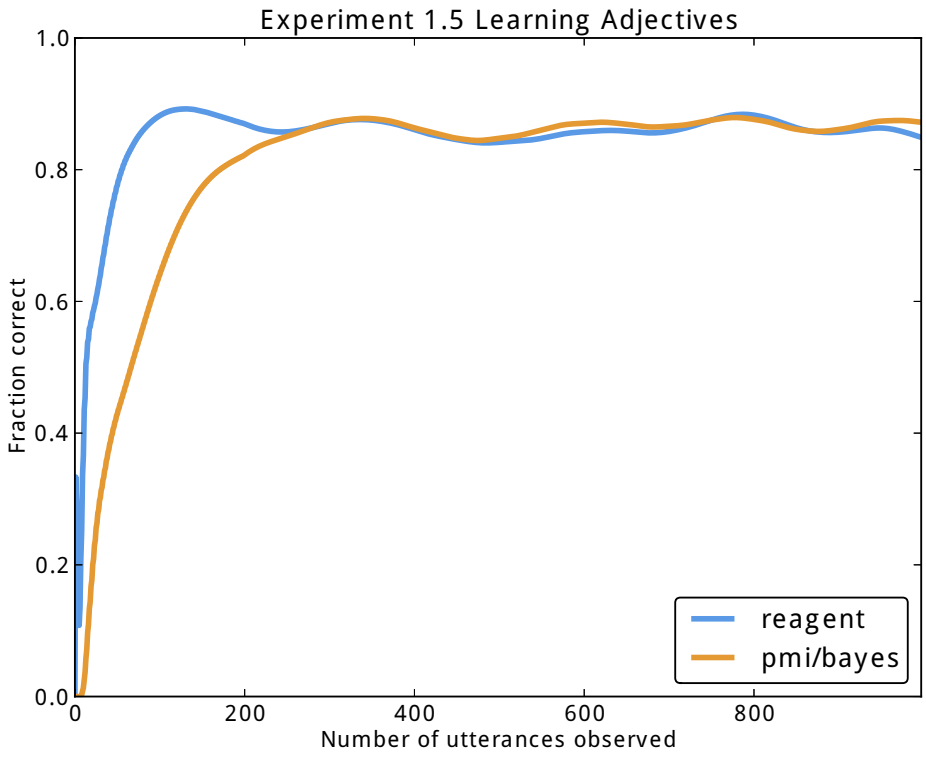



Figure 5.2: REAGENT and PMI accuracy over number of utterances encountered for Experiment 1.5, averaged over 3 runs, and with a running average over a window of 100 utterances

$$\text{HYP5} \mid \text{Adj} \rightarrow \text{“black”} : \{[\text{‘black’} : 1.0]_{color}\}$$

**Experiment 1.6: Learning Adjective and Nouns combinations**

This experiment was also set up exactly as experiments 1.3 and 1.5, except both the adjective and noun constructions were removed from the Listener's grammar. Therefore because REAGENT can include no more than one hypothetical construction for any given parse, for any utterance that contains both an adjective and noun that hypothetical construction must cover the meaning of both. For example, if the speaker says “the blue cylinder”, the Listener will attempt to learn a construction for “blue cylinder”, as neither word is known. This means that there will be many more hypothetical constructions in play as each combination of adjective and noun

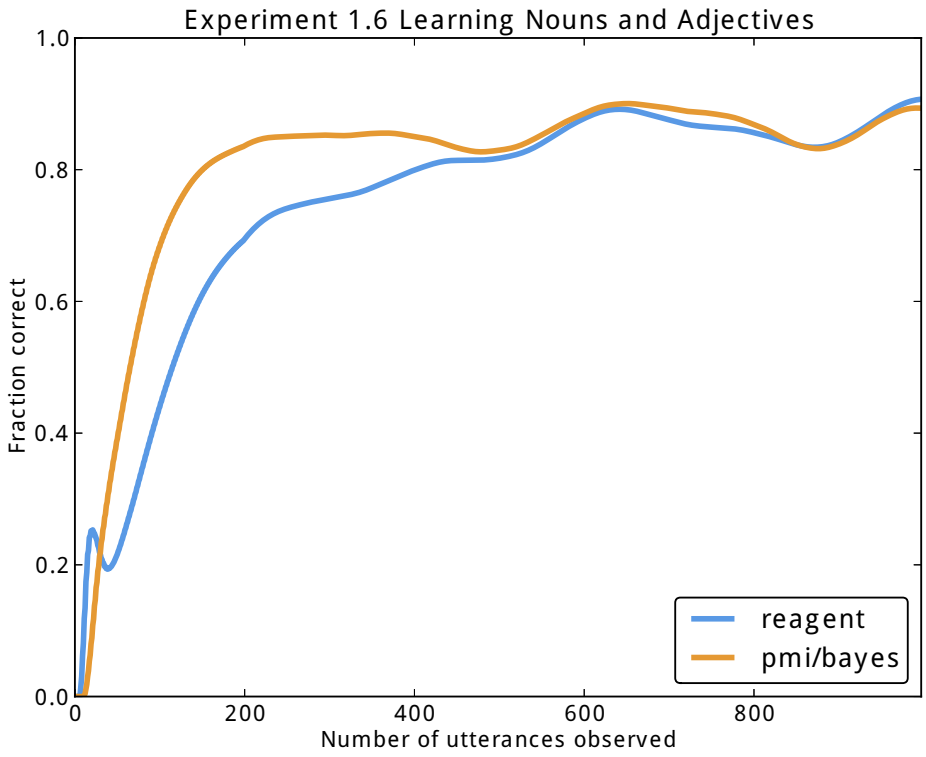



Figure 5.3: REAGENT and PMI accuracy over number of utterances encountered for Experiment 1.6, averaged over 3 runs, and with a running average over a window of 100 utterances

is learned separately, and thus there will be fewer examples of each. This also means that the Listener must learn semantics for these constructions that include two applicability functions. This experiment was run with 200 training episodes, with 5 utterances per episode for a total of 1000 training/testing utterances. As with the others, this experiment was run 3 times and the results were averaged over all runs.

As expected, REAGENT took longer to learn in this experiment. However after 300 utterances of training, REAGENT and PMI had achieved accuracies of 90% and 88% averaged over 3 runs, again very close to the upper performance limit at this task. The results of training are shown in Figure 5.3 averaged over the 3 runs.

After each round of training, the examination of the training database revealed a hypothetical construction for every combination of adjective and noun construction

| Exp | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 2.1 | 3.1 | 3.2 | 4.1 | 4.2 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Rgt. | 100% | 100% | 94% | 100% | 85% | 87% | 100% | 90% | 89% | 91% | 87% |
| PMI | | | | 100% | 87% | 88% | | | 63% | | 32% |

Table 5.1: Percent accuracy attained by both methods in each experiment. For learning experiments results for the PMI baseline method are also shown, and the number shown is the average over the last 100 utterances.

that were missing from the grammar, as well as solo constructions for each of the missing nouns for a total of 18 hypothetical constructions. In all 3 runs only 3 of the learned constructions' semantics did not match up precisely with the expected semantics, for an average of 17 out of 18 qualitatively correct constructions on average.

### 5.3.2 Experiment Set 2: Part-Of Expressions

This set consists of one experiment test the ability of Reagent's communication model to generate and interpret utterances about the table and its parts. Interpreting such expressions is crucial to being able to understand the landmark phrases of spatial referring expressions, as tested in Experiment Set 3. No learning experiments were conducted using this iteration of the grammar, and again no comparison to PMI was made.

#### Part-Of Grammar

Grammar 5.2 extends Grammar 5.1 to allow expressions such as "the table", "the left edge of the table", and "the front right corner of the table". This extension contains four noun constructions for describing the table and its parts, as well as four directional constructions which can be used as adjectives. These adjectives can make use of the TwoAP construction defined in Grammar 5.1 to use two directional adjectives when necessary, as in describing the corners of the table. This extension introduces the PartOf construction which expresses the relation that an entity is part of another entity. Lastly the ERefEx construction allows the creation of referring expressions that include a relation and a subordinate landmark referring

expression.

TABLE | N → "table" : $\{[\text{'table'} : 1.0]_{shape}\}$
EDGE | N → "edge" : $\{[\text{'edge'} : 1.0]_{rep}\}$
CORNER | N → "corner" : $\{[\text{'corner'} : 1.0]_{rep}\}$
MIDDLE | N → "middle" : $\{[\text{'middle'} : 1.0]_{rep}\}$

FRONT | Dir → "front" : $\{\text{Cen}_{angle}(0\,^\circ, 30\,^\circ)\}$
BACK | Dir → "back" : $\{\text{Cen}_{angle}(180\,^\circ, 30\,^\circ)\}$
LEFT | Dir → "left" : $\{\text{Cen}_{angle}(-90\,^\circ, 30\,^\circ)\}$
RIGHT | Dir → "right" : $\{\text{Cen}_{angle}(90\,^\circ, 30\,^\circ)\}$

PARTOF | Rel → "of" :: $[\{\text{True} : 1.0\}_{part\text{-}of}\}]$

DIRADJ | Adj → Dir :: $\lambda d.d$

EREFEX|RefEx → Det NP Rel RefEx : $\lambda d.\lambda n.\lambda r.\lambda re.(d \cup n \cup r)_{re}$

Grammar 5.2: Extends Grammar 5.1

**Experiment 2.1: Performance Limit**

Experiment 2.1 tested the accuracy of referring to the table and its parts. In this experiment each scene contained only a table which has nine parts: four edges, four corners and a middle. This experiment contained 100 episodes, during each of which the Speaker described referred to each entity in the scene in turn, and the Listener tried to recover the correct referent. This experiment was run for 100 episodes with 9 utterances per episode for a total of 900 utterances.

In this experiment, the Listener correctly interpreted 100% of the Speaker's referring expressions averaged over 100 episodes. This shows that confusion over table-based landmark referring expressions should not contribute to any error in interpretation of the referent of spatial referring expressions in experiments using

the next two grammars.

### 5.3.3 Experiment Set 3: Spatial Referring Expressions

The penultimate set of experiments tested REAGENT's methods for using and learning spatial relations in referring expressions. Unlike most of the lexical constructions in the previous grammars, these spatial relations have continuous, real-valued applicabilities which makes their applicability functions more of a challenge to acquire. The addition of these relations allows the production of expressions such as "the object near to the left edge of the table" or "the object to the right of the green block". These expressions have significant syntactic complexity and should prove a challenge to the hypothetical parsing system.

#### Spatial Refex Grammar

Grammar 5.3 extends Grammar 5.2, adding constructions for spatial relations. These relations allow referring to an entity by its relative location to another entity. NEARTO and FARFROM are applicable (roughly) when the distance between the referent and landmark is less than 0.15 (arbitrary scene) units, or greater than 0.55 units respectively. Similarly, ON is applicable when at least 20% of the referent overlaps with the landmark. This is useful for relating a referent to the parts of the table, where each part of the table covers some region of its area. The ANGLEREL construction can compose with the direction constructions defined in Grammar 5.2 to express that the angle formed between the viewer, the landmark and the referent is approximately $0\,^\circ$, $90\,^\circ$, $180\,^\circ$, and $-90\,^\circ$ respectively for FRONT, RIGHT, BACK, and LEFT respectively.

NearTo | Rel → "near to" : $\{\text{Sig}_{distance}(0.15, -0.1)\}$
FarFrom | Rel → "far from" : $\{\text{Sig}_{distance}(0.55, 0.1)\}$

On | Rel → "on" : $\{\ \text{Sig}_{overlap}(0.5, 0.1)\}$

AngleRel | Rel → "to the" Dir "of" : $\lambda d : \text{Sem}(d)$

Grammar 5.3: Extends Grammar 5.2

**Experiment 3.1: Performance Limit**

As with the first two sets of experiments, this performance limit experiment tested how well Grammar 5.3 works with Reagent's communication model to accomplish reference. This experiment involved 100 episodes of the referring game, where each episode was set up as in Experiment 1.6. Each scene contained one table (with its nine parts), plus five objects of randomly selected shape, color and location. For each scene the Speaker described each object on the table in turn, however all expressions were required to start with "the object..." in order to force the Speaker to use spatial relations. This experiment was run for 100 episodes with 5 utterances per episode for a total of 500 utterances. Again, no comparison to PMI is made here.

Averaged over 500 utterances, the Listener recovered the correct referent 90% of the time in this experiment.

**Experiment 3.2: Learning Spatial Relations**

This experiment tested Reagent's methods at learning the spatial relations in Grammar 5.3 by removing all of the Rel constructions (as well as the Dir constructions) from the Listener's copy of the grammar. Otherwise, this experiment was set up exactly as in Experiment 1.3. This experiment was run for 200 episodes of training with 5 utterances for a total of 1000 training/testing utterances.

The results of this experiment, averaged over 3 runs of the experiment and with

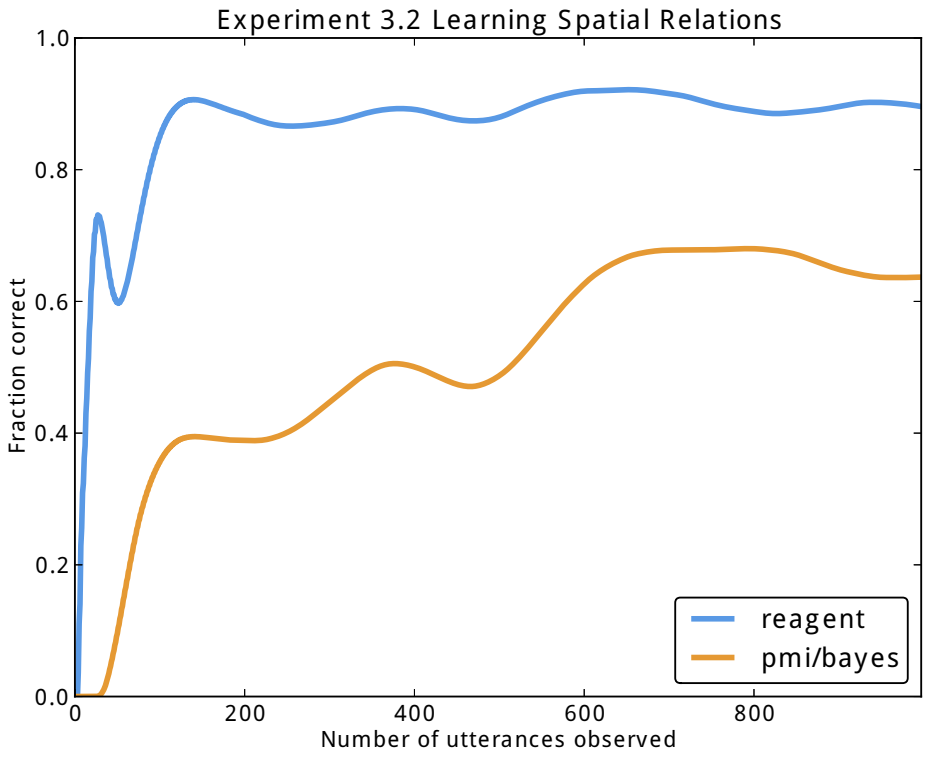



Figure 5.4: REAGENT and PMI accuracy over number of utterances encountered for Experiment 3.2, averaged over 3 runs, and with a running average over a window of 100 utterances

a running average over 100 utterances, are shown in Figure 5.4. On average, after about 200 utterances of training the Listener's performance was slightly below the performance limit, with an average over the last 100 utterances of 89% accuracy. PMI attained 63% accuracy, although it took much longer because it had to learn the correlations for all words rather than only spatial relations. These results for PMI validate the usage of kernel density estimation for continuous features, as such accuracy could not be attained without finding correlations between the spatial relation terms and the continuous features.

Figure 5.5 compares the Speaker's original semantics for the orientation relations to the representations learned by the Listener by the end of one training run. In every training run the Listener learned the correct general form (Centroidal applica-

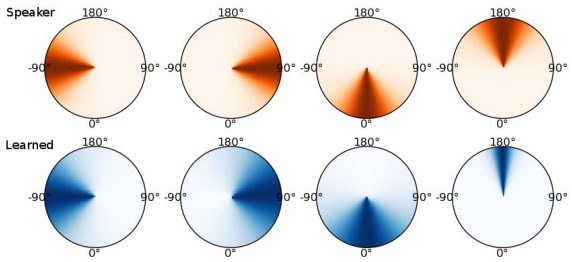



Figure 5.5: The applicability functions used by the Speaker (top) and learned by the Listener in Experiment 3.2, for the phrases "to the left of", "to the right of", "to the front of" and "to the back of" (from left to right)

bility function over the angle feature) for these semantics, and learned approximately correct values for the *loc* parameter, but did not accurately learn the *shape* parameters. It is likely that this is simply due to insufficient training, and that even longer training runs would produce better results. The other spatial relations were learned with similar accuracy. The two distance relations were always learned as functions of the distance feature, however in 2 out of 3 of the runs at least one of these relations was learned as Centroidal applicability function rather than Sigmoidal. This is likely due to the limited range of distances allowed by the table scenes. However these Centroidal semantics prove to be very close in performance to their Sigmoidal counterparts. The "On" relation was consistently learned as a Sigmoidal function of overlap across all runs.

### 5.3.4 Experiment Set 4: Spatial Referring Expressions 2

The final set of experiments test REAGENT's ability to learn AF sets containing multiple continuous applicability functions. This task

#### Spatial Refex Grammar 2

This second grammar for spatial referring expressions overrides the direction constructions introduced in Grammar 5.2 to add an additional constraint to each of their semantics. This constraint specifies that "front", "back" and the other direc-

tion words not only require entities oriented in a certain way given the viewer and landmark, but also that the referent must be not be very distant from the landmark. This not only provides more complex semantics for REAGENT to try to learn, but also accords better with human intuition. For instance if two objects are both aligned in the front direction from a given landmark, the one closer to the landmark would be considered the one in front of it.

FRONT | Dir $\rightarrow$ "front" : $\{\text{Cen}_{angle}(0\,^\circ, 30\,^\circ), \text{Sig}_{distance}(0.15, -0.1)\}$
BACK | Dir $\rightarrow$ "back" : $\{\text{Cen}_{angle}(180\,^\circ, 30\,^\circ), \text{Sig}_{distance}(0.15, -0.1)\}$
LEFT | Dir $\rightarrow$ "left" : $\{\text{Cen}_{angle}(-90\,^\circ, 30\,^\circ), \text{Sig}_{distance}(0.15, -0.1)\}$
RIGHT | Dir $\rightarrow$ "right" : $\{\text{Cen}_{angle}(90\,^\circ, 30\,^\circ), \text{Sig}_{distance}(0.15, -0.1)\}$

Grammar 5.4: Extends Grammar 5.3

**Experiment 4.1: Performance Limit**

This experiment proceeded exactly as experiment 3.1 but using the new grammar. The experiment was run for 100 episodes with 5 utterances per episode for a total of 500 training/testing utterances.

Averaged over all training episodes, the Listener correctly inferred the referent 91% of the time. This slight improvement is attributed to the increased specificity of the new orientation relations, which allow them to be used in situations where the previous versions could not.

**Experiment 4.2: Learning 2 AF Spatial Relations**

The final learning experiment had the same setup as experiment 3.2, but with the Speaker using the new direction words in its orientation relations. As before, this experiment was run for 200 episodes with 5 utterances per episode for a total of 1000 training/testing utterances.

The results of this experiment, averaged over 3 runs of the experiment and with a running average over utterances are shown in Figure 5.6. Averaged over the

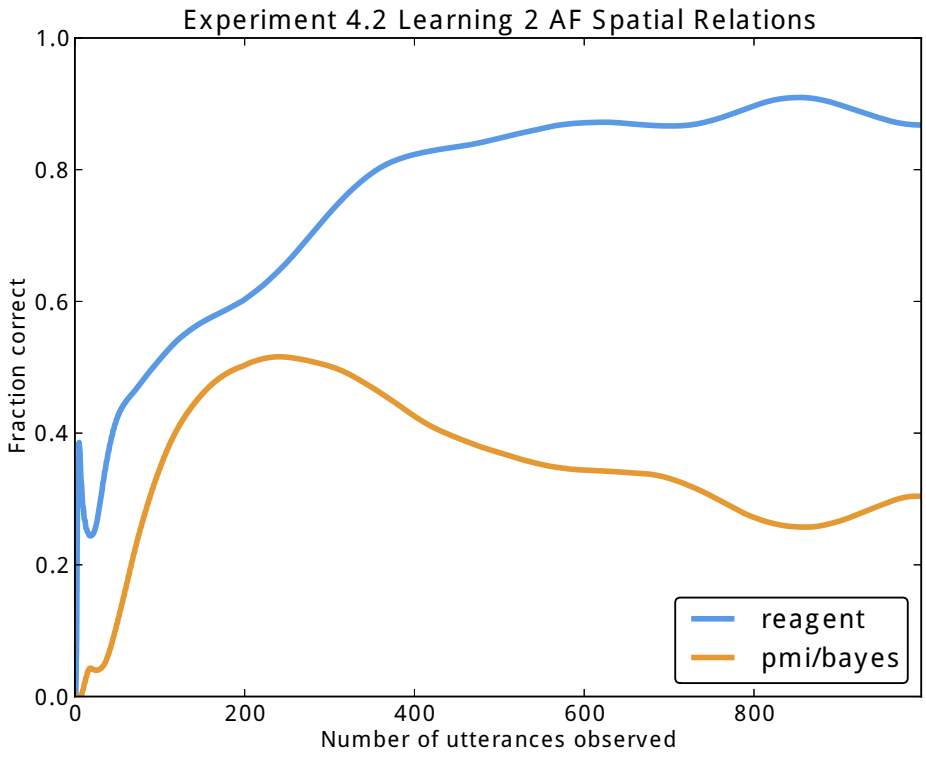



Figure 5.6: REAGENT and PMI accuracy over number of utterances encountered for Experiment 4.2, averaged over 3 runs, and with a running average over a window of 100 utterances

final 100 training utterances, REAGENT and PMI achieved accuracies of 87% and 31% respectively. An analysis of the Speaker's utterances showed that orientation relations were used much more frequently in this experiment, while in Experiment 3.2 distance relations were primarily used. The drop in PMI's performance is likely due to this difference, as the new spatial relations would provide a greater challenge.

Qualitatively, the constructions learned for the orientation relations were very similar to those learned in experiment 3.2, except that in each case a second applicability function using the distance feature was added to the representation. These distance applicability functions were not consistently fit with Sigmoidal functions, but like the distance relations learned in experiment 3.2, the were often fit with Centroidal functions instead.

## 5.4 Experiments: Series 2

The first series of experiments showed that when the Speaker and Listener are both using the same semantic model in the Referring Game, the Listener can learn to supplement an incomplete grammar from observing the Speakers utterances. In doing this, the Listener learns to choose the proper referent with accuracy approaching that of an agent with a complete grammar, and learns meanings for the missing constructions that are qualitatively similar to those used by the Speaker. But while the grammars and semantic model used in these experiments was based on linguistic research, it is unknown to what degree it truly corresponds to the semantics of English users.

In this second series of experiments, rather than utterances being generated by an instance of REAGENT they have been gathered from human speakers of English via Amazon's Mechanical Turk service, as part of the precursor to this work presented in (Dawson et al., 2013). Turk users (Turkers) were shown a three-dimensional graphical rendering of a tabletop scene such as the one in Figure 5.7. These scenes contained objects of up to three different types of shapes and seven different colors. The Turker was then prompted to describe one of the numbered objects in two ways: by the object's intrinsic qualities, as in "the orange cylinder", and by the object's relative location in the scene, as in "next to the red ball". Five scenes with five objects each were used, and a total of 5170 object descriptions and 5305 location descriptions were gathered. Out of the 5305 location descriptions, 350 were randomly selected for a second Mechanical Turk study. In this study each Turker was shown one of the 350 object location descriptions along with the image of the scene containing the object described. The Turker was then prompted to choose which object in the image they believed the description to refer to, also known as a 'binding'. A total of 2875 such bindings were gathered, with at least 5 bindings per description.

The data collected from Turk provided utterances for two types of experiments. As in Experiment Set 1 from the first series, the Turker's intrinsic descriptions of

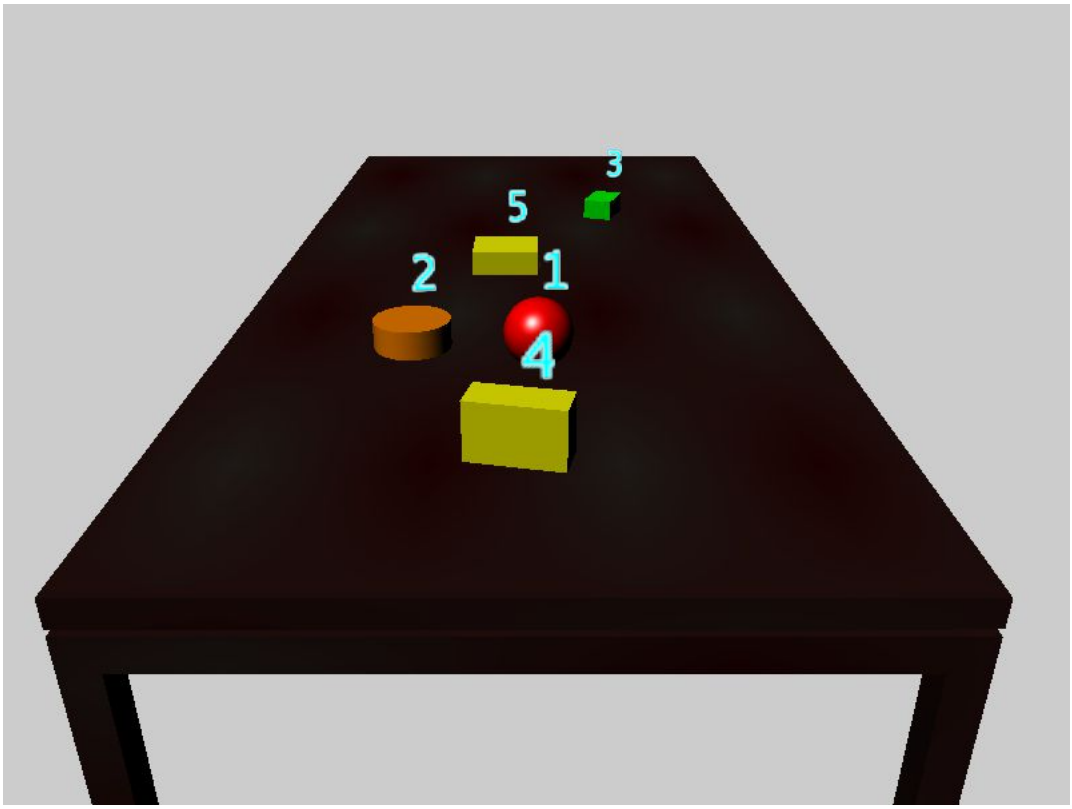



Figure 5.7: A scene shown to users of Amazon's Mechanical Turk

objects could be used to test REAGENT's ability to learn adjectives and nouns. And as in Experiment Set 2, the location descriptions could be used to test REAGENT's ability to learn spatial relation terms, with the added benefit that the Turk-gathered bindings could serve as a test set for comparing REAGENT's interpretive accuracy to that of human speakers.

The descriptions gathered from Turk contain a much larger variety of language than is covered in the grammars presented in the first series of experiments. However, the goal of this series is not to demonstrate wide coverage of English syntax and semantics, but rather to discover whether and to what degree the syntactic and semantics possibilities covered by REAGENT's model of language correspond to how humans use referring expressions, and whether REAGENT's learning methods can capture this correspondence. For that reason the Turk data was analyzed to extract those descriptions that are of a form that REAGENT can parse using Grammar 5.2 with minimal extension of its vocabulary, or which could be coerced into such a form with few changes.

### 5.4.1 Data Preparation

Prior to any changes, only 1% of both the object and location descriptions could be parsed under Grammar 5.2. In a manual survey of these descriptions there were two very recognizable reasons why most descriptions wouldn't parse: out-of-vocabulary words, and syntactic issues such as punctuation, conjunctions, and articles. After counting all of the words in both sets of descriptions, it was found that many of the most commonly used words such as color and shape adjectives, shape nouns, articles, and spatial relations could easily be added to Grammar 5.2. Other categories of words including dimensional adjectives (e.g. "small" or "wide"), adverbs (e.g. "slightly"), and verbs(e.g. "located"), and plurals could not be so easily added to the grammar. Instead, such words (except for plurals) were removed from the descriptions in order to make them parseable, although this runs the risk that removal also makes them uninterpretable. Likewise punctuation at the end of descriptions were removed. Punctuation such as commas and periods in the middle

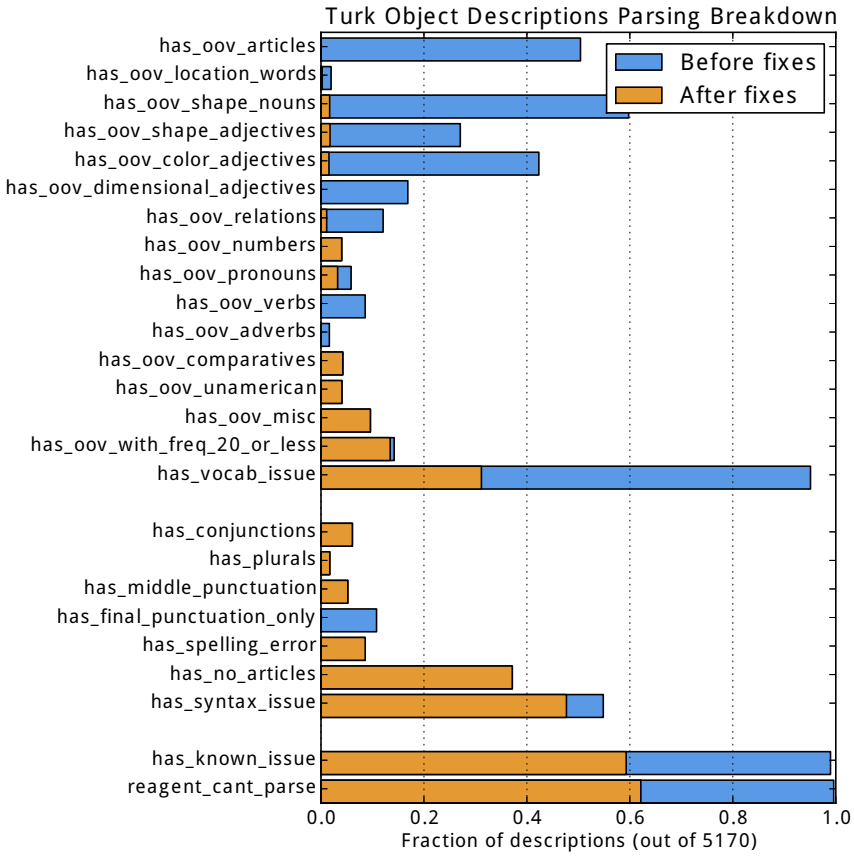



Figure 5.8: Fraction of Turk object descriptions with various barriers to parsing, before and after changes.

of a description usually indicated multiple phrases or sentences. In these cases, as well as in descriptions containing conjunctions the description was split into multiple phrases, and those phrases were parsed individually. These measures result in 38% (1958) of the object descriptions being parseable under the now extended Grammar 5.2, as shown in Figure 5.8. Of the location descriptions 22% (1112) of the 4955 training descriptions and 40% (140) of the 350 test descriptions were parseable after changes, as seen in Figure 5.9 (with the training and test set combined).

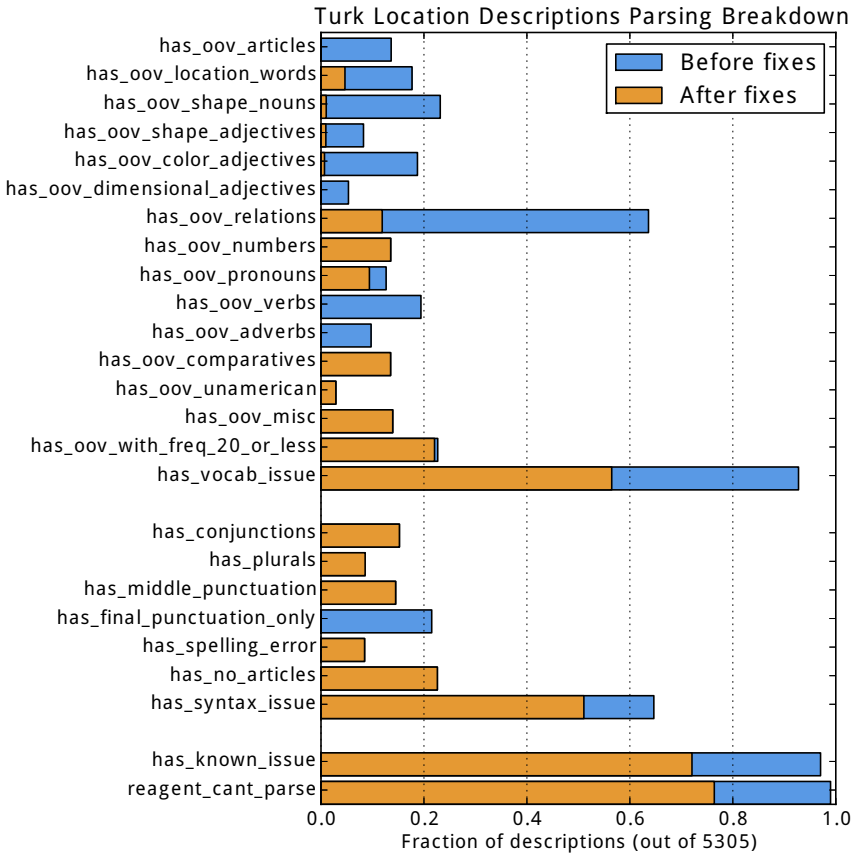



Figure 5.9: Fraction of Turk location descriptions with various barriers to parsing, before and after changes.

### 5.4.2 Object Description Experiments

The training set of 1958 parseable object descriptions were used for two Referring Game experiments. As in the first series of experiments REAGENT was instantiated as the Listener. In the first experiment the Listener was given a version of the extended Grammar 5.2 in which all of the nouns had been removed and then played the game using the Turk descriptions. The results of this experiment, averaged over 3 runs in which the training order was shuffled, can be seen in Figure 5.10. Here PMI slightly outperforms REAGENT. The same experiment was then run again with the

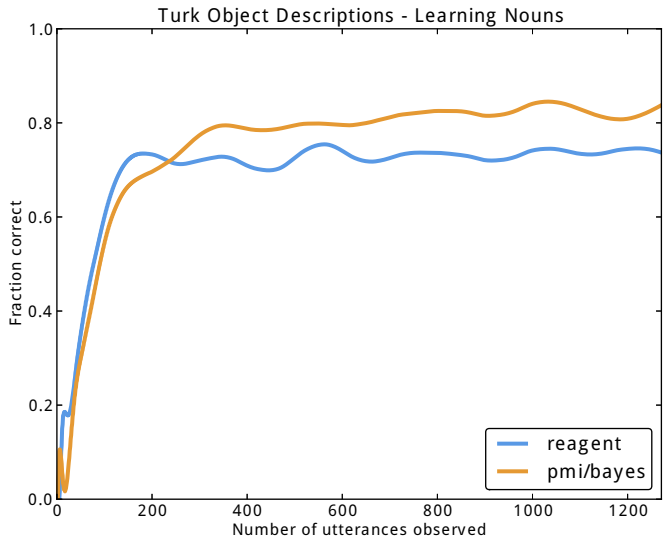



Figure 5.10: Results of the Referring Game on the Turk object descriptions with the Listener's nouns removed. Averaged over 3 runs in which the training order was shuffled, and with a running average over a 100-utterance window applied for smoothing.

Listener's adjectives removed rather than nouns, the results of which are in Figure 5.11. Here the two performed almost equally.

### 5.4.3 Error Analysis

A manual error analysis was done to determine what aspects of the data could be the cause of REAGENT's slightly depressed performance in these experiments. From the 1958 parseable object descriptions 100 were randomly selected for analysis, in which several potential issues were noticed.

Ten of these descriptions were underspecified, for example saying only "the box" when there are multiple boxes in the scene. Fifty-six descriptions were overspecified, as in "the yellow cube" when there is only one yellow object in the scene. Both of these issues may be due to the instructions given in the Mechanical Turk task, as users were not instructed to uniquely describe the object in a way that did not under-

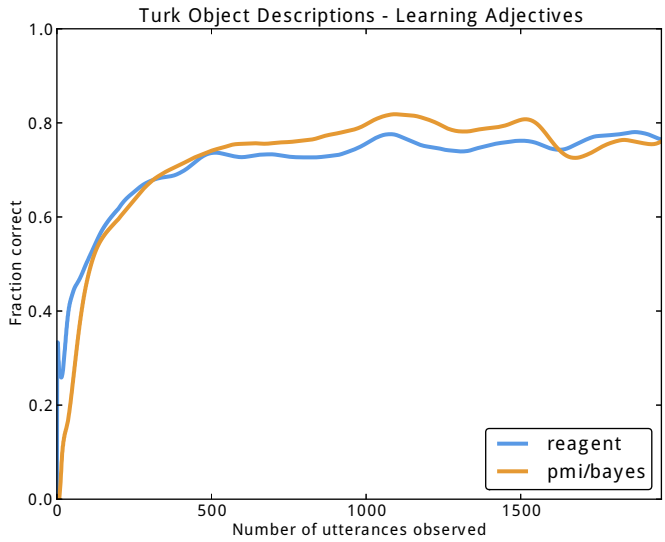



Figure 5.11: Results of the Referring Game on the Turk object descriptions with the Listener's adjectives removed. Averaged over 3 runs in which the training order was shuffled, and with a running average over a 100-utterance window applied for smoothing.

or over-specify. REAGENT assumes that descriptions are uniquely specified, and when this assumption is violated it affects which observations are used as (positive or negative) examples, which would have an adverse effect on REAGENT's performance. Because there are more colors than shapes in this dataset and nouns are only used to distinguish shape and not color, overspecification affected REAGENT's ability to learn nouns more. For instance, in the example of "the yellow cube", it is more likely that there are multiple cubes in the scene than multiple yellow objects, thus if REAGENT is trying to learn the word 'cube' but the known adjective 'yellow' already reduces the potential referents down to one object, then REAGENT will think there are no negative examples of 'cube' in the scene and thus will learn poorly. The reverse situation in learning 'yellow' is less likely to occur.

Twenty-three descriptions had originally contained dimensional adjectives which may have been necessary to determine the referent, as in "the small cube". Even if

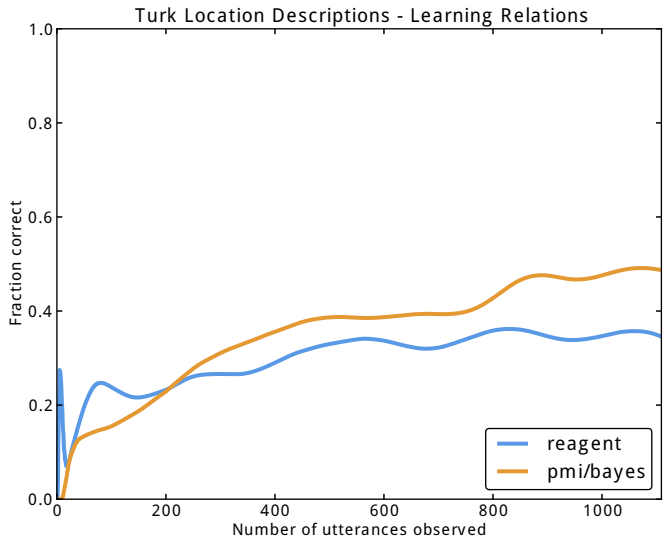



Figure 5.12: Results of the Referring Game on the Turk location descriptions with the Listener's spatial relation terms removed. Averaged over 3 runs in which the training order was shuffled, and with a running average over a 100-utterance window applied for smoothing.

these adjectives had been left in, the semantic model does not include any dimensional features with which to differentiate such objects. Nevertheless, this may have reduced the performance of both REAGENT and PMI at this task.

Twenty-two descriptions were necessarily ambiguous, that is they described an object that was indistinguishable from another object in the scene based on intrinsic qualities alone. This also surely reduced both method's performance at this task.

### 5.4.4 Location Description Experiments

In the final experiment REAGENT plays the Referring Game using the 1112 location descriptions in the training set gathered from Turk. Again REAGENT plays the Listener, and this time it is given a copy of the extended Grammar 5.2 with all spatial relation terms removed. The results of this experiment, again averaged over 3 runs with different shuffling of the training data, are shown in Figure 5.12.

These results show PMI significantly outperforming REAGENT at this task, although both methods do relatively poorly, not even achieving 50% accuracy. However it is likely that PMI is exploiting a lack of variety in the training data to achieve these results. Because there are only 5 scenes with 5 objects each, a given object is likely to be described similarly many times. Even if these descriptions do not actually make reference to the intrinsic properties of this object, there would nevertheless be a pattern in the data in which the words of the description are more likely to occur describing an object with those specific intrinsic properties. For example if a green cube is commonly described as "the object near the far edge of the table", these words will be more likely to co-occur when the referent's shape and color features are 'BOX' and 'GREEN'. PMI could easily take advantage of this pattern, while REAGENT would be less able to take advantage, as it would only learn an association to the relation term of the description rather than all terms. This can be seen by examining the semantics REAGENT learns for each spatial relation, and indeed after training REAGENT has learned all but 3 spatial relation terms as functions of the shape or color features.

In order to negate this effect, a second experiment was run with this data in which for each training utterance the referent is temporarily disguised by giving it alternate, randomly chosen shape and color tags. This means that even though a particular object may be described the same way frequently, there should be no correlation between that description and its observed intrinsic properties. This should have no effect on true understanding of the description as it does not reference shape or color. Additionally the original shape and color tags were always restored to an object at the end of each training instance. The results of this experiment, again averaged over 3 shuffled runs, are shown in Figure 5.13. In this version of the experiment REAGENT's performance is slightly improved, while PMI's performance is drastically reduced.

Following training both PMI and REAGENT played one final round of the Referring Game using the 140 location descriptions in the test set which Turkers had also interpreted. On this test set, the human Turkers were 65.1% accurate at interpreting

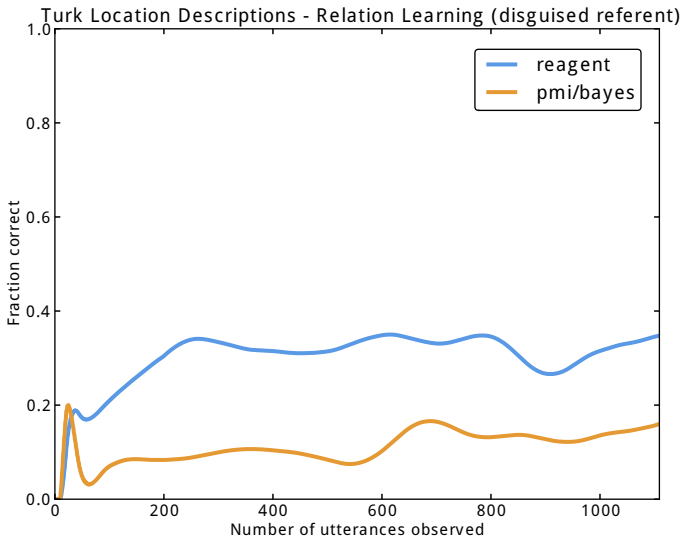



Figure 5.13: Results of the experiment shown in Figure 5.12, but with the referent "disguised" during each training example to avoid false patterns.

the correct referent, while REAGENT had 31.4% accuracy, and PMI 12.9%.

### Error Analysis

Both REAGENT and PMI performed significantly worse than humans at interpreting these spatial referring expressions even after exposure to many training examples. An error analysis of the training data was done in an attempt to determine what factors may have contributed to this poor performance. Of the 1112 parseable location descriptions in the training set, 100 were randomly selected for analysis. Of those 100, 9 contained multiple undelimited relation phrases, as in "near the edge behind the cube" in which case REAGENT would try to learn a meaning for a relation "near the edge behind" due to the stricture of only allowing one hypothetical construction per parse. Eleven descriptions inaccurately described the location of the referent, for instance by saying "behind" some landmark when the referent was actually in front of it. Twenty-one descriptions did not uniquely specify the referent, for instance saying "near the edge" when two objects were equally near the

edge. Fifteen descriptions used a landmark that was not included in REAGENT's model of the scene, such as "the middle of the left edge of the table" (many such landmarks-of-landmarks were excluded from the model in order to decrease the search space). And 26 descriptions had been split into multiple phrases on a comma, period, or conjunction yielding at least one parseable phrase, but required one of the unparseable phrases to be correctly interpreted. In all, 74 of these descriptions had one of the issues described above, with approximately have of these being syntactic or semantic errors on the part of the Turkers, and the other half being due to shortcomings in REAGENT's grammar, parsing, or semantic model. Assuming this sample is representative of the full training set, these issues would explain much of REAGENT's poor performance at this task.

One final issue with the Turk location description training set is the small number of scenes used. Because there were only 5 scenes with 5 objects each there could only be a few examples of each spatial relation. This small set of observation vectors, though each repeated many times, would make it very difficult to learn robust and accurate meanings for the missing constructions.

### Learned Constructions

Despite the issues with the training data described in the previous paragraph, REAGENT still managed to learn appropriate-seeming semantics for several spatial relations. 'In', 'on', and 'at' were all learned as discrete functions of containment. 'Near' was learned as a sigmoid function of distance. 'To the left of' and 'behind' were both learned as centroid functions of angle. However the remaining 7 relation terms were erroneously learned as functions of color or shape, and most of them were used in fewer than 50 descriptions in all.

# CHAPTER 6

# DISCUSSION

The goal in designing REAGENT was to create a system that could learn the syntax and semantics of complex referring expressions by observing only un-annotated, situated utterances and the entities to which they referred. The first series of experiments in the previous chapter showed that REAGENT can take a referring expression as complex as "the object near to the front left corner of the table" in which the meaning of the relation ("near to") is unknown, hypothesize appropriate syntax for a new construction, and determine which entities in the context do or do not exemplify that construction based on context. From roughly one hundred examples of an unknown construction being used, REAGENT can learn a syntax and meaning for it that not only functions as accurately as the speaker's construction, but is very close in semantic form as well. This was demonstrated for seven different relations of various lengths (in words) with meanings as complex as the product of two continuous-valued applicability functions, as well as for much simpler adjective and noun constructions.

This first series of experiments was a proof-of-concept of learning under ideal circumstances: from another software agent using the same scene representation, model of communication, semantic model (including semantic primitives), and the same grammar with the exception of the missing constructions. In the second series of experiments the conditions were far less ideal. The training observations lacked variety, many training instances were likely malformed or inaccurate, and many were misinterpreted due to REAGENT's limited grammar and semantic model. Nevertheless REAGENT learned appropriate-seeming meanings for half of the missing spatial relations, and significantly outperformed the baseline comparison method in accuracy. This small success suggests that not only is REAGENT capable of learning semantics from human utterances, but also that the features and general applicabil-

ity functions used in these experiments capture at least some of the functionality of human speaker's semantic models for English spatial referring expressions. Perhaps REAGENT only did not match human performance because the proper semantic primitives were missing from its repertoire.

But what are the semantic primitives underlying human languages? Linguistic researchers have been proposing models for human syntax and semantics for decades as seen in chapter 2, but so far the only criteria for evaluating the semantic hypotheses have been subjective analysis or the laborious process of finding definitive counter-examples. (Artzi and Zettlemoyer, 2013), (Liang et al., 2013) and (Dawson et al., 2013) present one way to evaluate semantic hypotheses: assume a set of semantics, learn a grammar mapping to these semantics, and then evaluate whether the learned grammar generates the proper behavior when interacting with human input. However this process would need to be performed for every semantic hypothesis, and it would be up to human analysis to decide what might be wrong with a given hypothesis if it performs poorly in the test.

However the experiments with REAGENT suggest a different approach. Rather than define a series of potential semantic systems, one could simply define the semantic primitives from which each of these semantic systems could be built. Given human training data, REAGENT could then sort through all the primitives and build only those semantics that best predict the data. This approach would not only produce a system that could use and understand the language it had learned, but would also provide valuable information about what semantics are likely to underlie that language for humans.

## 6.1 Future Work

REAGENT's current form is too limited for discovering most of the semantics of referring expressions, even the limited subset of spatial referring expressions. Much could be accomplished merely by expanding the scene model and feature set, and crafting grammars to make use of these. However, some limitations are built into

REAGENT itself.

### 6.1.1 Superficial limitations

The scene representation used in chapter 5 is of very limited form. Objects are represented only as axis-oriented bounding boxes, which makes for fast but necessarily inaccurate calculations of distances, angles and overlap. A more advanced scene representation which uses non-axis-oriented boxes, as well as circles and convex polygons to represent objects is almost ready for use with REAGENT. Such a representation will allow for more accurate measurements, as well as better support for objects with referable parts, such as sides. However, as this representation is still two-dimensional it precludes reasoning and learning about several important spatial prepositions such as "over"/"above" and "under"/"below", as well as accurate handling of "on".

The feature set presented in chapter 5 is fairly limited, and causes some inaccuracies among the supported relations. Angle is currently measured from the centerpoint of a landmark, but if a landmark were sufficiently wide then even referents clearly in front of it might have vastly different angles. This suggests that angles should be measured using the faces of objects rather than their centerpoints. The current semantics of "near to" and "far from" use distance as a feature, which is measured in absolute units. However, in natural usage these terms can be applied at vastly different scales, as in "the mouse is near to the cheese" or "Chicago is near to New York". This suggests that these terms should be based on not just distance, but also some sort of scaling feature.

The limited feature set also doesn't support the implementation of some spatial relations. In order to implement "between", it would need to be possible to compare a referent's angle or distance to two different landmarks. Similarly relations like "nearest" and "leftmost" or "on the left" would require features about a referent's ordering in some given direction among a group of referents.

Additional compositional constructions would need to be introduced to the grammar in order to support words like "between" and "among" as well. Both of these

words require multiple landmarks, requiring the addition of plurals or conjunctions to the grammar. Conjunctions of a different form would be useful for specifying that one referent is involved in multiple relationships, such as "behind the cube and to the left of the sphere".

If all or most of these additions were made to the scene model, features and grammar, REAGENT would be much closer to being able to learn from human input, and potentially make discoveries about how humans represent spatial meaning. Such additions are planned for the future.

### 6.1.2 Deeper limitations

Some issues with REAGENT are less straightforward to solve. Currently if an utterance could be parsed in multiple ways REAGENT does not have any basis for choosing between them, thus the grammars in chapter 5 were carefully written so as to produce only one parse per utterance. For this reason REAGENT does not add hypothetical parses to its grammar, but rather proposes them anew each time they are needed, and learns a new meaning for them as well. For most unparseable utterances, there are many hypothetical constructions that could make them parseable. If REAGENT added more than one such construction to its grammar it would be adding more than one way to parse certain utterances, leading to indecision.

One common solution to this issue in linguistics is to have some way of weighting parses. There are many strategies for such weights, one of the most popular is maintaining counts of how frequently each construction is seen, and then parses can be judged by how common their individual constructions are. This solution could be adapted to REAGENT's grammar framework.

Similarly, REAGENT's current treatment of learning is that the semantics of constructions in the grammar are sacrosanct, while hypothetical construction's semantics are subject to revision at every encounter. However if constructions were to be added to the grammar, then REAGENT would need some method of choosing if and when to update a construction's meaning. Such an update rule could be based on how accurately a construction predicts referents.

Another limitation of REAGENT's learning method is that it can only learn lexical meanings. In the grammars shown in chapter 5 compositional constructions only perform two operations: union of two or more AF sets, or subordinating one AF set to another in the case of a construction that has a landmark Refex as one of its constituents. Whenever a compositional construction is composed it would be fairly straightforward to support compositional learning by trying every possibility of these two operations between its constituents, as most constructions have very few constituents. However in some cases it is unknown ahead of time that a compositional construction should be learned. For instance, in Experiment 1.6, because the Learner does not have any nouns or adjectives in its grammar, it learns meanings for each adjective-noun combination as a single lexical noun construction, such as Noun → "red cube". In this case, REAGENT should be able to recognize that this lexical construction should be broken up into two separate lexical constructions, and potentially one compositional construction as well. Such a "generalization" procedure has already been explored somewhat, involving comparing all lexical constructions for similarities in their pattern, and when matches are found looking for similarities in their sempoles as well. When performed on all of the constructions learned in Experiment 1.6, this procedure finds that, for instance, all constructions with the string "red" in their pattern also contain a discrete applicability function of the form {'red':1.0}. With some extension, this procedure could be used to separate learned constructions that conflate two of the speaker's original constructions. Future extensions of REAGENT are planned to support both types of compositional learning discussed here.

One final limitation is the running time and space required for both hypothetical parsing and constructing semantics. The time and space needed for hypothetical parsing increases exponentially with both the size of the constructicon and the length of the utterance being parsed. This time and space complexity greatly constrained the variety of language that was addressed in the experiments of the last chapter. This issue would partially be addressed by adding new constructions to the lexicon as discussed above, in which case hypothetical parsing would be used less frequently.

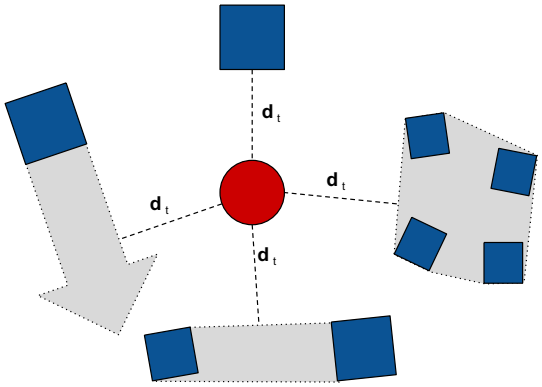
Similarly the running time required for semantic induction increases linearly with the product of the cardinalities of the feature set and applicability function set. Without an improvement in running time, adding many new features and functions in an attempt to capture the semantics of human utterances would slow experiments down beyond feasibility.

### 6.1.3 Advancements

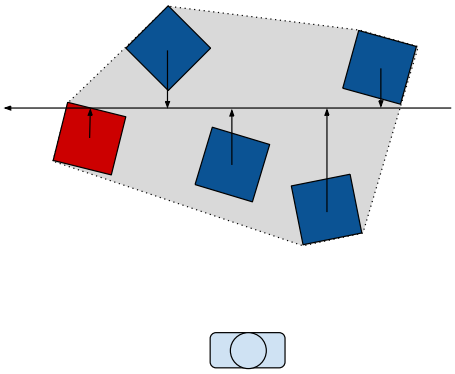
The REAGENT view of communication relies on the idea that agents maintain models of worlds in their mind. Under this view, communication is a way for two or more agents to compare and contrast their own world models and thus share information. This is accomplished by transmitting descriptions of relevant portions of a model along with instructions on how these partial models should be used by the receiver. So communication can be viewed as conveying model operations between agents. For instance, an assertion operation provides a partial model along with instructions that it should be incorporated into the receiving agents world model. In this view of language, reference is the most basic model operation, as it allows communicating agents to align their models by known or assumed homologous portions. Only after this can any modifications or discussion of differences be accurately performed. While there are is still much work to be done on reference, expanding into other model operations is an eventual goal of this research. Of particularly practical use in learning the meaning of words would be interrogative operations, as they would allow a learning agent to clarify ambiguous observations.

In the REAGENT framework presented here, agents only communicated operations on the world in which both agents were embedded. This use of preexisting models allows agents to create models based on their perception of whats around them, and thus be more assured that their separate models will be similar. This greatly facilitates learning about language. However human agents often communicate about models of hypothetical or fictional worlds as well. Such communications require advanced linguistic world building techniques. World building would be an interesting area of future research, and would require not only recognizing the

meaning of words, but being able to generate an instance with stereotypical values.




(a) In addition to the distance between two objects, it might be useful to measure distance to an object's directional shadow, or the convex hull surrounding two or more objects.



(b) A features such as ordering in a given direction could be useful for words like "leftmost".

Figure 6.1: Potential features for future experiments.

# REFERENCES

Artzi, Y. and L. Zettlemoyer (2013). Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational ...*.

Bahl, L. R., P. Brown, P. V. de Souza, and R. Mercer (1989). A tree-based statistical language model for natural language speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, **37**(7), pp. 1001–1008.

Bao, L. and S. S. Intille (2004). Activity recognition from user-annotated acceleration data. In *Pervasive computing*, pp. 1–17. Springer.

Barnard, K., P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan (2003). Matching words and pictures. *The Journal of Machine Learning Research*, **3**, pp. 1107–1135.

Bay, H., T. Tuytelaars, and L. Van Gool (2006). Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pp. 404–417. Springer.

Botev, Z., J. Grotowski, and D. Kroese (2010). Kernel density estimation via diffusion. *The Annals of Statistics*.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. *Proceedings of the Biennial GSCL Conference*.

Brill, E., D. Magerman, M. Marcus, and B. Santorini (1990). Deducing linguistic structure from the statistics of large corpora. In *Information Technology, 1990.'Next Decade in Information Technology', Proceedings of the 5th Jerusalem Conference on (Cat. No. 90TH0326-9)*, pp. 380–389. IEEE.

Brugman, C. and G. Lakoff (1988). Cognitive topology and lexical networks. *Lexical ambiguity resolution*, pp. 477–507.

Carbonell, J. G. (1979). Towards a self-extending parser. In *Proceedings of the 17th annual meeting on Association for Computational Linguistics*, pp. 3–7. Association for Computational Linguistics.

Church, K. W. (1988). A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, pp. 136–143. Association for Computational Linguistics.

Cooper, G. S. (1968). A semantic analysis of English locative prepositions. Technical report, DTIC Document.

Coventry, K. R., A. Cangelosi, R. Rajapakse, A. Bacon, S. Newstead, D. Joyce, and L. V. Richards (2005). Spatial prepositions and vague quantifiers: Implementing the functional geometric framework. In *Spatial Cognition IV. Reasoning, Action, Interaction*, pp. 98–110. Springer.

Coventry, K. R. and S. C. Garrod (2004). *Saying, seeing, and acting: The psychological semantics of spatial prepositions*. Psychology Press.

Cullingford, R. E. (1978). Script application: computer understanding of newspaper stories. Technical report, DTIC Document.

Dawson, C., J. Wright, A. Rebguns, M. Valenzuela Escarcega, D. Fried, and P. Cohen (2013). A generative probabilistic framework for learning spatial language.

Dostert, L. E. (1955). The georgetown-ibm experiment. *1955). Machine translation of languages. John Wiley & Sons, New York*, pp. 124–135.

Duygulu, P., K. Barnard, J. F. de Freitas, and D. A. Forsyth (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Computer VisionECCV 2002*, pp. 97–112. Springer.

Ge, R. and R. Mooney (2005). A statistical semantic parser that integrates syntax and semantics. *Proceedings of the Ninth Conference on ...*.

Geeraerts, D. and H. Cuyckens (2007). *The Oxford handbook of cognitive linguistics*. Oxford University Press, USA.

Gorniak, P. and D. Roy (2004). Grounded semantic composition for visual scenes. *J. Artif. Intell. Res. (JAIR)*, **21**, pp. 429–470.

Herskovitz, A. (1986). Language and spatial cognition. *Studies in Natural Language Processing. Cambridge University Press: Cambridge*.

Kate, R. and R. Mooney (2006). Using string-kernels for learning semantic parsers. *Proceedings of the 21st International Conference ...*.

Kate, R., Y. Wong, and R. Mooney (2005). Learning to transform natural to formal languages. *... of the National Conference on Artificial ...*.

Kellogg, C. H. (1967). Designing artificial languages for information storage and retrieval. *Automated language processing*, p. 325.

Kellogg, C. H. (1968). A natural language compiler for on-line data management. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pp. 473–492. ACM.

Krovetz, R. and W. B. Croft (1989). Word sense disambiguation using machine-readable dictionaries. In *ACM SIGIR Forum*, volume 23, pp. 127–136. ACM.

Kwiatkowski, T. and L. Zettlemoyer (2010). Inducing probabilistic CCG grammars from logical form with higher-order unification. *Proceedings of the ....*

Kwiatkowski, T. and L. Zettlemoyer (2011). Lexical generalization in CCG grammar induction for semantic parsing. *Proceedings of the ....*

Lakoff, G. (1987). Women, fire, and dangerous things: What categories reveal about the mind.

Lakoff, G. and M. Johnson (1999). *Philosophy in the flesh: The embodied mind and its challenge to western thought*. Basic Books (AZ).

Langacker, R. W. (1987). *Foundations of cognitive grammar: Theoretical prerequisites*, volume 1. Stanford university press.

Leech, G. N. (1969). *Towards a semantic description of English*. Longmans London.

Lehnert, W. G. (1977). A conceptual theory of question answering. In *Proceedings of the 5th international joint conference on Artificial intelligence-Volume 1*, pp. 158–164. Morgan Kaufmann Publishers Inc.

Lehnert, W. G. (1981). Plot Units and Narrative Summarization*. *Cognitive Science*, **5**(4), pp. 293–331.

Liang, P., M. Jordan, and D. Klein (2013). Learning dependency-based compositional semantics. *Computational Linguistics*.

Lowe, D. G. (1987). Three-dimensional object recognition from single two-dimensional images. *Artificial intelligence*, **31**(3), pp. 355–395.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pp. 1150–1157. Ieee.

Lu, W., H. Ng, W. Lee, and L. Zettlemoyer (2008). A Generative Model for Parsing Natural Language to Meaning Representations. *... Methods in Natural Language ....*

Magerman, D. M. and M. P. Marcus (1990). Parsing a Natural Language Using Mutual Information Statistics. In *AAAI*, volume 90, pp. 984–989.

Maurer, U., A. Smailagic, D. P. Siewiorek, and M. Deisher (2006). Activity recognition and monitoring using multiple sensors on different body positions. In *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, pp. 4–pp. IEEE.

Meehan, J. R. (1977). TALE-SPIN, An Interactive Program that Writes Stories. In *IJCAI*, volume 77, pp. 91–98.

Miller, G. A. and P. N. Johnson-Laird (1976). *Language and perception*. Cambridge university press Cambridge.

Niu, W., J. Long, D. Han, and Y.-F. Wang (2004). Human activity detection and recognition for video surveillance. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 1, pp. 719–722. IEEE.

O'Keefe, J. (1996). The spatial prepositions in English, vector grammar, and the cognitive map theory. *Language and space*, pp. 277–316.

O'keefe, J. and L. Nadel (1978). *The hippocampus as a cognitive map*, volume 3. Clarendon Press Oxford.

Olivier, P. and K.-P. Gapp (1998). *Representation and processing of spatial expressions*. Psychology Press.

Peters, J., J. Kober, K. Mülling, O. Krämer, and G. Neumann (2013). Towards Robot Skill Learning: From Simple Skills to Table Tennis. In *Machine Learning and Knowledge Discovery in Databases*, pp. 627–631. Springer.

Read, J. (2004). Recognising affect in text using pointwise-mutual information. *Unpublished M. Sc. Dissertation, University of Sussex, ....*

Recchia, G. and M. Jones (2009). More data trumps smarter algorithms: Comparing pointwise mutual information with latent semantic analysis. *Behavior research methods*.

Regier, T. (1996). *The human semantic potential: Spatial language and constrained connectionism*. Bradford Book.

Regier, T., L. A. Carlson, and Others (2001). Grounding spatial language in perception: An empirical and computational investigation. *JOURNAL OF EXPERIMENTAL PSYCHOLOGY GENERAL*, **130**(2), pp. 273–298.

Rice, M. L., F. Smolik, D. Perpich, T. Thompson, N. Rytting, and M. Blossom (2010). Mean length of utterance levels in 6-month intervals for children 3 to 9 years with and without language impairments. *Journal of Speech, Language and Hearing Research*, **53**(2), p. 333.

Robertson, N. and I. Reid (2006). A general method for human activity recognition in video. *Computer Vision and Image Understanding*, **104**(2), pp. 232–248.

Rublee, E., V. Rabaud, K. Konolige, and G. Bradski (2011). ORB: an efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564–2571. IEEE.

Samiuddin, M. and G. El-Sayyad (1990). On nonparametric kernel density estimates. *Biometrika*.

Schank, R. C., N. M. Goldman, C. J. Rieger III, and C. Riesbeck (1973). MARGIE: Memory Analysis Response Generation, and Inference on English. In *IJCAI*, pp. 255–261.

Schwarcz, R. M., J. F. Burger, and R. F. Simmons (1970). A deductive question-answerer for natural language inference. *Communications of the ACM*, **13**(3), pp. 167–183.

Simmons, R. F. (1965). Answering English questions by computer: a survey. *Communications of the ACM*, **8**(1), pp. 53–70.

Simmons, R. F. (1970). Natural language question-answering systems: 1969. *Communications of the ACM*, **13**(1), pp. 15–30.

Simmons, R. F., J. F. Burger, and R. M. Schwarcz (1968). A computational model of verbal understanding. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pp. 441–456. ACM.

Spranger, M. (2012). The co-evolution of basic spatial terms and categories. *Experiments in Cultural Language Evolution*, **3**, p. 111.

Su, Q., K. Xiang, H. Wang, B. Sun, and S. Yu (2006). Using pointwise mutual information to identify implicit features in customer reviews. *Computer Processing of Oriental ...*.

Tang, L. and R. Mooney (2001). Using multiple clause constructors in inductive logic programming for semantic parsing. *Machine Learning: ECML 2001*.

Taylor, J. (1986). *Contrasting prepositional categories: English and Italian*. Linguistic Agency University of Duisburg (previously Trier).

Thompson, F. B. (1966). English for the computer. In *Proceedings of the November 7-10, 1966, fall joint computer conference*, pp. 349–356. ACM.

Turk, M. A. and A. P. Pentland (1991). Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pp. 586–591. IEEE.

Veronis, J. and N. M. Ide (1990). Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proceedings of the 13th conference on Computational linguistics-Volume 2*, pp. 389–394. Association for Computational Linguistics.

Viola, P. and M. J. Jones (2004). Robust real-time face detection. *International journal of computer vision*, **57**(2), pp. 137–154.

Waller, D. and L. Nadel (2012). *Handbook of Spatial Cognition.* ERIC.

Waltz, D. L. (1978). An English language question answering system for a large relational database. *Communications of the ACM*, **21**(7), pp. 526–539.

Wilensky, R. (1978). Understanding goal-based stories. Technical report, DTIC Document.

Winograd, T. (1972). Understanding natural language. *Cognitive psychology*, **3**(1), pp. 1–191.

Wong, Y. and R. Mooney (2006). Learning for semantic parsing with statistical machine translation. *Proceedings of the main conference on Human ....*

Wong, Y. and R. Mooney (2007). Learning synchronous grammars for semantic parsing with lambda calculus. *Annual Meeting-Association for ....*

Woods, W. A. (1968). Procedural semantics for a question-answering machine. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pp. 457–471. ACM.

Woods, W. A. and R. Kaplan (1977). Lunar rocks in natural English: Explorations in natural language question answering. *Linguistic structures processing*, **5**, pp. 521–569.

Zettlemoyer, L. and M. Collins (2005). Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. In *Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pp. 658–666. AUAI Press, Arlington, Virginia.

Zettlemoyer, L. and M. Collins (2007). Online learning of relaxed CCG grammars for parsing to logical form. *In Proceedings of the 2007 Joint Conference ....*