Generating Reference to Visible Objects

A thesis presented for the degree of
Doctor of Philosophy
at the University of Aberdeen

Margaret Mitchell

2013

Declaration

I hereby declare that this thesis has been composed by me and is based on work done by me and that this thesis has not been presented for assessment in any previous application for a degree, diploma or other similar award. I also declare that all sources of information have been specifically acknowledged and all quotations distinguished by quotation marks.

Signed:

Date: 27.June.2013

Acknowledgements

This work would not have been possible without the care and insight of my supervisors, Kees van Deemter and Ehud Reiter; the guidance of my external supervisors, Ellen Bard and Brian Roark; the support and humor of the Natural Language Generation group at the University of Aberdeen and the Center for Spoken Language Understanding at Oregon Health and Science University; the constant support of Mary, my mom, and Scott, my partner; and the silliness of Wendell, my decrepit but well-meaning dog.

Summary

Research in the field of referring expression generation (REG) focuses on how a computer may refer to objects. In many of the approaches advanced in this field, a domain of *visible* objects is implicit, and generating human-like, natural-sounding expressions is an influential goal. In this thesis, I address such constraints head-on, examining how people refer to visible objects in the real world. Using previous work and the studies from this thesis, I propose an algorithm that generates human-like reference to visible objects. Rather than introduce a general-purpose REG algorithm, as is tradition, I address the sorts of properties that visual domains in particular make available, and the ways that these must be processed in order to be used in a referring expression algorithm. This method uncovers several issues in generating human-like language that have not been thoroughly studied before.

I focus on the properties of COLOR, SIZE, SHAPE, and MATERIAL, and address the issues of algorithm *determinism* and how speaker variation may be generated; *unique identification* of objects and whether this is an appropriate goal for generating human-like reference; *atypicality* and the role it plays in reference; and *multi-featured values* for visual attributes.

In Chapters 1 and 2, I discuss some background for this thesis, focusing on how the visual system works (Chapter 1) and what we can learn from previous research on reference, including work in philosophy, psychology, computational linguistics, and computer vision (Chapter 2). In Chapter 3, I run an exploratory study on reference to visible objects, discussing some aspects of initial reference that would be useful to capture and how structures for these phenomena may be represented in an algorithm. In Chapter 4, I focus in on the property of SIZE, detailing a hand-written algorithm and a machine learning approach to generate SIZE modifiers. In Chapter 5, I examine the role that atypicality plays in reference, using the properties of SHAPE and MATERIAL. In Chapter 6, I look briefly at what previous work in this thesis tells us about when people use (and when they do not use) COLOR modifiers. In Chapters 7

and 8, I introduce and extensively evaluate a referring expression generation algorithm that generates structures for initial reference to visible objects using an ideal computer vision output.

Technical contributions from this thesis include (1) an algorithm for generating SIZE modifiers from features in a visual scene; and (2) a referring expression generation algorithm that generates structures for varied, human-like reference. The main ideas I hope to communicate in this work are that *description* is an integral part of how people refer; that generating non-deterministically helps to better capture human-like reference; that referring expression generation may be improved by modelling the domain of reference; that redefining values for visual attributes as multi-dimensional rather than single-dimensional aids in generating richer, more natural variation; that probabilistically generating descriptions based on prior property likelihoods and description length can lead to human-like variation; that a knowledge base of what is typical about objects may be used to guide more descriptive reference; and that approaches in a visual domain may be influenced by how the visual system works, giving COLOR and SIZE a privileged status and providing mechanisms for properties that are *interconnected*.

Contents

Chapter 1

Introduction

How do people refer to objects? As a first pass, we can say that people refer to an object by saying *things* about it. The next step is to figure out, what are the things?

This thesis narrows the problem by looking at reference to visible objects. Shiny round beads, books wrapped in blue covers, thin pink sponges, brown ceramic bowls shaped like a flower: Reference to all these objects and more is explored to discover what characterizes initial reference to visible objects. I am particularly interested in verbal, conversational reference; the kind that a person may utter when viewing a scene with another person and discussing the objects. My findings are used to inform the design of a referring expression generation algorithm that connects vision to language by describing visual properties. I focus particularly on the visual properties SIZE, SHAPE, MATERIAL, and COLOR, and examine what role these play in reference. I do not attempt to develop a cognitive model of vision to language; rather, I use information from cognitive processes to guide the generation of varied, human-like expressions.

Throughout this thesis, I examine the descriptive nature of initial verbal reference: What do people tend to mention to a hearer when they identify an object in a set of visible objects (a *scene*)? Are there commonalities underlying this reference when a hearer is viewing the scene simultaneously with the speaker, versus when a hearer may have later access to the description? Is it reasonable to assume that speakers identify an object by producing descriptors that rule out all other competitor objects, or do we find evidence that other factors – such as visual salience or object expectations – are at play?

The type of reference that I develop I will call an *identifying description*. This is initial, exophoric reference to an object in a visual scene, produced by a speaker for a hearer

(either present at the time of utterance or receiving the description later), with the intent of directing the hearer's attention to the referent object. This reference may arise in spoken dialogue, or in typed descriptions; I implement a reference algorithm and evaluate in both typed and spoken modalities.

FIGURE 1. Visible everyday objects.

This work therefore makes three primary contributions:

(1) It **analyzes** how people describe real world visible objects through psycholinguistic experiments and corpus analysis.
(2) It **models** what people describe about these objects.
(3) It **evaluates** the proposed model against several other models on several visual corpora.

In this chapter, I lay the groundwork for the ideas and approaches that inform the thesis as a whole, covering an introduction to human vision (Section 1.1), reference and referring expression generation (Section 1.2), and summarizing relevant work in computer vision (Section 1.3). Section 1.4 provides a guide to the overall layout of the thesis, with a short summary of the content of each chapter.

1.1. Vision

To understand reference in the visual world, let us first consider the visual system itself (see Figures 2 and 3).

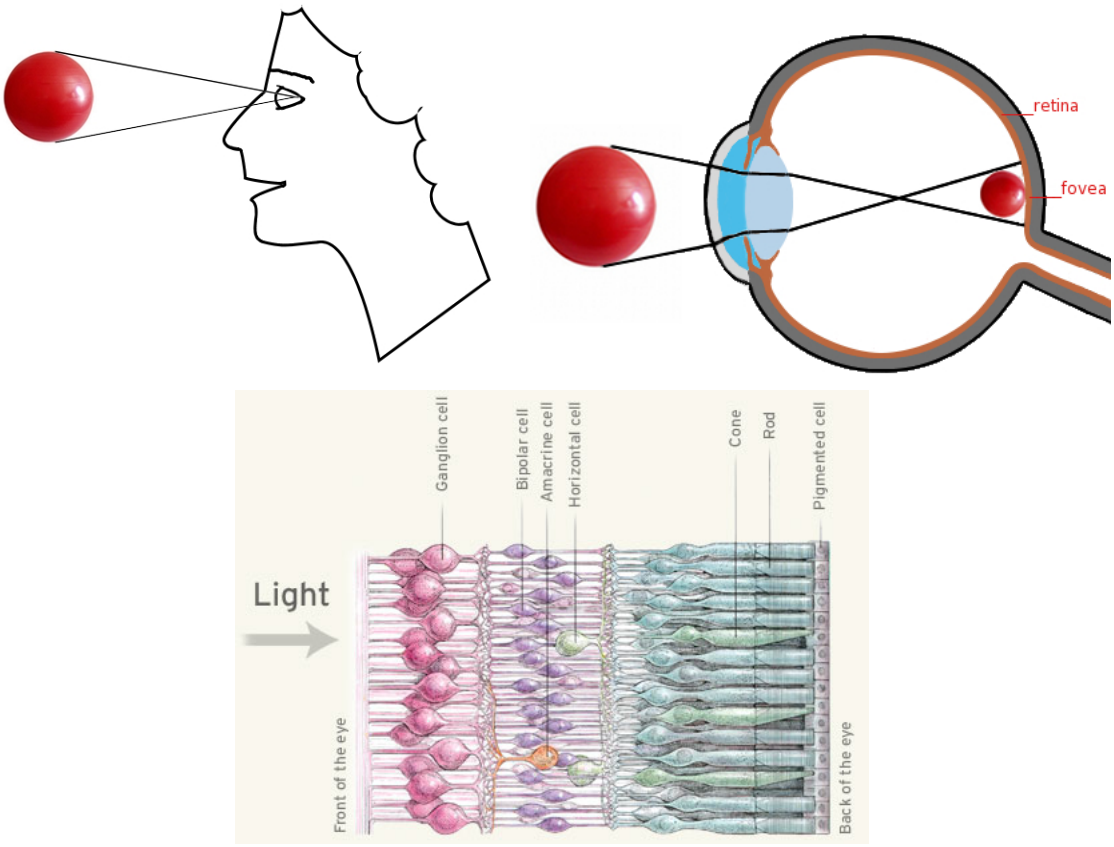


FIGURE 2. The vision system discriminates objects using color, luminance, contrast, edges, shape, etc; these features have linguistic correlates that are produced in visual description.
(Source: http://www.webexhibits.org/colorart/ganglion.html)

Free viewing of a scene is guided in part by the rods and cones in our eyes that respond to light reflecting off objects. Upon fixating on an object, cone cells in the fovea respond selectively for color, and this is passed to ganglion cells that use this information to detect contrast, defining the object's edges along with its colors and luminance.

This visual information captured in the eye travels through the optic nerve to the first cortical visual area in the brain (primary visual cortex, or V1). Here, cells respond selectively to edge orientation (Hubel & Wiesel, 1962; see Figure 4), and the size of the

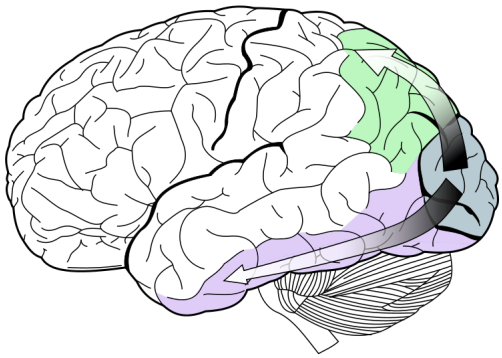


Figure 3. Originating occipitally, a ventral pathway (purple) runs to the inferior temporal lobe and processes object properties such as color and shape, while the dorsal pathway (green) projects to posterior parietal areas, processing spatial attributes and movements.
(Source: http://commons.wikimedia.org/wiki/File:Ventral-dorsal_streams.svg)

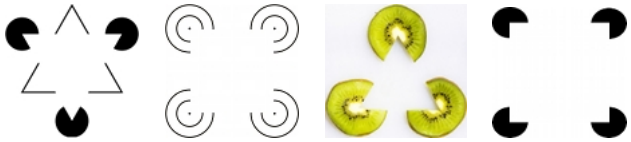


Figure 4. Experiencing edge detectors (without the edges): Cells in the visual system respond selectively to edges and shapes. You can experience this effect directly in optical illusions where the visual system responds to edges that are not there.

object is processed (Murray, Boyaci, & Kersten, 2006; Fang, Boyaci, Kersten, & Murray, 2008; Schwarzkopf, Song, & Rees, 2010). This feeds forward to areas that respond in parallel to shape (Logothetis, Pauls, & Poggio, 1995; Riesenhuber & Poggio, 1999; Tarr & Gauthier, 2000) and location (Haxby et al., 1991; Mishkin, Underleider, & Macko, 1983) (see Figure 3). Further areas of the visual system discriminate an object's texture, material, opacity, sheen, and other visual characteristics (cf. Kosslyn, 1994; Wolfe & Myers, 2010; Anderson, 2011).

It is unsurprising that the properties the visual system perceives are properties that we have language for. Color and size, for example, are processed relatively early by the visual system; they are also the most common adjectives when referring to visible objects (this is further detailed in Chapters 3 though 5).

The connection between vision and language continues as our brain uses the perceived visual features to find the object type. Particularly salient visual properties for this task include SIZE, SHAPE (E. V. Clark, 1973; Landau, Smith, & Jones, 1998), and MATERIAL (Markman, 1989). Using these features to find the object type and assign an object name requires generalizing what we see in a specific instance to a stored category label, a function of the property similarity between the current context and what we have previously seen (Kosslyn, 1994; Logothetis & Sheinberg, 1996).

Reference to visible objects is therefore brought about by the interaction between two modalities: (1) a visual, spatial input modality that creates the perception of objects as they exist in space, and (2) a language output modality that is affected by visual structures to produce certain kinds of reference. In an automatic vision to language system, visual and spatial input may be provided by a computer vision system, and I discuss this connection further at the end of the chapter. In the next section, I focus on the language aspect of this interaction, discussing what it means to *refer*, what this says about the kinds of properties that will be mentioned, and how such visual characteristics as those discussed above may be used to generate referring expressions for visible objects.

1.2. Reference

The object is a principal unit in early language learning, with the earliest vocabulary items for children being nouns that name objects (Landau, 2001), primarily concrete objects (Nelson, 1973; Caroll, 1999). As far back as Aristotle, nouns – particularly referring nouns – have been noted for being different from, and conceptually more basic than, the concepts referred to by verbs and prepositions (Aristotle, 335 BCE; Whorf, 1956; MacNamara, 1972). Nouns provide a way to communicate about the natural world's endless variety as distinct entities.

To begin understanding language generation as a whole, it is therefore reasonable to begin with the basic building block of nouns; by developing robust models of what people pick

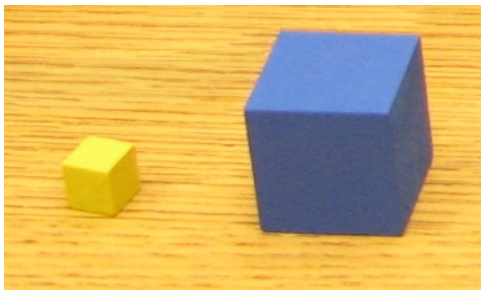


FIGURE 5. One cube, Two cube, Big cube, Blue cube? Many speakers would call the object on the right a *big blue cube*, although either *big* or *blue* can clearly and uniquely identify the referent.

out when they refer to basic, concrete objects, we can create a solid base from which to produce more complex linguistic structures. This thesis attempts to provide some groundwork in this direction. Note I do not intend to build a cognitive model; rather, I learn from what people do to develop models that can reflect their behavior.

To detail what people say when they refer, it is useful to know what reference is. Donnellan (1966) proposes a distinction between *referential* and *attributive* descriptions. Referential descriptions, he proposes, are made to enable the audience to pick out whom or what the person is talking about, while attributive descriptions are made to state something about that thing. Searle (1969) argues that there is no significant difference between the two uses; an alternative definition is that referring must either (a) contain descriptive terms true uniquely of the object; (b) present the object demonstratively, or (c) provide some combination of demonstrative presentation or description sufficient to identify it alone. Appelt and Kronfeld (1987) establish a theory that an agent is referring when he has a mental representation of an object, and uses a noun phrase with an intention of bringing a mental representation of the object to the hearer. Further details about the philosophy of reference are provided in the Literature Review in Chapter 2.

These ideas on reference give rise to several issues explored in this thesis: How do we account for *underspecification*, when speakers do not include enough information for the hearer to uniquely identify the referent? When do *overspecification* and *redundancy* come into play, when speakers include overlapping details of a referent (see Figure 5)?

It is an open question whether it is possible to create an algorithm that generates naturalistic reference for reference *in general*. This is because the form reference takes is profoundly affected by modality, task, and audience. Reference in a dialogue is affected by the focus of attention (H. H. Clark, Schreuder, & Buttrick, 1983; Beun & Cremers, 1998), and the mutual understanding of referents between interlocutors (Sacks & Schegloff, 1979; Brennan & Clark, 1996). Reference that is spoken has different characteristics from reference that is written (Chapanis, Parrish, Ochsman, & Weeks, 1977; Cohen, 1984), and reference that is produced in a collaborative task is different from reference that is produced in isolation (Krauss & Weinheimer, 1967; H. H. Clark & Wilkes-Gibbs, 1986; H. H. Clark & Krych, 2004). A speaker will refer differently to an object depending on whether he or sure can gesture towards it or manipulate it (H. H. Clark & Krych, 2004; Bard, Hill, & Foster, 2008). Even when these aspects are controlled, different people will refer differently to the same object (Reiter & Sripada, 2002; Mitchell, 2008).

1.2.1. Overview of Reference in this Thesis. The approach taken in this thesis is to to learn how natural reference behaves end-to-end. To focus this task, I examine reference that brings the attention of a hearer to a visible object for the first time in the discourse. This reference may be characterized as *descriptive* and *verbal* or *conversational* (see Chapters 3, 4, and 5) as opposed to *literary* (see Chapter 2). I do not intend to make any claims about whether this kind of reference requires that speaker and hearer be viewing the same object or gazing at the same scene at the moment of the speaker's utterance; I simply intend for the speaker to perceive visual properties of a target referent, and convey these to a hearer who also has access to the visual scene (either simultaneously or later). The hearer may use the description to successfully identify a referent, or else may be confused and not able to identify the referent, and, if present with the speaker, may interrupt or ask for clarification. This kind of reference is *exophoric*, introducing an item into the discourse focus.

The underlying assumption here, which has ramifications for the development of an algorithm that generates referring expressions, is that the speaker acts *egocentrically*, conveying visual properties that are salient to him or her, without focusing on whether they rule out distractors. A key idea is that there are tendencies in initial reference to visible objects whether the hearer is immediately present or not; these tendencies are what I hope to capture in this thesis. The hearer may either already have the object in the focus of attention, may simultaneously scan for the object as the speaker refers, or may use the speaker's full description to identify an object at a later point; the algorithm generates property descriptors independently of the hearer. This does mean the hearer will not have an effect on reference – my overall approach allows for a referring expression to be built end-to-end; however, it may also be interrupted, e.g., by a hearer who has a question, and I discuss this further in Chapter 7.

To generate human-like variation, the method I develop is also *non-deterministic*, generating properties with different likelihoods (Chapters 8 and 7). There is a preference for certain kinds of expressions in human-produced reference (Chapters 3, 4, 5), and the algorithm attempts to model these preferences using a stochastic function.

Throughout this thesis, I trace reference generation from an initial visual stimulus all the way through to the initial descriptive reference to identify an object. This approach allows me to define a clear context and specific goals, uncovering the detailed interplay between visual features and the process of reference. The algorithm I introduce may therefore be used to generate human-like reference to visible objects in particular, and the reasoning processes it uses may or may not be extended to further domains, with changes in the overall context clearly defined.

1.2.2. REG and NLG. This thesis fits within the broad research area of *natural language generation* (NLG). NLG systems are concerned with how to produce linguistic text – summaries, descriptions, etc. – from non-linguistic data – things like temperature readings and timestamps. Example systems generate weather forecasts from readings of

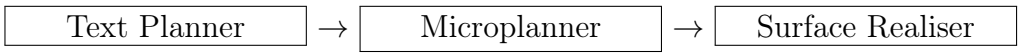


FIGURE 6. NLG pipeline architecture (after Reiter & Dale, 2000).

temperature, pressure, precipitation (Sripada, Reiter, & Davy, 2003) and summaries of daily activities for children with difficulty speaking (Black, Waller, Reiter, Tintarev, & Reddington, 2011).

In the traditional three-tiered model of language generation, *text planning* picks out what is going to be talked about; *microplanning* associates this to linguistic structures; and *surface realization* produces final utterances (see Figure 6). Similar three-tiered incremental models have been suggested in psycholinguistic research on language generation as well (Pechmann, 1989; Levelt, 1989).

A predominant research question within natural language generation systems is how best to refer to the entities being discussed. How should a weather system first be introduced ("a fast-moving cold front"), or a desired toy be selected ("the brown monkey")? This is the problem of *referring expression generation*, which generally takes place in the microplanning step of natural language generation. This thesis focuses on this subtask of referring expression generation (REG) in particular, teasing out details of reference in a visual domain.

1.2.3. Description and Reference. Although there have been different views on what reference is, all can roughly be generalized as allowing that when people *refer*, they select specific entities known by the speaker, with the intention for a hearer to:

(1) identify the entity
(2) learn information about it

Proposed computational models for reference have tended to focus on (1) – how to uniquely identify a referent so that it is distinguished from other confusable items in the scene. This follows a view first outlined by Olson (1970), where reference is "the specification of an intended referent relative to a set of alternatives." In these approaches, the

properties of the referent are compared against the properties of other items in the scene, and those that differ are selected (using a variety of different methods; further details of modern REG algorithms are available in Chapter 2 Section 2.2.3).

Choosing properties of the referent relative to a set of alternative items is a simplified view of reference that has since evolved in the psycholinguistic world (H. H. Clark & Bangerter, 2004). There are additional goals with each utterance in addition to unique identification (Jordan & Walker, 2005; Appelt, 1985), leading to more descriptive reference. This includes intentional influences (Jordan, 2000), such as those with a communicative purpose (conveying mood, certainty, etc.), conceptual pacts (Brennan & Clark, 1996) to render agreement on an object between two speakers, and conceptualizations or perspectives on the referent (H. H. Clark & Bangerter, 2004). When viewing a scene, subjects will begin referring to objects before they have even begun scanning the alternatives (Pechmann, 1989) and visual characteristics of different objects will tend to "pop out" without focusing on surrounding items (Treisman & Gelade, 1980). The idea that the primary mechanism driving human-like reference is selecting properties relative to a set of alternatives does not appear to be well-founded, at least in visual domains.

This thesis therefore attempts to propose a model of reference that brings in both (1) and (2) above, looking at what information people convey about a referent in addition to how a referent may be identified. Although there is a rich history of work focusing on what people describe when they refer, a robust computational model for the generation of descriptive initial references to everyday objects has not emerged. This thesis fills this gap, building off of earlier philosophical, psycholinguistic, and computational work on reference. Experiments are designed to further our understanding of reference to visible objects (Chapters 3, 4, and 5), and from this work, the thesis proposes an algorithm for generating natural-sounding identifying descriptions of a large set of real world objects (Chapter 7).

Examining reference to objects in a visual domain provides a straightforward extension of the sorts of reference REG research already tends to consider. Examples in the literature outline reference to objects, people, and animals that are perceptually available to both speaker and hearer (Dale & Reiter, 1995; Krahmer, van Erk, & Verleg, 2003; van Deemter, van der Sluis, & Gatt, 2006). Example referents may be referred to by their color, size, type ("dog" or "cup"), whether or not they have a beard, etc. This work also contributes to recent research examining naturalistic reference in visual domains explicitly (Kelleher, Costello, & van Genabith, 2005; Viethen & Dale, 2010; Koolen, Goudbeek, & Krahmer, 2011).

1.2.4. Real-World Visible Objects. I focus specifically on reference to *objects* – inanimate entities – as distinct from reference to people or animals. This is because the brain has evolved to recognize and process representations of inanimate entities differently from animate entities (Caramazza & Shelton, 1998), and so I suspect reference to animate entities involves slightly different processes and may be better modeled with a separate algorithm. The problem of generating natural reference is therefore pinpointed to the specific problem of generating natural reference to visible, real world objects.

One of the difficulties in examining real objects is that objects can be incredibly complex. Additional to the properties of color, size, and orientation, which have been isolated in previous studies involving semantically transparent objects (van Deemter et al., 2006; Viethen & Dale, 2008), visible real world objects exhibit the properties discussed in Section 1.1, with different textures, materials, patterns, sheen, luminance, etc., and often have parts exhibiting different values for such properties as well.

Using real world objects allows us to examine reference to the visual properties of objects as they may appear in everyday life. However, I lose a fair amount of control by using such objects, as they must be physically found and/or created (e.g., they are not from pictures). However, I hope that this exercise helps us to gain insight on how reference in visual scenes operates in the face of noisy, complex, real world objects.

1.3. Computer Vision

An important part of the research in this thesis, particularly in working towards an algorithm, is what the input is and what a scene model provides. It is useful when studying the generation problem to keep the input hypothetical, but being realistic about what the input may be and what kind of information it provides can help us to guide the construction of our downstream generation algorithms. For a system that automatically generates identifying descriptions of visible objects, automatic visual input is likely to be provided by a computer vision system that provides representations of visual objects. It therefore helps to look briefly at what computer vision can do, with an eye towards the following task of generation.

A fact to be aware of in understanding the state of the art in computer vision is that in an open domain, it basically does not work. For most images, computer vision can find hats and cat faces and people in an image that to a human observer is just a picture of a mountain. For the rest of the thesis, I am thus not focusing on computer vision; but it is important to keep in mind to motivate our input representations. With more work, we may be able to connect vision and the approaches discussed in this thesis directly.

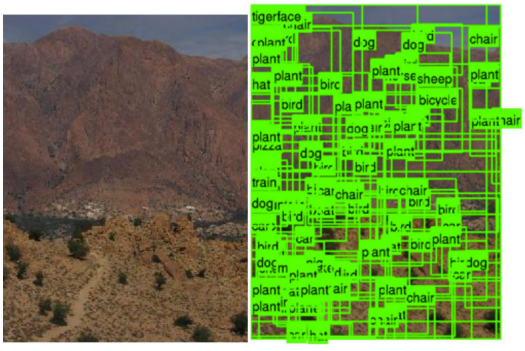


FIGURE 7. Output of running several object classifiers on a novel image.

This is not to say that computer vision does not work well when some constraints are in place. Object identification – when the system is told what kind of object to look for – achieves relatively high accuracy (Dalal & Triggs, 2005). Object segmentation –

when there is not too much clutter or lighting variation – can also work relatively well (Friedland, Jantz, & Rojas, 2005).

FIGURE 8. Output of running an object classifier for a given object on a novel image.

However, in unconstrained, novel situations, a computer vision system will tend not to make sense of the world. Until the state of the art improves, a system that links vision to language must rely heavily on further semantic knowledge to constrain what the vision system sees.

1.3.1. The State of the Art in Computer Vision. Currently, vision techniques detect objects in scenes by converting the pixels of an image into a large collection of local feature vectors, storing histograms of color, texture, intensity, and edges. Some approaches are cognitively motivated (Riesenhuber & Poggio, 1999; Itti & Koch, 2001), taking into account changes in intensity, contrast, and orientation. Several approaches to object detection are in use, with the most accurate systems finding key locations for categorization using a difference-of-Gaussian function on points in the image (Lowe, 1999; Dalal & Triggs, 2005).

A significant advance in computer vision related to this thesis is the idea of training detectors for *object attributes*. Given a bounding box where an object is suspected to exist, visual detectors may be trained for colors, materials, and parts (Farhadi, Endres, Hoiem, & Forsyth, 2009), and initial results show color detectors to have reasonable accuracy (Berg et al., 2011). Example computer vision output with attribute detections is shown

stuff:	*sky*	.999	
	id:	1	
	atts:	clear:0.432,	blue:0.945
		grey:0.853,	white:0.501 ...
	b. box:	(1,1 440,141)	
stuff:	*road*	.908	
	id:	2	
	atts:	wooden:0.722	clear:0.020 ...
	b. box:	(1,236 188,94)	
object:	*bus*	.307	
	id:	3	
	atts:	black:0.872,	red:0.244 ...
	b. box:	(38,38 366,293)	

FIGURE 9. Example computer vision output with a bounding box around the detected object, and attributes found within the bounding box. Values correspond to scores from the vision detections.

in Figure 9. Here we see several kinds of visual feature sets used in the detections (*stuff* and *object*), several detectors (*grass*, *sky*), attributes detected within the bounding box (*b. box*) where the object is predicted to exist. Values here correspond to scores from the detector for each object/stuff/attribute, and are not comparable across different items. Further details on object and attribute detection are provided in Chapter 2, Section 2.4.

Having such information available suggests that vision may begin to be connected to language at the level of the object, tying detected attributes and spatial relations computed from the bounding boxes to their corresponding surface forms.

1.4. Thesis Outline

I focus on exophoric initial reference to visible objects – the first mention of the object as it is introduced into discourse – and propose an algorithm that generates this kind of reference. Expressions produced by the algorithm can be characterized as *descriptive*, picking out an object in the visual scene by mentioning several of its properties. I assume a hearer will have access to the speaker's visual scene, and will use the speaker's descriptions to identify referents. Throughout the thesis, I focus on the attributes of

SIZE, SHAPE, MATERIAL, and COLOR, looking at what kind of attribute is used when, and how each is used; this helps to inform the design of the algorithm introduced at the end. I do not examine spatial relations or part-whole relations in great depth, but both are directly relevant and would immediately advance the work I discuss here.

Traditionally, to create an algorithm for the generation of reference, one considers a set of different properties and develops ways to decide which to include in a final surface string. This may be considered a *breadth-based* methodology, where many properties are considered, but the details of how those properties are input to the algorithm are left unspecified. Here, I begin creating an algorithm for the generation of naturalistic reference by considering individual properties – SIZE, SHAPE, MATERIAL, and COLOR – and trace how they may be realized based on a variety of different inputs and outputs. This I will call a *depth-based* methodology.

This is a departure from previous approaches to the construction of an REG algorithm. Instead of a more general-purpose algorithm, a small set of abstract semantic types are mapped to a variety of surface forms. This allows us to understand the task of referring expression generation at a fine-grained level, analyzing the specific characteristics of a visual feature that need to be considered in order to generate reference similar to that produced by people.

After a literature review in the next Chapter, I begin testing how people initially refer to visible objects in an exploratory study with no hypotheses and a great number of visual properties. From here, I suggest some basic ideas of the mechanisms at play and what aspects of reference to examine further. I find that in particular, there is need for a model for SIZE.

In Chapter 4, I therefore introduce a connection between the visible dimensions of objects and the kinds of SIZE language people use to refer to them. Height and width of the target object are compared to the rest of the objects in the scene within a function that returns an appropriate SIZE type, corresponding to surface forms like “big”, “fat”, or “short”.

Examining the SIZE property in isolation provides a tractable problem to solve within the larger problem of moving from visual input to natural language output. This makes it possible to begin untangling a few of the complex and interacting features that affect reference while minimizing conflating factors that may also affect reference, e.g., COLOR.

In Chapter 5, I focus on the role of *typicality* in reference, and the visual attributes of SHAPE and MATERIAL. This chapter explores how properties chosen to describe a referent are selected both as a function of the surrounding context, as well as a function of the expected or typical properties of referents. Chapter 6 looks briefly at COLOR, examining its prevalence throughout the studies in this thesis and considering the factors that may affect whether or not COLOR modifiers are included in a final description.

Findings from the previous chapters are implemented in a generation algorithm in Chapter 7 that produces naturalistic, descriptive referring expressions given specifications of objects' visual features, using a stochastic process to capture speaker variation. The algorithm is evaluated in Chapter 8, and in Chapter 9 I outline some of the main contributions I hope to make with this thesis.

This thesis introduces an approach to referring expression generation for exophoric, initial reference to visible objects. I aim to generate the wide variety of natural-sounding, human-like expressions of this form. This kind of reference can be characterized as both conversational and descriptive – that is, it is not constructed with the primary goal of ruling out distractors, but rather, with the goal of including information that is visually salient for the speaker. This project is framed within an engineering approach, but I will use cognitive process to inform our decisions. I find that taking a verbal, descriptive view of initial reference to visible objects captures what people appear to do better than previous work on generating reference. I hope that this work helps in the development of computational models that bridge the symbolic realm of language with the physical realm of real world referents (Herzog & Wazinski, 1994; Roy & Reiter, 2005; Tanenhaus, Spivey-Knowlton, Eberhard, & Sedivy, 1995).

Chapter 2

Literature Review

2.1. Introduction

One of the primary goals in this thesis is to discuss how to generate descriptive, human-like *referring expressions*. This literature review provides a summary of background research in areas that inform this goal, covering models of object perception, theories of language production, and strategies in computer vision.

The chapter is split into three main parts. Section 2.2 focuses on referring expressions, discussing work in the philosophy of reference in order to define what a referring expression is (Section 2.2.1); introducing some work in the psychology of reference and language generation to discuss how referring works (Section 2.2.2); and detailing modern frameworks for referring expression generation, or REG, to examine how referring expression generation has been modeled computationally (Section 2.2.3). Section 2.3 discusses research in visual perception, which guide the referring expression generation algorithm introduced in Chapter 7. Section 2.4 provides a brief overview of the state of the art in computer vision, which further informs the design of the input to the algorithm in Chapter 7.

2.2. Referring Expressions

When people identify an entity or set of entities in the discourse using an expression, they are using a *referring expression*. "The brown bowl", "me", and "Bobby McGee" can all be referring expressions. Referring expressions are a vital part of communication – a way for people to communicate what things are in the natural world – and a critical building block in the acquisition of language. Thus when speakers of mutually unintelligible

languages begin communicating, they start building a pidgin language using referring nouns ("me", "you", "money"), and when children first learn to communicate, their first words are primarily nouns (Landau, 2001) that refer to concrete objects (Nelson, 1973; Caroll, 1999).

To begin laying the foundation for a robust natural language generation system capable of describing the visual world, it therefore seems reasonable to start at the level of nouns, particularly focussing on definite descriptions; these are noun phrases of the form "the ...", as in "the brown bowl". I first begin with a review of what a definite description used referentially *is* and what people do when they refer.

2.2.1. Philosophy of Reference.

2.2.1.1. *Early views on the philosophy of reference – Donnellan and Searle.* In this thesis, I take the view that initial reference to a visible object can be descriptive, picking out properties that are interesting or visually salient rather than picking our properties that uniquely distinguish a referent from other objects. This view is motivated in part by rich philosophy on what a referring expression is, which I discuss briefly here.

The term *referring expression* can be traced to P. F. Strawson (1950), who uses the term as a shortened form for the "referring use of expressions". Strawson and subsequent philosophical work suggests that you can do useful things describing what you believe to be true of a referent, without necessarily capturing what is true. There is a separation between the properties that are *true* of the referent, and the properties that may be used to *refer* to the referent; the latter may sometimes be inaccurate, but still useful in conveying a referent to a hearer.

Following this view, the definite descriptions used to describe an item – expressions of the form "the so-and-so" – can be characterized as having a *referential* use or an *attributive* use (Donnellan, 1966). A definite description used referentially is a description that directs the audience's attention to the entity that the speaker is talking about, as well as states something about that person or thing. A definite description used attributively

may have the same form, but the speaker does not have a particular object in mind. Having a *particular object in mind* is a fundamental component of generating a referring expression.

In this work, I take the view that describing a referent may be part of both types of expressions; description is not limited to attributive phrases. In work on REG following Donnellan, description is largely separated from referring (Kronfeld, 1986; Reiter & Dale, 1992). How people *describe* when they refer – picking properties of a referent because they are salient or interesting in some way, rather than because they rule out other distractor objects – has not received much attention. In this thesis, I take the view that referring expressions can be produced via description, and what this means computationally.

An alternative view on what a referring expression is comes from John Searle (1969), who rejects the referential/attributive distinction, and argues instead that there is a distinction between illocutionary acts (asserting, requesting) and propositional acts (reference, predication). Propositional acts can only be performed in the course of performing some illocutionary act; thus we can assert information about an entity as a vehicle to refer to it. Searle proposes:

> - Whatever is referred to must exist.
> - If a predicate is true of an object it is true of anything identical with that object regardless of what expressions are used to refer to that object.
> - If a speaker refers to an object, then he identifies or is able on demand to identify that object for the hearer apart from all other objects.
>
> (Searle, 1969: 77–79)

This representation does not require that an initial referring expression *uniquely identify* an object; rather, a speaker producing a referring expression is able on demand to identify that object apart from all other objects. Searle uses the term *identifying description* in describing this kind of referring expression. This is the kind of expression that work in this thesis aims to better understand and model.

To briefly summarize, early philosophy on referring expressions tells us that the umbrella of *referring expression* includes (but is not limited to) expressions that:

(1) are definite descriptions
(2) are used to identify a referent
(3) have properties that may (or may not) be true of the intended referent
(4) may (or may not) contain description
(5) may (or may not) uniquely identify a referent

To denote the expressions that fall under this definition, I borrow Searle's term *identifying description*, and I will be using this definition in the work that follows.

2.2.1.2. *Grice.* Around the same time as Searle's work, in 1967, H. P. Grice gave a series of lectures at Harvard on the nature of conversation. His philosophy of language did not focus on reference or referring expressions, but the maxims that he proposed to underlie conversation have been extremely influential in studies on language production and models for referring expression generation (particularly Dale & Reiter, 1995).

Grice was a philosopher and a linguist, and these lectures were a product of years of research on the relationship between logic and natural language. Grice sought to represent conversation and natural language in logical terms, and believed that the fundamental processes underlying communication could be represented in logical form. Notes from these lectures circulated for years, and in 1975 a journal published a portion of this work in an article called "Logic and Conversation" (Grice, 1975). This article outlined one of the basic ideas Grice had developed, the idea that conversation operates with underlying expectations that can be characterized by four basic maxims. The maxims have apparent violations in conversation, and our interpretations of the conversational exchange are influenced by how we resolve a speaker's utterance against these maxims. The maxims have come to be known as the Gricean maxims, and are listed in Figure 1.

The maxims provide insight into the *cooperative principle* in conversation, explaining how people may balance *informativeness* with *brevity* when they communicate. This work has

Quantity:

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

Quality: Try to make your contribution one that is true.

1. Do not say what you believe to be false.
2. Do not say that for which you lack adequate evidence.

Relation: Be relevant.

Manner: Be perspicuous.

1. Avoid obscurity of expression.
2. Avoid ambiguity.
3. Be brief (avoid unnecessary prolixity).
4. Be orderly.

FIGURE 1. The Gricean Maxims.

been particularly influential in algorithmic approaches to REG (Section 2.2.3), and so is discussed here. However, they are not used to inform the computational approaches introduced in the later chapters of this thesis (at least not intentionally). As Grice writes,

> I have stated my maxims as if this purpose were a maximally effective exchange of information; this specification is, of course, too narrow. (Grice, 1975: 58)

The maxims provide guidelines for how people should speak if they want to communicate with optimal effectiveness, but not necessarily for how people tend to communicate.

The Gricean maxims may be understood as what listeners expect in some kinds of conversation, but it is clear that they are not what speakers generally do. Grice's maxim of quantity appears to be overridden by factors such as lexical availability and perceptual salience (Brennan & Clark, 1996), the maxim of quality is flouted when people guess or make ironic statements (Grice, 1975), and the maxim of manner does not account for the fact that people include unnecessary prolixity (H. H. Clark & Wilkes-Gibbs, 1986)

(see Section 2.2.2). The maxims are therefore not ideal as a model of how people naturally *generate* referring expressions, although may be useful in later work on *interpreting* referring expressions.

2.2.1.3. *Recent views on the philosophy of reference – Appelt and Kronfeld.* Philosophical work in referring expression generation began to take on a more computational framework with the thesis of Douglas Appelt (1981) and his subsequent work on the KAMP system (Appelt, 1985), which sought to computationally model the different goals at play when people communicate. Appelt saw that producing natural utterances requires a powerful system capable of reasoning not only about the physical world, but beliefs and intention of the communicants.

To begin defining what the problem of generating referring expressions is, and what kinds of knowledge a system must have to generate them, Appelt proposed that object reference may be broadly categorized under four types of *concept activation actions* when a speaker utters a noun phrase for a hearer:

(1) A referent is mutually known by speaker and hearer, where reference picks out that referent.

Example: Use *the same wrench you used to unfasten the pump.*

Category: Shared Concept Activation with Identification Intention (SI)

(2) A referent is not mutually known by speaker and hearer, where reference picks out that referent.

Example: Get me *the large wrench in my toolbox.*

Category: Nonshared Concept Activation with Identification Intention (NSI)

(3) A referent is mutually known by speaker and hearer, where reference is not intended to pick out a referent.

Example: *The man who murdered Smith* is insane.

Category: Shared Concept Activation with No Identification Intention

(4) A referent is not mutually known by speaker and hearer, where reference is not intended to pick out the referent.

<table>
<tr><td></td><td colspan="2">Identification Intention</td><td></td></tr>
<tr><td></td><td>NSI</td><td>SI</td><td></td></tr>
<tr><td rowspan="2">No Shared Knowledge</td><td>Type of noun phrase:
Referential, attributive, definite and indefinite

Planning strategy:
Useful description to facilitate hearer's identification plan

Subsumption intentions recognized only after identification is complete</td><td>Type of noun phrase:
Referential, definite including demonstratives

Planning strategy:
Efficient identifying description

Subsumption possible</td><td rowspan="2">Shared Knowledge</td></tr>
<tr><td>Type of noun phrase:
Referential, indefinite

Planning strategy:
Informative description

Subsumption impossible</td><td>Type of noun phrase:
Attributive, definite and indefinite

Planning strategy:
Efficient description

Subsumption possible</td></tr>
<tr><td></td><td>NSNI</td><td>SNI</td><td></td></tr>
<tr><td></td><td colspan="2">No identification intention</td><td></td></tr>
</table>

FIGURE 2. Appelt's 4-way reference distinction.

Example: I met *an old friend from high school* yesterday.

Category: Nonshared Concept Activation with No Identification Intention (NSNI)

Expressions with *Shared Concept Activation with No Identification Intention (SNI)* may be somewhat comparable to Donnellan's *attributive* expressions, and expressions with *Identification Intention* may include the *identifying descriptions* examined in this thesis. Whether these identifying descriptions are *Nonshared* or *Shared* in Appelt's sense is an open question; for the reference I generate, I assume that speaker and hearer view the same visual scene, although perhaps not simultaneously. It may be the case that a referent is *Shared* if it is visible (or to become visible) to both speaker and hearer, or it may be the case that a referent is *Nonshared* if the hearer has not yet identified a target object as a possible referent. However, a clear commonality between the identifying descriptions discussed in this thesis and in Appelt's reference distinctions lies in the fact that the speaker utters the expression with the intention of identifying it to a hearer; further the reference is *exophoric*, introducing an item into the discourse focus.[1]

[1]It may also be said to be *new* as opposed to *given*.

In 1986, Amichai Kronfeld proposed that the kinds of expressions discussed by Appelt (with several more fine-grained distinctions) can be generated following a three-tiered system. This is a precursor to modern referring expression generation within a natural language generation pipeline, including a *database* that includes representations of objects, a *planner* that constructs strategies for carrying out referring intentions, and an *utterance generator* that produces referring expressions.

Both Appelt and Kronfeld propose that an agent is referring when he has a mental representation of what he believes to be a particular object, and he intends the hearer to come to have a mental representation of the same object, at least in part through the use of a noun phrase that is constructed to be a linguistic representation of the object (Kronfeld, 1987; Appelt & Kronfeld, 1987). In a language generation system, this is implemented as a set of object representations that may be either *perceptual*, *discourse*, or *functional*. Relevant to this thesis, the perceptual representations are the agent's mental representations of objects that result from perceptual acts (e.g., looking).

It is not a far leap from this research to propose such a perceptual individuating set be composed of visual properties that an agent believes to be true of a target object – for example, <red, small>. Similar representations have been the basis of most work in referring expression generation to follow. Important to Appelt and Kronfeld's representation, echoing Strawson and Searle, but lacking in later computational approaches, an individuating set is the result of an agent's beliefs, not a mirror of what is actually the case.

The philosophy of reference and referring expressions has therefore provided a relatively clear picture of what a referring expression is (Section 2.2.1.1) and how it may be represented computationally (this section), providing the scaffolding from which to build a referring expression algorithm. Before turning to details of modern REG algorithms and what we may learn from them to construct an algorithm that generates more human-like reference, it is useful to bolster these considerations by studying what people actually *do*

when they refer. I turn to this issue in the following section, providing some background in the psychology of reference relevant to visual domains.

2.2.2. Psychology of Reference. In this section, I focus on areas most important for this thesis, examining the effect of *modality* on reference, in particular how reference in a visual modality in particular behaves; and the cognitive processes in reference production, specifically focusing on mental representations of objects and the incremental and parallel processes proposed to underlie object naming.

2.2.2.1. *Object Representations.* To begin the construction of an algorithm for natural reference, it is useful to develop a representation of the object before reference begins, that is accessed during the referring generation process. One extremely influential view in this regard concerns the mental representations of objects presented by Eleanor Rosch and colleagues. Rosch argued that people categorize everyday objects by comparing a given object against a 'prototypical' object within the same category. For example, chair and radio may both be categorized as *furniture*, and chair is a more reasonable exemplar of the furniture category than radio is. Rosch and Mervis (1975) showed that such natural semantic categories can be represented as networks of overlapping attributes; members of a category come to be viewed as *prototypical* of the category as a whole in proportion to the extent to which they have attributes that overlap those of other members of the category. Rosch et al. (1976) illustrated that there is one level of abstraction at which the most basic category cuts are made, a category which they termed the *basic level*. Basic level categories are those which are the most differentiated from one another, and members within a category at this level possess significant numbers of attributes in common and have similar shapes.

This suggests that the knowledge base in an REG algorithm capable of producing human-like reference should include a data structure that lists typical properties of various objects at the basic level, especially if typicality has an effect on reference. I return to the hypothesis that typicality has an effect on reference in Chapter 5.

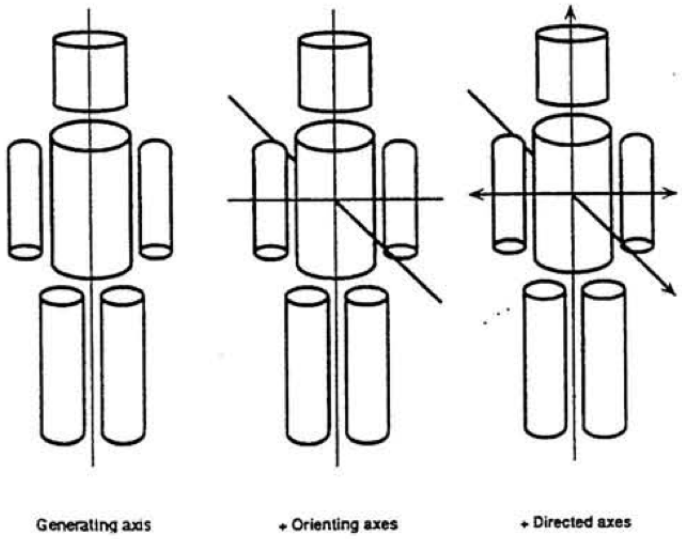


FIGURE 3. Geometric properties underlying object naming (from Landau and Jackendoff, 93: 221).

An alternative view on mental representations of objects is presented by Wu and Barsalou (2009), who showed that participants construct perceptual simulations of objects when generating properties for noun phrases. Participants produced large amounts of information about background situations associated with objects, suggesting that mental representations of objects are *situated*, bringing to mind past experiences that include the object. The authors also found that participants instructed to produce feature listings without an image created similar distributions of properties as participants instructed to describe images, which provides some evidence that the conceptual representations used by both groups were similar. From a computational perspective, this may be applied using statistics about how objects were presented in prior contexts – e.g., using a corpus containing text that describes situations involving objects – to establish the kinds of properties that people tend to associate with objects, and ultimately refer to.

Further ideas about object representation are presented by Landau and Jackendoff (1993), who discuss how a visible object may be cognitively represented during naming. Analyzing the geometric properties that underlie object nouns, the authors suggest that intersecting axes define an object's relation to space (see Figure 3). These are outlined as follows:

Generating axis: This is an object's principal axis, and can be seen as running through the top and bottom of the object.

Orienting axes: These are secondary and orthogonal to the generating axis and to each other (corresponding to the front/back and side/side axes).

Directed axes: These differentiate between the two ends of each axis, marking top/bottom and front/back.

(Landau and Jackendoff, 93: 221)

With these axes in place, the authors make a distinction between surface-type and volume-type objects, suggesting that adjectives are used differently depending on which category an object falls into. *Surface-type* objects are those that principally extend in two dimensions (such as a record), while *volume-type* objects are those that extend in all three (such as a box). The utilization of these two types to describe objects can be seen in the fact that, for example, a record only needs to be wide to be called a big record (not thick), while a box needs to be both wide and tall to be called a big cube (otherwise it would just be called tall or wide).

Appropriate reference to parts of objects can be seen as stemming from these underlying axial structures. For example, if an object is long and narrow, it has a horizontal generating axis that is longer than the other axes, and can thus be said to have ends; the regions at the termination of the axis. This idea provides a way to represent the human perception of objects, and the generation of referring expressions may benefit from incorporating these ideas of object properties. I focus on the role that the generating axis and one of the orienting axes[2] play in the generation of size modifiers for volume-type objects in Chapter 4.

Psycholinguistic work on mental object representations therefore provide some evidence for:

(1) A knowledge base of typical object properties

[2]The width or x-axis, running side/side.

(2) A dimensional representation of the objects in the scene

Although I do not attempt to design a cognitive model, this provides some guidance on the kinds of structures that may affect the kind of reference an algorithm can generate. Both (1), typical object properties, and (2), represented as the height and width of objects, are implemented in the final algorithm introduced in Chapter 7.

2.2.2.2. *Referring in a Visual Modality.* Language behaves differently in different situations and modalities, and this is why it is important to define the kind of reference being modeled when proposing an algorithm that generates human-like output. Object reference can vary as a function of beliefs, intentions, common ground, and the modality in which it is occurring. Modality includes whether the language is in a dialogue or monologue, face-to-face or over the telephone. Rubin (1980) classifies language along several lines, including voice/print, ability of speaker and hearer to interact, spatial commonality, age, and mutual involvement in the discourse. Problems are solved twice as fast when done through voice than when done through writing, even though subjects use twice as many words when speaking than when typing (Chapanis et al., 1977). Written language is syntactically more complex than spoken language, which tends to exhibit many false starts, incomplete sentences, and hesitations (Hindle, 1983). Requests for the hearer to identify referents of noun phrases dominate spoken instruction-giving discourse, but is largely absent from keyboard discourse (Cohen, 1984). It is not enough to attempt to generate "human-like" reference; spoken object reference to visible objects will not likely be the same as written object reference in a narrative.

Some early ideas about the beliefs and intentions at play during spoken reference to visible objects come from Clark et al. (1983), who establish the *Principle of Optimal Design*: "The speaker designs his utterance in such a way that he has good reason to believe that the addressees can readily and uniquely compute what he meant on the basis of the utterance along with the rest of their common ground" (p. 246). This is a forerunner to the Principle of Mutual Responsibility, discussed in Clark and Wilkes-Gibbs (1986) below.

The Principle of Optimal Design asserts the presence of a *common ground* between interlocutors, which is based in part on perceptual evidence – what the interlocutors experienced or are jointly experiencing at the moment. This may lead to linguistic *underspecification* of a referent, when speakers do not include enough properties for a hearer to uniquely identify the referent based on the semantics of those properties alone, that is still perfectly well understood in context. For example, if there are only a few stars in the sky, and I am talking about one of them, or one of them is particularly bigger or brighter than the rest, then the phrase "that star" may still identify the intended referent to the person I am talking to.

The authors examined the effect of common ground in the interpretation of underspecified reference in a visual saliency task. In this task, they elicited responses to two pictures containing bunches of flowers. The pictures were identical except that one had the target referent, a group of daffodils, appearing more vividly. Random students around campus were asked, "How would you describe the color of this flower?" The authors compared responses for the two pictures, and found that in the picture where the daffodils were only slightly more prominent than the others, 3/20 of the participants gave the color of the daffodils while 12/20 asked, "Which one?". In the other picture, where the daffodils were clearly more salient, 11/20 of the participants immediately gave the color of the daffodils and only 5/20 asked which one was meant.

This study illustrates how visual saliency plays a key role in reference to visual objects and how that reference is understood. Properties used to distinguish a referent may be underspecified based on their semantics alone, but can create sufficiently distinguishing descriptions if they denote particularly visually salient properties of the referent. For COLOR properties of objects in an image, this suggests that a system aiming to generate human-like output should not use only the COLOR value itself (e.g., *yellow*), but also the COLOR value's visual salience based on its hue, saturation, and contrast within the scene – and this is a direct link for an REG algorithm that accepts a computer vision input,

using pixels in an image to determine which properties to select in order to create an identifying description.

Following this work, Clark and Wilkes-Gibbs (1986) suggest that the "classical" analysis of reference discussed in previous work (Zipf, 1935; Brown, 1958; Olson, 1970; Krauss & Glucksberg, 1977) follows a *literary model* of definite reference, where speakers refer as if writing to distant readers. This model predicts that every reference is (a) controlled by the speaker alone (b) made with a standard (literary) noun phrase (c) that is as short as possible and yet (d) specifies the referent uniquely in that context. Literary models of reference may be well-suited for literary uses of language, such as novels, newspapers, and letters, where speakers may play, edit, and rewrite their reference; as well as non-scripted radio and television broadcasts, sermons, tape-recorded messages, etc. Such a literary model informs the referring approach behind two of the most influential algorithms in the field of REG, the Incremental Algorithm and the Graph-Based Algorithm discussed in Section 2.2.3, and the kind of reference elicited to subjects in REG corpora such as the GRE3D3 or TUNA corpora discussed in Chapter 8. This thesis questions whether such a model can actually predict what people do.

A conversational or verbal model should look quite different. In conversation, unlike writing, speakers have limited time for planning and revision. The listener has to attend to, hear, and try to understand an utterance at virtually the same time as it is being issued. Listeners in conversation are not mute or invisible during conversation, and speakers may alter what they say midcourse based on what addressees say and do. Understanding the difference between reference produced with a speaker present vs. without, written vs. spoken, is important if we wish an REG algorithm to be flexible enough to generate initial human-like reference.

In this work, I demonstrate that speakers do not generally produce reference predicted by the literary model at all; even in monologue settings (Chapter 3), reference is much closer to the conversational or verbal model. Speakers produce reference that is not

clear, reference that does not uniquely identify a referent, reference that is redundant, and reference that is overspecified. Creating a referring algorithm with this in mind (Chapter 7) leads us closer to generating human-like reference in visual settings (Chapter 8), even without an interlocutor. And using a conversational model as our starting point, it should be easier to extend to dialogue in future work.

Two interlocutors in conversation follow a process of reference synchronization that Clark and Wilkes-Gibbs call the *Principle of Least Collaborative Effort*. This principle predicts a trade off between effort in producing initial noun phrases and the effort in refashioning, with the goal being to minimize the amount of effort necessary for members of the dialogue to both identify the referent and refer to it. Following this principle, reference to an entity may be presented in installments, which minimizes the complexity of a first reference and allows the hearer to present their understanding of the reference. Reference can therefore be made with (a) nonstandard, nonliterary noun phrases, (b) with phrases the speaker believes are not adequate in context, and (c) with devices that draw addressees into the process. This occurs in non-literary description as well, even without an interlocutor present (Chapter 3).

Somewhat paradoxically, speakers are frequently *more* informative than they need to be and may describe what is salient rather than those features that will distinguish it from its neighbors (Ford & Olson, 1975; Mangold & Pobel, 1988; Brennan & Clark, 1996; Koolen et al., 2011); they are *overspecified* or *redundant*. This is especially clear in the inclusion of COLOR descriptors (Pechmann, 1989; Koolen et al., 2011). This tendency is one of the motivations for calling the initial reference I aim to generate *descriptive*.

Beun and Cremers (1998) argue that such descriptive tendencies do not contradict the Principle of Least Collaborative Effort (which they call "minimal cooperative effort"), but rather shows that such a principle does not operate at the level of descriptive features, but at the level of identification and speech acts. With this view, the principle makes predictions about the kinds of properties people will use when describing objects and the

contrast set against which a target referent will be compared. For property selection, speakers will have a preference for using absolute features (like COLOR) since these do not require comparison processes; thus SIZE and RELATIVE LOCATION expressions ("the left block", "the block next to the red one") will be dispreferred. For the contrast set, participants in a conversation will establish a focus space that enables speakers to use less information than would be needed by taking the complete conversational domain into account (a similar idea is also suggested by Grosz and Sidner (1990)). This focus space includes a visual *implicit focus*, for objects that share features with or are physically close to the one just mentioned, and objects with *inherent salience* because they stand out in the context (e.g., the pop out effect of Treisman and Gelade (1980) discussed in Section 2.3). Although I do not find a preference for absolute properties over relative properties in the studies (but rather, just a preference for COLOR, e.g., Chapters 3 and 6), I do find that using a focused contrast set of nearby objects of the same type leads to the generation or naturalistic expressions in Chapters 4 and 8.

The complementary *Principle of Mutual Responsibility* in collaborative language use predicts that participants in a conversation try to establish, roughly by the initiation of each new contribution, the mutual belief that the listeners have understood what the speaker meant in the last utterance to a criterion sufficient for current purposes. It follows that initial references are often *provisional*: when speakers present a reference, they do so as a starting point to then work with their addressees to establish that it has been understood (Brennan & Clark, 1996).

In literary uses of language, speakers (or writers) may adhere to a related *Principle of Distant Responsibility*. This predicts that the speaker tries to make sure, roughly by the initiation of each new contribution, that the addressees *should have been* able to understand his meaning in the last utterance to a criterion sufficient for current purposes. This is clearly a mode of language use immediately applicable to work in referring expression generation that does not use a hearer model. Reference without a hearer is further explored in Chapter 3, while reference with a hearer is explored in Chapter 5.

Another important factor in work that aims to generate human-like reference to visible objects is speaker variation. Furnas et al. (1987) found that the likelihood in their study that any two people would use the same label conceptualizing an action in the same way was 7-18%. In another study (Schober & Clark, 1989), pairs of people in conversation referred to the same abstract geometric form variously as "the rice bag", "the whale", "the complacent one", "the stretched-out stop sign", and "the baby in a straitjacket". Such speaker variation speaks to the kind of output that should be produced by an algorithm aiming at natural reference: Varied, individualized reference. For different speakers, or in different moments, the kind of expression that a person will produce will be different; a naturalistic reference algorithm should therefore aim to produce several possible expressions for a given referent.

2.2.2.3. *The Production Process.* Psycholinguistic models also provide some evidence as to what people do when producing language. Such work may provide insight into how to create referring outputs that otherwise would not be possible with existing REG techniques.

An extremely influential view on how reference to visible objects proceeds in a visual context is provided by Pechmann (1989). Before continuing, it is important to note that the final results of this paper are limited to the results from seven participants, one of whom had only eight usable responses, and so conclusions from this study should be followed with caution.

Using eye-tracking, subjects were presented with a variety of objects and asked to name one. This exercise showed that people begin producing descriptions of items before they scan the entire scene, and in fact, describe the target object as they fixate on the other objects in the scene. In this process, which the author calls *incremental speech production*, the features necessary to distinguish an object are not formulated before the utterance begins, but rather chosen as the utterance progresses. This is another motivation for viewing the kind of human-like initial reference I aim to generate as being descriptive

and conversational – properties are not chosen to optimally rule out other competitor objects, but are chosen because they appear salient to the speaker as he or she refers, without fully considering all the competitor objects.

The process of scanning the scene and not returning to fixate on the target referent is also well known in work in visual processing, viewing behavior known as *inhibition of return* (Posner & Cohen, 1984), and this may offer a further explanation of the participants' behavior. It may be possible that participants are not viewing each additional object in the scene in order to produce a contrastive modifier, but rather looking around the scene as they produce an utterance that has already planned. Levelt and Meyer (2000) show a related effect for object viewing and naming, where when referring to two objects (e.g., "the dog and the baby"), a speaker's attention stays on the first object just long enough to retrieve its phonological code, and then the speaker continues viewing the scene. This experiment differs from Pechmann's because the subjects were told what to say and described two objects rather than one, but it further supports the idea that people will move to view a second object before they have fully linguistically encoded a first object; whether or not people view other objects in the scene to contrast it with a target referent is unclear.

Pechmann's research has become one of the most influential psycholinguistic studies in referring expression generation. However, there are elements of Pechmann's work that are often overlooked. To describe incremental speech production, Pechmann supports a descriptive interpretation of initial reference, where properties are chosen not because they contrast the referent item with the competitor objects, but because they emerge as visually salient to the speaker. He writes:

> . . . The speaker initially pays attention to the target object without seriously considering the context. . . The speaker starts to articulate features of the target object which are easily cognizable. One such feature is, for instance, color, which can be determined without considering any other contextual objects. In contrast, describing an object as either 'small' or 'large' required comparison processes. . . Such an incremental strategy of object naming implies that the speaker does not absolutely intend to

> mention only distinguishing features of the target object while carefully trying to avoid the incorporation of any non-distinguishing information into his utterance. **It is rather characteristic of such a strategy that the speaker articulates features of the target before he had determined whether they are distinguishing or not**.
> (Pechmann, 1989: 98. Boldface my own.)

Pechmann proposes that there are at least two distinct kinds of features at play in generating reference, those that are easily cognizable and those that require comparison processes. Something like COLOR may therefore be selected whether or not it rules out any other items in the scene. The idea that not all properties are selected based on contrast with other objects in the scene, and the distinction between properties that are 'easily cognizable' (COLOR) and those that require 'comparison properties' (SIZE/LOCATION/ORIENTATION) is used in the algorithm in Chapter 7.

A hallmark of Pechmann's proposed incremental process is that words in the utterance are generated following the order in which they are cognized; the ordering of the words in the utterance come directly from the order in which the attributes are processed. Such a direct link between planning and speaking helps explains Pechmann's finding that utterances in the study often have COLOR before SIZE. However, the majority of the utterances do not follow this order: People tended to produce SIZE before COLOR. I return to what implications this may have for an algorithm in Chapter 7.

Another view of the production process is that the creation of the noun phrase proceeds with different mechanisms operating in parallel. Schriefers (1992) suggests that in an adjective-noun noun phrase, both words are accessed in parallel, with access to the noun taking longer than the adjective. However, in many languages, setting features of the adjective depends on retrieving features of the noun; e.g., in Dutch, where you must mark the noun's gender on the adjective. This suggests that adjectives are not uttered until there is a *lemma* for the head noun, a specific meaning without phonological information attached to it (Schriefers, 1993; Levelt & Meyer, 2000).

Ferreira and Swets (2002) found no evidence of incrementality while speakers produce an utterance, but found evidence for such incrementality when there was added time

pressure on generating the utterance. An important component of speech planning may therefore be to determine whether the situation calls for "blurting out" the information, or for more careful planning, and switching between the two modes are under the agent's control when producing an utterance.

These studies suggest that some aspects of the referring expression generation process may operate in parallel, and that the language production system is capable of interleaving planning processes and articulation. Language generation has a "horizontal" as well as a "vertical" aspect (Levelt, 1989; Roelofs, 1998), and the extent to which serial planning occurs is at least partly under speakers' control; when incremental processes are used depends on the intentions that motivate the speech.

Such a solution fits well within the approach to natural language generation articulated by Levelt (1989) and Bock et al. (1994; 2001), whose models of language generation are very similar to those currently used in natural language generation. In their proposed models, the generation procedure is architecturally incremental, split between the three major levels of the Message Component, where the speaker forms an idea of what s/he wishes to say; the Grammatical Component, where lemmas are created; and the Phonological Processing Component, where articulation orders are sent and the words are spoken with the appropriate sounds. When a piece of information at a level becomes available, activity in the next level is triggered. This overall incremental approach to language generation allows that operations at each level can run in parallel; while one piece of information is produced, so are other pieces of information.

Applying these ideas to the current thesis, different components of the referring expression planning process may occur in parallel, or run independently, and establishing such mechanisms may lead to more human-like expressions. I discuss such an approach in Chapter 7, positing that the process for producing the absolute property of COLOR runs in parallel to the processes for the relative properties of SIZE, LOCATION, and ORIENTATION. This distinction follows the idea of different processes for the two kinds of

properties suggested by Pechmann (1989), and may help to generate the descriptive reference this thesis explores.

2.2.3. Computational Approaches to Reference. In the previous sections, I discussed ideas behind what a referring expression is and cognitive models of how reference works. In this section, I turn to how referring expression generation has been implemented algorithmically. I focus especially on the Incremental Algorithm (Dale & Reiter, 1995) and the Graph-Based Algorithm (Krahmer et al., 2003), two approaches to REG that have gained considerable traction in the REG community, and which I evaluate against in Chapter 8. A particularly important commonality between these algorithms, and much of the work on REG that they have influenced, is the focus on *unique identification* and operating *deterministically*. Both produce one referring expression (and only one) and stop once a target item has been uniquely identified (or else fail).

As I have discussed, speakers are *varied* in their references, and the properties selected to identify a referent may not uniquely identify an object (Section 2.2.2.2). In order to get closer to the variation that humans have, the algorithm introduced in Chapter 7 produces *several* referring expression non-deterministically, and does not focus on unique identification, instead finishing as the likelihood of including the next attribute diminishes.

The same year as Pechmann's watershed work discussed in the last section, Robert Dale (1989) introduced the first explicit algorithm for the generation of referring expressions. In this paper, Dale describes the referring expression generation mechanisms used in the system EPICURE, and introduces what came to be known as the Full Brevity Algorithm. This algorithm produces the minimal description of an object necessary to uniquely identify it. That is, this algorithm produces sets with the fewest amount of attribute-values necessary to distinguish an object in a group of objects.

This kind of reference is defined as a *distinguishing description*. This term is adopted in most later work in the area. The definition Dale uses is as follows:

> Suppose that we have a set of entities U such that
> $U = \{x_1, x_2, \ldots x_n\}$
> and that we wish to distinguish one of these entities, x_i, from all the others. Suppose, also, that the domain includes a number of attributes (a_1, a_2, and so on), and that each attribute has a number of permissible values (v_{n1}, v_{n2}, and so on); and that each entity is described by a set of attribute-value pairs. In order to distinguish x_i from the other entities in U, we need to find some set of attribute-value pairs which are together true of x_i, but of no other entity in U. This set of attribute-value pairs constitutes a *distinguishing description* of x_i with respect to the context U.
> (Dale, 89: 71)

In Dale's terminology, the object being referred to is the *intended referent*, the group of entities including the intended referent is the *context set*, and the group of entities not including the intended referent is the *contrast set*.

The Full Brevity Algorithm in no way models what speakers actually do, as suggested by the research discussed in Section 2.2.1; for example, redundancy and overspecification are not allowed. It also has a high computational complexity, and is in the worst case scenario NP-hard.

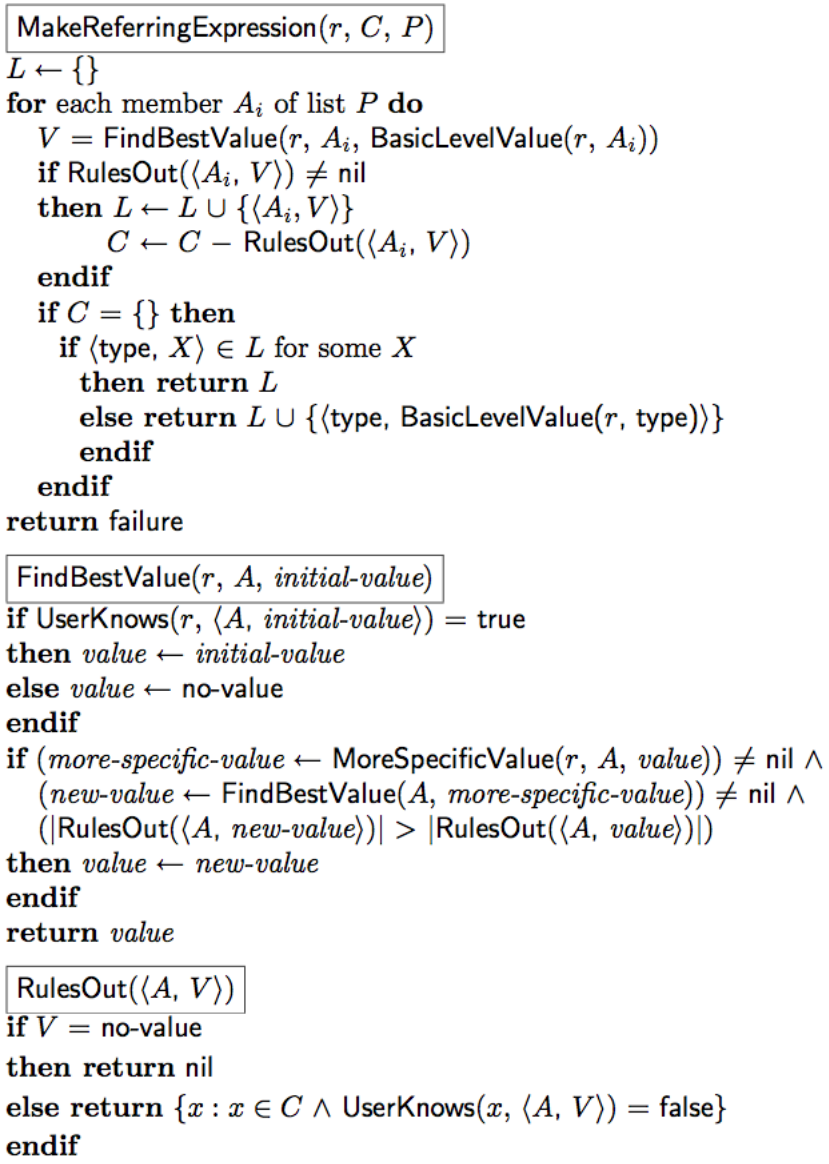
As a response to such problems, Reiter and Dale (1992) propose a new referring expression algorithm, called the Local Brevity Algorithm. This algorithm avoids some of the computational complexity of the first, checking that each description component cannot be replaced by a briefer description component without losing discriminatory power.

In this paper, referring expressions are defined as those expressions corresponding to Kronfeld's (1986) modal aspect of Donnellan's attributive/referential distinction discussed in Section 2.2.1. This is somewhat at odds with the vision of referring expressions adopted through the rest of this thesis, and so deserves some discussion here. Following Kronfeld's (1986) work (and in much REG research to follow), the referential use of an expression is presented as mutually exclusive from an attributive use. Reiter and Dale write: "We consider a noun phrase to be referential if it is *intended* to identify the object it describes to the hearer, and attributive if it is *intended* to communicate information about that

object to the hearer." This does not clearly include the possibility that a referring expression may be both intended to identify the referent as well as to attribute something to the referent. But I argue that the referential use may include attribution (Donnellan, 1966, see Section 2.2.1.1); thus a referring expression can be intended to communicate information as well as or in order to identify the referent. The view that attribution and the closely related phenomenon of description may be a part of referring expression generation is adopted throughout the thesis.

Work in the generation of referring expressions began to focus on generating human-like references in the paper "Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions" (Dale & Reiter, 1995), possibly the most cited paper in work on the generation of referring expressions. In this paper, Dale and Reiter introduce the Incremental Algorithm for generating distinguishing descriptions of referents. Drawing on the Gricean maxims (see Section 2.2.1.2) and the idea of incremental speech production developed by Pechmann (1989, see Section 2.2.2.3), the Incremental Algorithm aims to pick out a referent by incrementally analyzing the properties that are true of the referent, and finishing when the intended referent has been uniquely identified. The authors argue that such an algorithm is preferable to those algorithms introduced in their earlier work, because it is less computationally complex and more reflective of what humans actually do.

As in previous algorithms by the authors, the Incremental Algorithm operates on the properties in the context set represented as attribute-value pairs. It proceeds by iterating through attributes in a predefined order, and for each attribute, it checks whether specifying a value for that attribute would rule out at least one referent in the current discourse that has not already been ruled out. If it does, that attribute is selected. The algorithm then chooses a value for that attribute that is known to the user and that is as basic as possible while ruling out the maximum number of referents possible. Once this descriptor is selected, the algorithm adds the attribute-value to the distinguishing



```
MakeReferringExpression(r, C, P)
L ← {}
for each member A_i of list P do
   V = FindBestValue(r, A_i, BasicLevelValue(r, A_i))
   if RulesOut(⟨A_i, V⟩) ≠ nil
   then L ← L ∪ {⟨A_i, V⟩}
        C ← C − RulesOut(⟨A_i, V⟩)
   endif
   if C = {} then
     if ⟨type, X⟩ ∈ L for some X
       then return L
       else return L ∪ {⟨type, BasicLevelValue(r, type)⟩}
       endif
   endif
return failure

FindBestValue(r, A, initial-value)
if UserKnows(r, ⟨A, initial-value⟩) = true
then value ← initial-value
else value ← no-value
endif
if (more-specific-value ← MoreSpecificValue(r, A, value)) ≠ nil ∧
   (new-value ← FindBestValue(A, more-specific-value)) ≠ nil ∧
   (|RulesOut(⟨A, new-value⟩)| > |RulesOut(⟨A, value⟩)|)
then value ← new-value
endif
return value

RulesOut(⟨A, V⟩)
if V = no-value
then return nil
else return {x : x ∈ C ∧ UserKnows(x, ⟨A, V⟩) = false}
endif
```

FIGURE 4. The Incremental Algorithm (Dale and Reiter, 1995: 22)

description. This process continues until there are no longer any referents confusable with the intended referent. Pseudocode for this algorithm is given in Figure 4.

An interesting piece of this algorithm is the UserKnows function. This was not developed in the paper, but it provides a way to account for the interlocutor's model of the common ground. The algorithm thus has a way of reasoning about shared knowledge. The algorithm also calls to a BasicLevelValue function and a MoreSpecificValue function, echoing Rosch and colleagues' work (see Section 2.2.2) which are not defined as part of

the algorithm but provide system-dependent information on the basic and more specific forms for each selected attribute.

One idea to which I take particular exception in this thesis is that "computationally simple interpretations of the Gricean maxims of conversational implicature should be used" (234) to produce human-like REG. As I discussed in Section 2.2.2.2, speakers may produce initial references that they believe to be inadequate in context (H. H. Clark & Wilkes-Gibbs, 1986) and are frequently *more* informative than they need to be, including description of what is salient rather than those features that will distinguish a target object from its neighbors (Ford & Olson, 1975; Mangold & Pobel, 1988; Brennan & Clark, 1996). This suggests something beyond straightforward application of the Gricean maxims, and I do not use these maxims to guide the work in this thesis.

An alternative to the incremental approach is provided by Krahmer et al. (2003), who cast referring expression generation as a graph-based problem. The authors formalize a scene (consisting of a set of objects with various properties and relations) as a labeled directed graph and describe content selection as a subgraph construction problem. Using a graph-based approach allows for better generation of *relational expressions*, referring expressions that include references to other objects.

Pseudocode for this approach is provided in Figure 5. Assuming a scene graph G = $<V_G,E_G>$ is given, the algorithm systematically tries all relevant subgraphs H of the scene graph G by starting with the subgraph containing only the vertex v (the target object) and expanding it recursively by trying to add edges from G that are adjacent to the subgraph H constructed up to that point. This set of adjacent edges is denoted G.neighbors(H). The algorithm returns the cheapest distinguishing subgraph H that refers to v, if such a distinguishing graph exists; otherwise it returns the undefined null graph $\perp$.

```
makeReferringExpression(v) {
    bestGraph := ⊥;
    H := ⟨{v}, ∅⟩;
    return findGraph(v, bestGraph, H);
}

findGraph(v, bestGraph, H) {
    if [bestGraph ≠ ⊥ and cost(bestGraph) ≤ cost(H)]
    then return bestGraph
    fi;
    distractors := { n | n ∈ V_G ∧ matchGraphs(v, H, n, G) ∧ n ≠ v};
    if distractors = ∅ then return H fi;
    for each edge e ∈ G.neighbors(H) do
        I := findGraph(v, bestGraph, H + e);
        if [bestGraph = ⊥ or cost(I) ≤ cost(bestGraph)]
        then bestGraph := I
        fi;
    rof;
    return bestGraph;
}
```

FIGURE 5. Pseudocode for the main function in the graph-based approach (**makeReferringExpression**) and the subgraph construction function (**findGraph**).

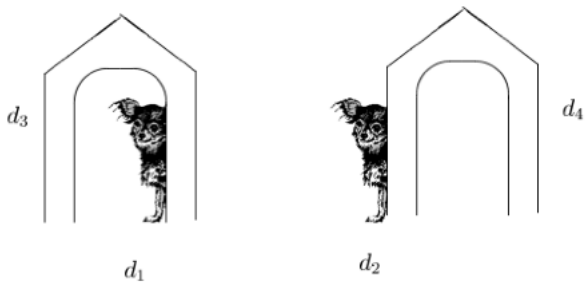


FIGURE 6. Example scene for the Graph-Based Approach.

Using this method, referring expressions are constructed based on the kind of graph that can be placed over a larger graph available from the knowledge base. Arcs between referents correspond to relations, such as *next to* and *left of*, and concentric circles represent arcs that stem from and return to the same referent, representing descriptors of that referent (see Figures 6 and 7). Costs for each circle dictate the order in which descriptors are chosen, where those with the least cost are chosen first. In this way, distinguishing descriptions for referents can be created by following a path to the referent.

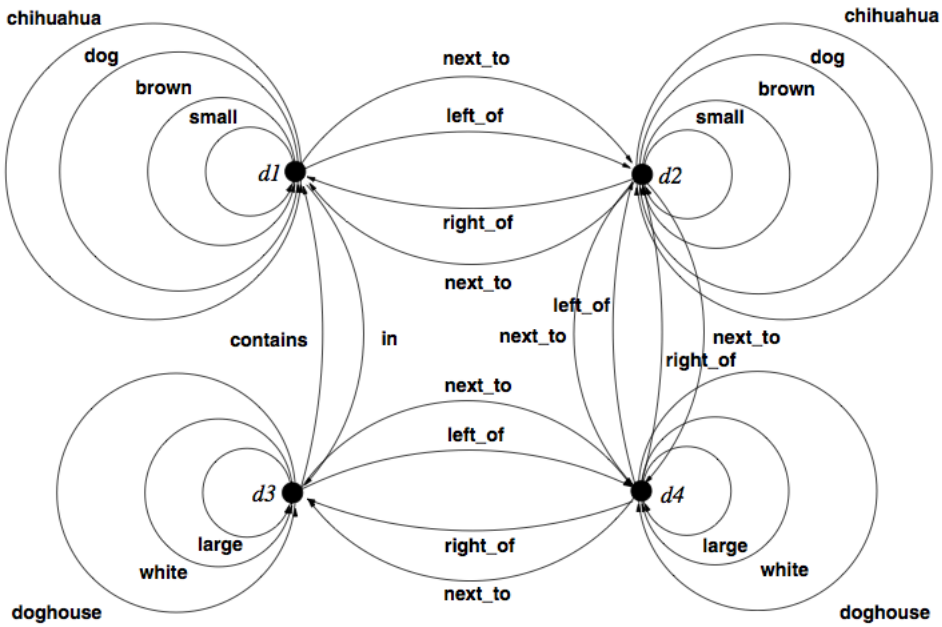


FIGURE 7. Example graph for the scene in Figure 6.

Since the publication of the Dale and Reiter (1995) paper, many steps have been made towards advancing the Incremental Algorithm's scope. Approaches to the generation of referring expressions have used the Incremental Algorithm to build reference to sets (Stone, 2000; van Deemter, 2000), generate more complex kinds of modifiers (van Deemter, 2000, 2002; Gardent, 2002; Kelleher et al., 2005), and include pointing gestures (Krahmer & van der Sluis, 2003). The Graph-Based approach has also been somewhat improved upon in recent years, with researchers proposing different methods for assigning costs to the edges in the graph in order to approximate human redundancy (Viethen, Dale, Krahmer, Theune, & Touset, 2008; Theune, Koolen, Krahmer, & Wubben, 2011) and pointing gestures (van der Sluis & Krahmer, 2005).

Work specifically useful for visual domains using an incremental framework comes from van Deemter (2000, 2006) and Kelleher and colleagues (2005; 2006; 2009). Van Deemter (2000) makes a distinction between *absolute properties*, properties that are inherent to the noun, and *gradable properties*, properties that can apply to an object to a greater of lesser degree (e.g., *small*, *large*). Van Deemter (2006) explores how absolute measurement values for the intended referent can be used to generate size-denoting adjectives based

$$Weighting = 1 - \left(\frac{P}{M+1}\right)$$

FIGURE 8. Visual salience weighting equation from Kelleher et al. (2005).

on the contrast set. In Chapter 4, I continue in van Deemter's footsteps, examining how to generate size-denoting modifiers in particular (a main component of descriptive visual expressions), but expand the kinds of SIZE language we can generate, and suggest a standalone approach that can be plugged into several different kinds of algorithms.

The related problem of generating language sensitive to the visual salience of the objects has been examined in depth by Kelleher and colleagues, who have developed models for salience – both visual and linguistic – that play a role in structuring visually situated discourse. The visual salience of an object is approximated as a function of its centrality and its size, using the weight of the pixels that compose it. A pixel is weighted using the equation shown in Figure 8 (Kelleher et al., 2005), where P is the distance between the pixel and the image center and M is the maximum distance between the image center and the image border. An object's visual saliency is then the sum of the pixel weights that compose it, normalized by the overall maximum summed pixel weight ascribed to an object in the scene.

Using visual salience scores, Kelleher et al. (2006) tackle the problem of defining the contrast set in a visual domain. The authors propose a dynamic model that orders objects in the scene by their visual saliency. Given a hierarchy of spatial relations ordered by cognitive load, their generation approach iterates through candidate landmarks in descending visual salience order, and for each, iterates through a hierarchy of spatial relations. For each relation that may be applied to the candidate landmark, a new contrast set of distractor landmarks is created, and the basic Incremental Algorithm can then be used to distinguish the target landmark from the distractor landmarks. Kelleher et al. (2009) refine such an approach further, accounting for the influence of

other objects on the semantics of spatial relations as a function of the objects' visual salience and proximity to the target object.

Throughout the thesis, I focus on properties that apply to the referent alone, and do not explore properties such as relative spatial relations that relate the referent to another object. Kelleher et al.'s models are therefore quite complementary to the ideas discussed through the rest of this thesis, and joining Kelleher's approach to visual salience and spatial relations with the algorithm introduced in Chapter 7 offers an opportunity for immediate improvement in generating natural language in visual scenes.

It is useful to take a step back at this point and again examine how we may improve the naturalness of the expressions an algorithm can generate in light of the state of the art in REG and the psycholinguistic research discussed in Section 2.2.2. Given the phenomena of under- and overspecification (H. H. Clark et al., 1983; H. H. Clark & Wilkes-Gibbs, 1986), the descriptive tendencies in initial reference (Ford & Olson, 1975; Furnas et al., 1987; Schober & Clark, 1989; Brennan & Clark, 1996; Koolen et al., 2011) the fact that when viewing a scene, subjects will begin referring to objects before they have even begun scanning the alternatives (Pechmann, 1989), that visual characteristics of different objects will tend to "pop out" without a focus on surrounding items (Treisman & Gelade, 1980) and that speech may be "blurted out" without more careful planning (Ferreira & Swets, 2002), one way to improve the generation of human-like initial expressions is to switch the focus from generating *uniquely identifying* referring expressions by selecting properties that rule out other items to generating *descriptive* referring expressions that include salient visual properties. A similar philosophy on moving away from unique identification by ruling out other items has been proposed by Siddharthan and McKeown (2005), who generate noun phrases based on distributional similarity of collocations rather than discriminatory power; and by Siddharthan, Nenkova, and McKeown (2011), who learn a model for generating references by focusing on global salience and familiarity to the reader. There has not been a great amount of work on description within referring

expression generation, but notable exceptions include Jordan and Walker (2005) and Hervás and Finlayson (2010). I will discuss these approaches briefly.

Jordan and Walker use machine learning to determine the content of referring expressions, which allows many features to be used – conceptual pact features as well as visual features – to decide the expression to generate. The idea is not to focus on properties that uniquely distinguish the referent, but to learn from a corpus of what people tend to do. Hervás and Finlayson (2010) define descriptive referring expressions as those that provide additional information not required for distinction. The authors contend that descriptive referring expressions are those that (a) unambiguously identify the intended referent and (b) contain a constituent unnecessary for identifying the referent. They present a corpus analysis in which approximately one-fifth of the referring expressions in news and narrative text are descriptive according to their definition. However, inter-annotator agreement on the descriptive referring expressions is low (Cohen's κ below 0.7), which speaks to the difficulty in separating identification from description in reference, suggesting that the distinction may not be very clear-cut.

In the next section, I focus on how people perceive objects. I examine visual perception with the hope that the processes underlying visual perception can be extended and joined with the processes underlying referring expression generation discussed above. This would lead to a model of referring expression generation for visible objects that, because it borrows from our understanding of how people see and speak about objects, may lead to more human-like referring expression generation.

2.3. Vision

To advance an algorithm capable of generating natural reference to objects presented visually, it is useful to understand the details of object perception, and how people may view a scene as they refer. This may inform the kinds of phenomena we want to account for in an algorithm. Research in this area also suggests models for representing

object perception, which when combined with the psycholinguistic models for mental representations of objects, provide a powerful basis for developing the structures an algorithm should analyze to produce human-like object reference.

In an initial glance, the visual system forms a spatial representation of the outside world that is rich enough to grasp the meaning of the scene, recognizing a few objects and other salient information in the image before attention is focused on a single object; this representation is known as the "gist" of the scene (Oliva, 2005). When fixating on an object, our eyes are aimed towards informative regions, even during the casual inspection of pictures (Mackworth & Morandi, 1967). This suggests that processing and rejection (what to focus on and what not to) must be mediated by the scene's overall gist as well as information from peripheral vision as we scan the scene. The whole display receives parallel processing within each fixation (Treisman & Gelade, 1980), with color and spatial frequency properties guiding our attention intelligently (Wolfe & Myers, 2010).

When we fixate on an object, Mishkin et al. (1983) show that there are two distinct cortical visual systems affecting our perception:

- The temporal cortex is involved in recognizing what objects look like (the ventral or "what" pathway).
- The parietal cortex determines where they are located (the dorsal "where" pathway).

Originating occipitally, the ventral pathway runs to the inferior temporal lobe and processes object properties such as color and shape, while the dorsal pathway projects to posterior parietal areas and processes spatial attributes and movements. The dorsal pathway processes locations, sizes, distances, orientations, and spatial properties in three-dimensional space, while the ventral pathway detects edges, regions of common color, texture, and geometric properties (Kosslyn, 1994). Neurons in the ventral pathway are *view-tuned*, preferentially active for specific views of objects. Such neurons act like blurred templates, with tolerance for small object rotations, and this preference is

preserved over large changes in size and position (Logothetis et al., 1995), which may be facilitated by the separation between the two pathways.

When viewing objects in a scene, property recognition from these pathways precedes object recognition. In a seminal study, Treisman and Gelade (1980) find that people perceive properties of scenes in parallel, and these then recombine in the brain to give the sense of whole objects. As such, identifying a target object requires scanning the scene only to take in the properties, the target object becoming clear once these properties are integrated. In an object finding task, a parallel search of properties is followed by serial visual fixations on the more limited set of possible targets until the true target is found. Properties which are noticed simultaneously and in parallel are suspected to be orientation, color, brightness, movement, and spatial frequency. Connecting vision to language, this fits well with the parallel planning processes suggested in work on object naming (Schriefers, 1992, 1993) and language generation (Levelt, 1989; Roelofs, 1998; Levelt & Meyer, 2000). Together, these studies suggest that proposing an algorithm that first analyzes simple properties like COLOR and SIZE in parallel (rather than serially or incrementally) may lead to the generation of more natural referring expressions.

A possible computational model of the visual representation of objects is proposed by Glasgow and Papadias (1992), who suggest representing objects as 3D and hierarchical with a corresponding object description. An object is stored as a structured, descriptive, *deep representation* that contains all the relevant information about it; it is a description of the object. This fits in well with the idea of a stored object prototype that lists typical object properties (Rosch et al., 1976) and an object description that lists situations in which the same type object has been used before (Wu & Barsalou, 2009). Following Glasgow and Papadias and connecting this with the work of Landau and Jackendoff (1993), such an object description may be accessed along with a visual object representation that depicts the space that the object takes up, and may be used to retrieve information such as size and spatial relations between parts. Representing such aspects of the object depictively avoids the combinatorial explosion that would result from needing to explicitly list

them, and provides an implementation of the visual system's ability to store prototypical shapes and exemplar shapes (Kosslyn, 1994).

With these structures in place, a system can make predictions about where people look during free-viewing, and relatedly, what they may mention. For example, including Mackworth and Morandi's (1967) finding that people look to informative regions of an image, you may see a suitcase, access its prototypical and exemplar shapes and parts and think, "there will be a handle at the top"; and then look to its top (Kosslyn, 1994). If the handle is not there, you may call the object "the suitcase without the handle".

Vision research therefore suggests some of the same ideas on mental object representations and parallel processing found in the psycholinguistic work discussed in Section 2.2.2. These complementary areas of research suggest that human reference may be affected by stored object prototypes (Rosch et al., 1976) and exemplars (Wu & Barsalou, 2009) which include property-based information for objects (Glasgow & Papadias, 1992; Kosslyn, 1994), and that the analysis of some properties may occur in parallel before an object is named (Treisman & Gelade, 1980; Mishkin et al., 1983; Schriefers, 1992; Levelt & Meyer, 2000). To implement these ideas in a computational framework, the stored knowledge of an object's typical properties can be represented in a knowledge base that is accessed during object description, and the properties accessed in the ventral stream (like COLOR) may be run in parallel to the properties accessed in the dorsal stream (like SIZE). These approaches are implemented in an algorithm in Chapter 7, and the algorithm is evaluated in Chapter 8.

2.4. Computer Vision

In developing a system that generates reference to visible objects, it is useful to define where your input comes from, and what it provides. For a system that can automatically recognize objects and describe them, automatic recognition is most plausibly provided by computer vision. That is, an expected front-end for the REG work discussed in this

FIGURE 9. ReCaptcha is used on websites to separate computers from people. Users are asked to type the letters they see; computer vision techniques at this point cannot figure out what the images say, while people can.

thesis is a visual front-end that uses computer vision. I therefore develop my approach in this thesis in light of where I expect this research to go, connecting to an automatic visual input that is visible to both speaker and hearer. Below, I provide a summary of the state of the art of computer vision, and what it makes available for an NLG system. I do not get into great detail about the mechanics of computer vision, but instead, how computer vision breaks apart a scene and what information it aims to gain from this, focusing on what is most relevant for referring expression generation.

An important thing for computational linguists to understand about computer vision is that it basically does not work. Consider the idea behind reCaptcha: words we can recognize relatively easily, despite changes in shading, color, and deformations, are impossible to detect automatically (see Figure 9). Creating a system the uses automatically retrieved visual input is not useful because the input is generally quite poor. Another approach is to design models that work with gold-standard visual input; the kinds of things computer vision is *aiming* to be able to do. In this section, I touch on some of these goals.

Objects that computer vision systems currently aim to detect include cars (Savarese & Fei-Fei, 2007), faces (Epshtein & Ullman, 2007), pedestrians (Dalal & Triggs, 2005), and household objects (Deng et al., 2009). Object recognition is difficult in part because there are so many interacting factors that are not constant across images. This includes variation in pose/orientation/viewpoint (the angles of the objects and the angles from which they are photographed); clutter (how many things are in the scene); occlusion

(objects in front of other objects); lighting (different amounts of darkness and light); shape; and size.

Approaches to object recognition include classification (Dalal & Triggs, 2005) and template matching (Biederman, 1987; Epshtein & Ullman, 2007; Su, Sun, Fei-Fei, & Savarese, 2009). There has also been progress on visual-perception based features (Itti & Koch, 2001; Serre, Wolf, & Poggio, 2005), but these generally underperform classification-based models. Current models often use overall context, or the "gist" of the scene (Rabinovich, Vedaldi, Galleguillos, Wiewiora, & Belongie, 2007; Oliva & Torralba, 2007), to guide object recognition. Recent research has also focused on creating detectors for action or pose, and these are beginning to work when developed for specific objects; for example, creating a RIDINGHORSE detector or a PERSONSITTINGINCHAIR detector (Rogez, Rihan, Ramalingam, Orrite, & Torr, 2008; Grabner, Gall, & Gool, 2011; Desai & Ramanan, 2012), although these are not yet suitable for large-scale detection.

Some on the most influential early work on computer vision comes from Irving Biederman (1987), who suggested recognizing objects by their components. The basic idea in this work is that there are a small number of geometric components that constitute the primitive elements of the object recognition system (like letters to form words). The ability to recognize a set of generalized shapes, or *geons*, would aid in recognizing objects without absolute measurements, at different orientations, sizes, modest degradations and partial occlusion.

The stages in object perception that Biederman accounts for include:

(1) Edge extraction

(2) Detection of nonaccidental[3] properties, and parsing at regions of concavity

(3) Determination of components

(4) Matching of components to object representations

[3] "Nonaccidental" properties are those properties that are unlikely to be a consequence of an accident of viewpoint. For example, a flat figure rotated 90° along the z-axis will appear to be a line segment, and this is an *accidental* property.

More recent models focus on geometry rather than on defining constituent elements (Lowe, 1999). Objects are modeled as a set of points, with relative locations between each. There is debate in the field over how to model location, how to represent appearance, whether the representation should be sparse or dense (e.g., pixels or regions) and how to handle occlusion and clutter. These approaches generally do not work well, however, due to lack of reliable methods for low-level and mid-level vision (material, sheen, etc.), as well as lack of data.

Some methods inspired by human vision include work by Itti and Koch (2001), who, usefully for this thesis, examine not only the interplay of object perception and language (or, more broadly in this paper, the interplay between object perception and any given task, which may be language generation), but also how this interaction may be modeled computationally. The authors suggest that the perceptual saliency of stimuli depend on the surrounding context, and that scene understanding and object recognition strongly constrain the selection of attended locations. A *saliency map* can be constructed based on those things that are visually salient independent of the task, which operates in parallel with a control mechanism that guides attention towards regions based on the task.

Reference to objects may therefore be brought about in this framework by the interplay of bottom-up, salient features and top-down, task-dependent control related to the need to identify the object. Together, these cues direct attention to locations in the scene and influence what sort of reference is generated about it.

Newer methods have started relying less on learning from human vision and intuitions about visual perception, and more on machine learning and classification. In these models, images are partitioned into a set of overlapping windows, and the classifier makes a decision at each window whether or not it contains the target object. A series of classifiers can then be used to give each data point in the image a label with an associated weight. For example, Felzenszwalb et al. (2010) build cascade classifiers from part-based deformable models. Dalal and Triggs (2005) use grids of histograms of oriented gradients

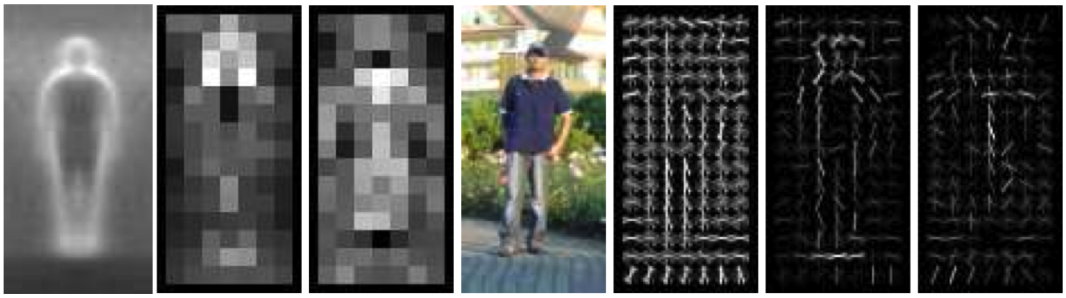


Figure 6. Our HOG detectors cue mainly on silhouette contours (especially the head, shoulders and feet). The most active blocks are centred on the image background just *outside* the contour. (a) The average gradient image over the training examples. (b) Each "pixel" shows the maximum positive SVM weight in the block centred on the pixel. (c) Likewise for the negative SVM weights. (d) A test image. (e) It's computed R-HOG descriptor. (f,g) The R-HOG descriptor weighted by respectively the positive and the negative SVM weights.

FIGURE 10. Example classification for the presence of a person using Histograms of Oriented Gradients. Image from Dalal and Triggs (2005).

(HOG) over different regions within a linear SVM. The models are considerably more robust to pose variation.

In Felzenszwalb's (2010) method, a star-structured model is used to consider all possible locations of a "root" object part. For each possible object root, the approach finds the best configuration of the remaining parts. The placement of each part is represented by its degree of displacement from its ideal location relative to the root. The score of an object configuration is then the sum of the scores of the parts at their locations minus deformation costs associated with the part displacements. The models are used to find the highest-scoring configuration, which may be specified as the coordinates encompassing the area of the configuration. Scores using such a method are illustrated in the computer vision output/NLG input shown in Chapter 1.

In Dalal and Triggs' (2005) method, an input image is first normalized for gamma/color, and then edge orientation gradients are computed using Gaussian smoothing. Each pixel casts a weighted vote for an edge orientation histogram channel based on the orientation of the gradient in which it is a part, and the votes are accumulated into orientation bins over local spatial regions. Detection windows are then tiled over the image, and in each an SVM classifier is used to decide whether there is or is not a pedestrian. Figure 10 illustrates this process, showing an example image and its weighted gradients.

Once an object has been located, it is possible to detect properties where the object is thought to exist. Farhadi et al. (2009) find that they can detect several values for shape, material, and parts. Their method uses color, texture, HOG descriptors, and edges calculated for the pixels in the image in a linear SVM to classify whether something is "round", "3D boxy", whether it has a "head", "ear", etc., and whether it is made of "plastic" or "cloth". These classifications return a score computed by the SVM. As with many things in computer vision, it often does not work. However, detecting a set of basic colors – *red*, *orange*, *yellow*, *green*, *blue*, *purple*, *black*, *brown* – works quite well using the same approach (Berg et al., 2011). To my knowledge, finding values for relative properties that require understanding the three-dimensional space represented in the image, such as SIZE and object ORIENTATION, has received less attention in computer vision.

The work discussed in this section illustrates that computer vision aims to detect objects, and returns the locations where they are likely to exist. Within this space, different absolute visual properties, like COLOR, SHAPE and MATERIAL (but not SIZE!) may be found. Input to an NLG system from a computer vision system therefore may include the height/width and pixel locations where the object is likely to exist in the image, and the different absolute properties that may be applied to it. Each is associated to a score that is relative to the recognition method used and the object or object property that is being classified.

2.5. Summary

In this chapter, I have reviewed previous work in the philosophy of reference, the psychology of reference, computational approaches to reference, object perception, and computer vision, pointing out commonalities between complementary representations advanced in each of these fields and the role they play in later work in the thesis. This review has provided historical information about the problem I address in this thesis as well as information about the kind of knowledge an algorithm should have access to in order to generate human-like reference to objects, as well as the kinds of features in the visual

scene that it could analyze, how these may be represented, and what kind of output it could generate. Many of these ideas are built on and expanded in the algorithm in Chapter 7.

To summarize, early philosophy on referring expressions tells us that the umbrella term of *referring expression* includes (but is not limited to) expressions that:

(1) are definite descriptions
(2) are used to identify a referent
(3) have properties that may (or may not) be true of the intended referent
(4) may (or may not) contain description
(5) may (or may not) uniquely identify a referent

To denote the expressions that fall under this definition, I borrow Searle's term *identifying description* (Searle, 1969) and I will be using this definition in the work that follows.

In an REG algorithm that generates human-like reference to visible objects, the formal structure from which the identifying description is constructed is a set of perceptual properties that an agent believes to be true of a target object (Appelt & Kronfeld, 1987; Kosslyn, 1994; Dale & Reiter, 1995; Krahmer et al., 2003), the result of an agent's beliefs and not a mirror of what is actually the case (Strawson, 1950; Searle, 1969; Appelt & Kronfeld, 1987). The algorithm requires a knowledge base of typical object properties (Rosch et al., 1976; Wu & Barsalou, 2009) and a dimensional representation of objects in the scene, providing, for example, height and width information (Glasgow & Papadias, 1992; Landau & Jackendoff, 1993). The analysis of some properties may occur in parallel (Treisman & Gelade, 1980; Schriefers, 1992; Levelt & Meyer, 2000), with COLOR analyzed differently from SIZE (Pechmann, 1989), LOCATION, and ORIENTATION (Mishkin et al., 1983). Input provided by a computer vision front-end may provide labels for detected objects, the approximate location where the object is found, values for different absolute visual properties like COLOR, SHAPE, and MATERIAL, and may be used to extract the height and width of the area where the object is located.

Given the phenomena of under- and overspecification (H. H. Clark et al., 1983; H. H. Clark & Wilkes-Gibbs, 1986), the descriptive tendencies in initial reference (Ford & Olson, 1975; Furnas et al., 1987; Schober & Clark, 1989; Brennan & Clark, 1996; Koolen et al., 2011) the fact that when viewing a scene, subjects will begin referring to objects before they have even begun scanning the alternatives (Pechmann, 1989), that visual characteristics of different objects will tend to "pop out" without a focus on surrounding items (Treisman & Gelade, 1980) and that speech may be "blurted out" without more careful planning (Ferreira & Swets, 2002), one way to improve the generation of human-like initial expressions is to switch the focus from generating *uniquely identifying* referring expressions by selecting properties that rule out other items to generating *descriptive* referring expressions that include salient visual properties such as COLOR and SIZE.

In the following chapters, I turn to experiments that isolate specific visual properties of the scene that this literature review suggests are particularly important in object viewing and naming, including COLOR and SIZE, and examining the effect that *typicality* of SHAPE and MATERIAL has on reference. I draw generalizations from each study that may be applied to an algorithm that generates human-like reference, and evaluate this algorithm in Chapter 8. But first, I run an initial, exploratory experiment to understand first-hand what people do when referring in visual scenes and to begin characterizing their behavior specifically for an algorithm that generates initial reference to visible objects. This experiment is detailed in the next chapter, Chapter 3.

Chapter 3

Exploring Reference to Visible Objects: Initial Findings

To begin understanding how reference to objects works in a visual domain, it is useful to run an open-ended, exploratory study that examines very generally what people do. To this end, I introduce an experiment where people refer to craft objects, physical manifestations of visible properties. Craft objects are designed to vary visual characteristics – COLOR, SIZE, SHAPE, TEXTURE, SHEEN, etc. – and this is basically their only function: to be visually rich. These objects therefore provide an interesting starting point to begin learning what people do as they refer. Some example objects are shown in Figure 1.

This study attempts to provide information about reference on a number of different dimensions. First, the study is conducted in-person, using real-world objects. This design invites referential phenomena that may not have been previously observed in

FIGURE 1. Example craft supplies.

FIGURE 2. What is the contrast set for the computer on the right?

REG research, which tends to use simpler domains. Second, the referring expressions are produced orally rather than typed out. This allows access to reference as it is generated, without the participants revising and so potentially obscuring information about their reference. Third, I use a relatively complicated task, where participants must explain how to use pieces to put together a picture of a face. The fact that I am looking at reference is not made explicit, which lessens any experimental effects caused by subjects guessing the purpose of the study. This approach also situates reference within a larger task, which may draw out aspects of reference not usually seen in experiments that elicit reference in isolation. Fourth, the objects used display a variety of different properties: TEXTURE, MATERIAL, COLOR, SIZE along several dimensions, etc. This brings the feature set closer to those features that people interact with every day.

As I have discussed in the previous chapters, one of the goals in this thesis is to understand how people generate referring expressions beyond the literary model (H. H. Clark & Wilkes-Gibbs, 1986) (see Chapter 2 for a further review of what this means). That is, this thesis removes the assumption that initial reference will be made with a standard (literary) noun phrase, that is as short as possible and yet specifies the referent uniquely in that context. In this study, reference may follow the traditional literary model, or it may not: The monologue instruction scenario offers a chance to better understand the phenomena at play during a single individual's referring expression generation, without the prior assumption of a literary model. This helps us guide hypotheses for specific phenomena in a speaker's initial reference for the experiment in Chapter 4 and in Chapter 5, where an interlocutor is present.

3.1. Motivation

Much research in REG focuses on generation from a solitary agent, referring to an item for the first time. There are exceptions, including work by Heeman and Hirst (1995), who modeled some aspects of collaborative referring expression generation by approaching the process as a series of interacting goals; the GREC challenges (Belz, Kow, Viethen, & Gatt, 2008), which sought to generate appropriate references to an entity in a context over several sentences; and experiments by Goudbeek and Krahmer (2012), who propose a model for referring expression generation that uses earlier references from the interaction.

But for a large swath of REG research, reference generation does not take into account an interlocutor or a prior context. Influential algorithms in REG such as the Incremental Algorithm (Dale & Reiter, 1995) and the Graph-Based Algorithm (Krahmer et al., 2003) produce one-shot referring expressions to uniquely identify a referent, following the traditional literary model of reference (H. H. Clark & Wilkes-Gibbs, 1986). Studying reference to real objects in a monologue instruction task therefore provides information about reference that is immediately applicable to much of the work on referring expression to date.

The fact that much of our knowledge about how human reference behaves utilizes psycholinguistic work on reference to visible objects suggests an obvious starting point for generating human-like reference: generating reference to visible objects. A system that generates such reference would be useful to provide image captions, conversation in an assistive device, or descriptions from a mobile robot. This approach is also well within the spirit of most work in REG; examples when introducing an REG algorithm often illustrate reference to objects, people, and animals that are perceptually available and physically situated in a group of competitor items (Dale & Haddock, 1991; Dale & Reiter, 1995; Krahmer & Theune, 2002; Krahmer et al., 2003; Areces, Koller, & Striegnitz, 2008), and several algorithms have worked to generate reference within visual domains explicitly (Kelleher et al., 2005; van der Sluis & Krahmer, 2005).

Given a visual domain, it is useful to examine the assumptions behind traditional, general-purpose REG and how these are borne out within the visual domain specifically. One clear difficulty can be found in nearly all published REG algorithms, which assume a predefined scene model listing the properties of the contrast set and the target object (Dale & Haddock, 1991; Dale & Reiter, 1995; Gardent, 2002; Krahmer et al., 2003; Areces et al., 2008). This includes, for example, values for the objects' COLOR, SIZE, LOCATION, etc. REG algorithms also tend to assume a clearly defined *contrast set*, the set of objects against which the target object may be contrasted.

But for a system whose goal is human-like reference using *visual input*, such assumptions are unrealistic. If input to the system is a computer vision front-end providing information about the visual scene, the information about properties that require comparison processes, such as SIZE and RELATIVE LOCATION (location relative to other objects), is not provided; instead, a vision system returns heights (size along a y-axis), widths (size along an x-axis), and approximate locations of objects within an image (see Chapter 2 Section 2.4). The contrast set is not defined in a complex visual domain (see Figure 2), and should be constructed as a smaller, focused subset of the objects in the scene (Beun & Cremers, 1998; Krahmer & Theune, 2002). To process novel visual input, we must

develop a referring approach where the locations of objects in the scene, along with their heights and widths, are used to construct scene models dynamically.

It is therefore instructive to begin characterizing the structures at play when people refer to visible real world objects. I follow in steps similar to those taken by Kelleher et al. (2005), seeking to model properties of the visual scene in an algorithm that generates natural, human-like reference. I diverge from prior work by proposing an algorithm that specifically generates reference for *real world* objects, focusing on how to model human-like reference to the kinds of objects that may be recognized in an image of a room. In later chapters, following some of the findings in this initial study, I address several of the assumptions discussed above, introducing an algorithm that produces a SIZE modifier type from the measurements of objects in the scene (Chapter 4), and using subsets of same-type objects to define the contrast set rather than all items in the scene (Chapters 4 and 8).

3.2. Introduction

This experiment looks at reference to craft items by asking subjects to describe how to recreate pictures of faces using crafts on a board. There are at least two major factors at play in this study: picking out objects, and talking about the face as a whole. In this work, I examine the initial references to pick out objects from the board; later references within the construction of the face are not analyzed.

This study is intended to be exploratory, designed with the following, basic idea: people will initially refer to visible objects using visual properties; this may not follow a literary model of reference, even in a monologue setting. Beyond that, I do not test specific hypotheses; I use this study to better understand how visual properties are used to shape our hypotheses in later work. As such, I utilize objects with a variety of visual properties (shape, texture, sheen, etc) beyond the more commonly used properties of color and size (Jordan & Walker, 2005; van Deemter et al., 2006; Viethen et al., 2008).

Because I am using objects intended for art projects with a large amount of variation, I expect that the language in this study may be more artistic or "flowery" than language in other domains. I also expect that since conversation is the usual site of language use, a monologue reference task is somewhat unusual, which may affect how well participants perform. We know humans do well at tasks that they practice regularly and get feedback on; while most linguistic tasks satisfy these conditions, monologue reference does not for most people. On the other hand, a monologue reference task sheds light on referring expression generation as it is currently approached, specifically within a highly complex visual domain, and removes the interacting factor of another speaker. This offers a starting point from which to look at further, more controlled aspects of reference.

The study reveals several interesting properties of reference that have received little attention in the field. One of the most remarkable is how people chose to refer to the objects, which can be best characterized as *description*. This phenomenon was also noted in dialogue by Clark and Wilkes-Gibbs (1986), and falls well within our understanding of how people visualize and talk about objects in a scene (see Chapter 2). Rather than introducing objects by uniquely identifying them with a minimal set of properties, participants tended to overspecify the object they intended, including detailed information about the objects' parts and analogies to other things that the object is most like. This suggests that people select distinguishing properties not just as a function of their discriminatory power (how many objects they rule out) or a linear preference order (selecting properties one-by-one from a list of preferred properties), but by how visually salient they are within the scene, and how they compare to stored representations of similar objects. Similar indications have also been found in previous work in vision (cf. Treisman and Gelade (1980)) and cognitive models of object recognition (Rosch and Mervis (1975); Rosch et al. (1976); Wu and Barsalou (2009)).

Other aspects of reference found in this study that have not yet been addressed in REG include the use of part-whole modularity, size comparisons across three dimensions, and analogies. These phenomena are interrelated, and may be possible to represent in a

computational framework. I also find that the object TYPE – corresponding to the object's head noun in REG algorithms – may often correspond to another property, particularly SHAPE or MATERIAL. This has also been found in psycholinguistic research on object naming (Markman, 1989; Landau & Jackendoff, 1993).

In the next section, I describe the study. In Section 3.4, I analyze the results and discuss what they tell us about natural reference. In Section 3.5, I draw on the results and cognitive models of object recognition to begin building the framework for a referring expression algorithm that generates naturalistic reference to objects in a visual scene. In Section 3.7, I offer concluding remarks and outline areas for further study.

3.3. Method

3.3.1. Subjects. The subjects were 20 residents of Aberdeen, Scotland, and included undergraduates, graduates, and professionals. All were native speakers of English, had normal or corrected vision, and had no other known visual issues (such as color-blindness). Subjects were paid for their participation. Two recordings were left out of the analysis: one participant's session was not fully recorded due to a software error, and one participant did not pick out many objects in each face and so was not included. The final set of participants included 18 people, 10 female and 8 male.

3.3.2. Materials. A board was prepared with 51 craft objects. The objects were chosen from various craft sets, and included pom-poms, pipe-cleaners, beads, and feathers (see Table 1). The motley group of objects had different colors, textures, shapes, patterns, and were made of different materials. Similar objects were grouped together on the board, with a label placed underneath. This was done to control the head noun used in each reference. The objects were used to make up 5 different craft "face" pictures. Subjects sat at a desk facing the board and the stack of pictures. A picture of the board is shown in Figure 3 and the faces that subjects described are shown in Figure 4. An annotated board is shown in Figure 5, and annotated faces are available in Appendix A.

14 foam shapes	2 large red hearts t3,t6/A1,A5
2 small red hearts A2,A4	2 small neon green hearts t4,t5
2 small blue hearts t2,t7/B7,B9	1 small green heart A3
1 green triangle C10	1 red circle C7
1 red square C6	1 red rectangle t12
1 white rectangle B11	
11 beads	4 large round wooden beads A6/C11-C14/D5
2 small white plastic beads B4,B5	2 brown patterned beads t1,t8
1 gold patterned bead t11	1 shiny gold patterned heart D4
1 red patterned heart D6	
9 pom poms	2 big green pom-poms A9/C8,C9
2 small neon green pom-poms t9,t10	2 small silver pom-poms B2,B3
1 small metallic green pom-pom A7	1 large white pom-pom D2
1 medium white pom-pom D3	
8 pipe cleaners	1 gold pipe-cleaner D10
1 gold pipe-cleaner in half B1	1 silver pipe-cleaner A10
1 circular neon yellow soft pipe-cleaner A11	1 neon orange puffy pipe-cleaner B12
1 grey puffy pipe-cleaner C15/D1	1 purple/yellow striped pipe-cleaner t13
1 brown/grey striped pipe-cleaner A8	
5 feathers	2 purple feathers C2,C4/D9
2 red feathers C1,C5	1 yellow feather C3
3 ribbons	1 gold sequined wavy ribbon B6
1 silver wavy ribbon B10/D7	1 small silver wavy ribbon A12/D8
1 star	1 gold star B8

TABLE 1. Board items with annotation labels. Letters correspond to the face the item is used in. Items of the same type where only one is used in the face (e.g., 4 large round wooden beads) are given a single label (e.g., A6).

Subjects were recorded on a head-mounted microphone, which fed directly into a laptop placed on the left of the desk. The open-source audio-recording program Audacity (Mazzoni, 2010) was used to record the audio signal and export it to wave format.

3.3.3. Procedure. Subjects were told to give instructions on how to construct each face using the craft supplies on the board (the instructions given to the participants is available in Appendix B). They were instructed to be clear enough for a listener to be able to reconstruct each face using the same craft objects as shown in the pictures. We instructed participants to use the same craft objects as shown to make it clear that we

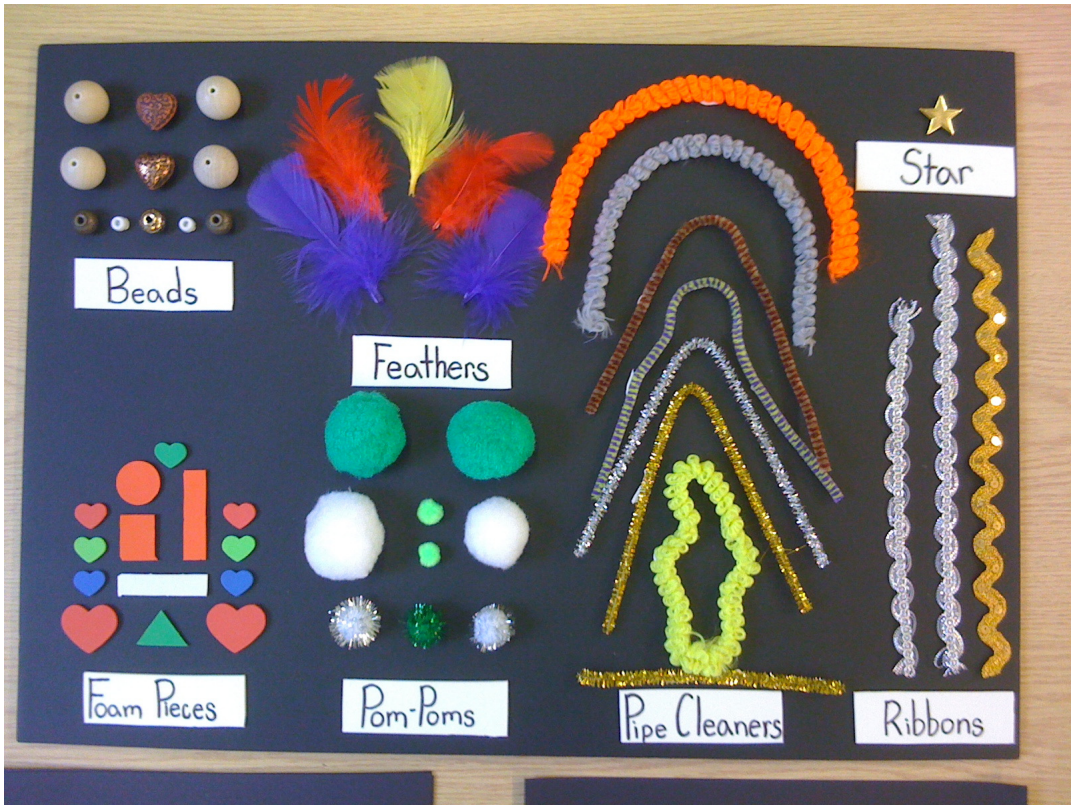
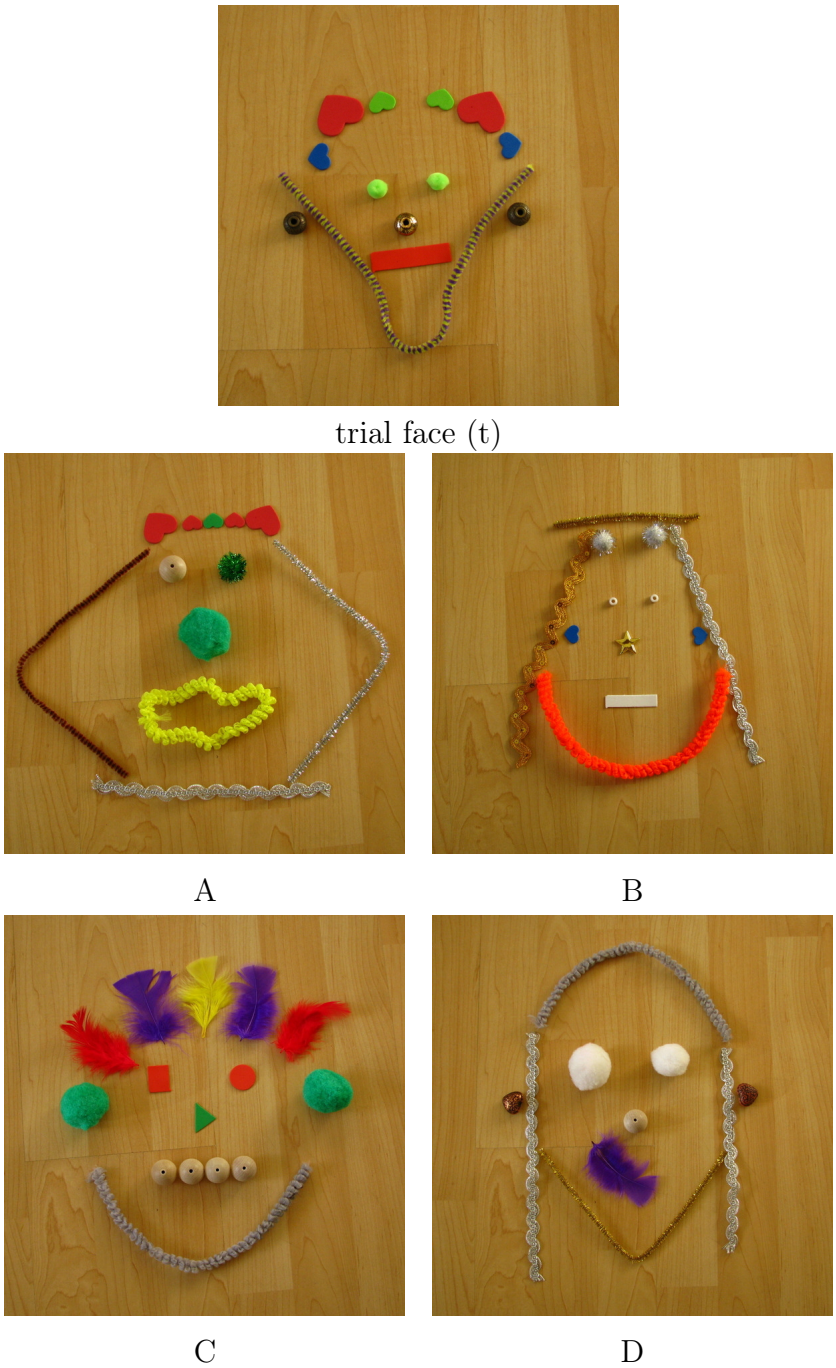


FIGURE 3. Object board.

desired *specific* objects. A pilot study revealed that the original instructions left some subjects spending an inordinate amount of time on the exact placement of each piece, and so in subjects were additionally told that each face should take "a couple" of minutes, and that the instructions should be as clear as possible for a listener to use the same objects in reconstructing the pictures without being "overly concerned" with the details of exactly how each piece is angled in relation to the other.

Subjects were first given a practice face to describe, trial face t in Figure 4. This face was the same face for all subjects. The subjects were then allowed to voice any concerns or ask questions, but the experimenter only repeated portions of the original instructions; no new information was given. The subject could then proceed to the next four faces, which were in a random order for each subject. A transcript of a single face from a session is provided in Figure 6.



trial face (t)

A B

C D

FIGURE 4. Faces used in study.

3.3.4. Analysis. The recordings of each monologue were transcribed (by me alone), including disfluencies, and each face section ("eyes", "chin", etc.) was marked. First reference to items on the board were annotated with their corresponding item numbers,

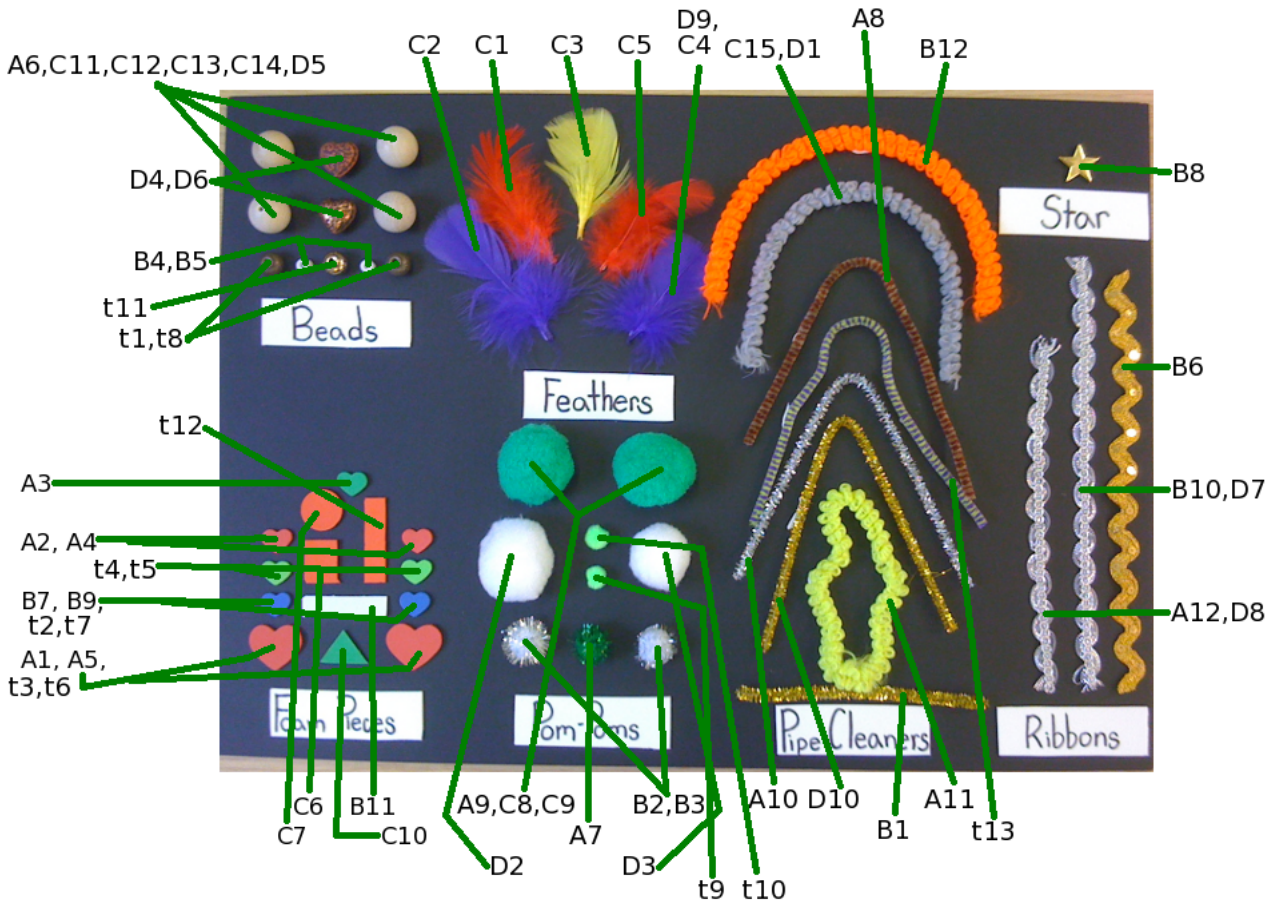


FIGURE 5. Object board with annotations. Letters correspond to the face the item is used in.

yielding 722 references.[1] Initial references to single objects were extracted, creating a final data set with 522 references to single objects.[2] I do not examine reference to parts of the face, but reference to pick out objects from the board.

There are two issues that arose in annotating the data. One concerns the use of definite versus indefinite determiners ("the" versus "a") in the noun phrases used to pick out the referents, and the other concerns the distinction between referring to an object type (any one of its kind will do) and referring to an object token (one specific object).

[1]This corpus is available at http://www.m-mitchell.com/corpora/craft_corpus/.

[2]Originally published as 505 references – one referent was missed due to a notation error.

3.3.4.1. *Definite/Indefinite Distinction.* Initial references to objects on the craft board are *indefinite* in about half of the data, beginning with the determiner "a" rather than "the". 41% (213/522) begin with the indefinite determiner and 41% begin with the definite determiner (212/522).

It deserves some consideration whether the indefinite noun phrases should be regarded as referring expressions. Referring expressions are defined as *definite* (Strawson, 1950; Donnellan, 1966; Dale & Haddock, 1991; Dale & Reiter, 1995; Krahmer & Theune, 2002), and for noun phrases this is indicated by the presence of the definite determiner "the". Dale and Haddock (1991) write: "If we have a distinguishing description, a definite determiner can be used, since the intended referent is described uniquely in context." Linguistic tradition is that "[unique identifiability] is both necessary and sufficient for appropriate use of the definite article *the*" (Gundel et al. 1993: 277). However, I find that intended referents are described uniquely, in context, using *indefinite* determiners as well as definite. For example, the gold ribbon (item B6) is referred to as both "the gold ribbon" and "a gold ribbon". This is true despite the fact that subjects were told to give instructions for someone to use the same objects, with the same board in front of them (see Appendix B).

It makes sense that initial reference to some objects contains an indefinite determiner because initial reference tends to be marked by an indefinite determiner, serving to introduce the referent into the discourse. Although the task of REG is generally construed as one of producing initial definite reference, the fact that initial reference is usually *indefinite* – and the related issue that subsequent *definite* reference to the object will tend to be reduced descriptions (Krahmer & Theune, 2002) – has not received attention in most REG algorithms.

The reason that some subjects use an indefinite determiner may be due to the fact that this is a monologue setting: Because the hypothetical listener cannot see the board at the time that the subject is giving instructions, the use of the indefinite article serves to

introduce the object into the common ground (cf. H. H. Clark and Wilkes-Gibbs (1986); Horton and Keysar (1996); Bard et al. (2000)). Because subjects use both kinds of articles (both "a" and "the") for the same objects when they initially refer to them, and I am interested in understanding what kinds of properties people pick out in initial reference, I do not make a distinction between the two forms. I analyze both as initial reference to visible objects, informing the kind of reference I aim to model in this thesis; whether the subset of expressions that use the indefinite determiner are rightly called *referring expressions* or not is noted for future work.

3.3.4.2. *Type/Token Distinction.* The indefinite/definite distinction is especially striking in references to one object of several that are of the same type. For example, the wooden beads (see Figure 3) are virtually identical. When a craft face involved a wooden bead (for example, Face D in Figure 4), subjects regularly used the indefinite determiner. In these cases, subjects did not appear to be making a unique reference to a single object token, but rather a reference to an object *type* – instructing listeners to pick up "a wooden bead", any one would do. This can be interpreted as doing one of two things: (1) failing to uniquely identify a single object token or (2) succeeding in uniquely identifying a single object type. For many references, it is unclear if the reference is underspecified or picking out an object token, for example, the phrase "a small green heart" is used to refer to the darker small green heart in the set of small green hearts (object A3). As above, I include all initial references in the data set, and note the issue for future work.

3.4. Results

I annotated each reference for the properties used to pick out the referent. For example, "the red feather" was annotated as containing the <ATTRIBUTE:value> pairs <COLOR:red, TYPE:feather>. Discerning properties from the modifiers used in reference is generally straightforward, and all of the references produced may be partially deconstructed using such properties. Table 1 shows the frequency of each attribute annotated in this corpus, with example values. COLOR is a predominant attribute, followed by SIZE,

<CHIN> Okay so this face again um this face has um uh for the chin, it uses (D10 *a gold pipe-cleaner in a V shape*) where the bottom of the V is the chin. </CHIN>

<MOUTH> The mouth is made up of (D9 *a purple feather*). And the mouth is slightly squint, um as if the the person is smiling or even smirking. So this this smile is almost off to one side. </MOUTH>

<NOSE> The nose is uh (D5 *a wooden bead, a medium-sized wooden bead with a hole in the center*). </NOSE>

<EYES> And the eyes are made of (D2,D3 *white pom-poms*), em just uh em evenly spaced in the center of the face. </EYES>

<FOREHEAD> Em it's see the person's em top of the person's head is made out of (D1 *another, thicker pipe-cleaner that's uh a grey color, it's kind of uh a knotted blue-type pipe-cleaner*). So that that acts as the top of the person's head. </FOREHEAD>

<HAIR> And down the side of the person's face, there are (D7,D8 *two ribbons*) on each side. (D7,D8 *And those are silver ribbons*). Um and they just hang down the side of the face and they join up the the grey pipe-cleaner and the top um of the person's head to the to the chin and then hang down either side of the chin. </HAIR>

<EARS> And the person's ears are made up of (D4,D6 *two beads, which are um love-heart-shaped beads*), where the points of the love-hearts are facing outwards. And those are just placed um around same em same em horizontal line as the nose of the person's face is. </EARS>

FIGURE 6. Excerpt transcript, face D.

then MATERIAL and SHAPE/FORM. Words denoting both SHAPE and MATERIAL often appear as the head noun, represented in the table as type/shape and type/material.

Using sets of properties to distinguish referents is nothing new in REG. Algorithms for the generation of referring expressions commonly use this as a starting point, proposing that properties are organized in some linear order (Dale & Reiter, 1995) or weighted order (Krahmer et al., 2003) as input. However, there is evidence that more is at play. Examples of referring expressions current REG algorithms cannot produce from the visual information available on the craft board are listed in Table 4.

3.4.1. From Dialogue to Monologue: How Speakers Introduce Referents. Because much of the influential psycholinguistic work on referring examines reference with an interlocutor (Krauss & Glucksberg, 1969; H. H. Clark et al., 1983; H. H. Clark & Wilkes-Gibbs, 1986; Beun & Cremers, 1998; Bard et al., 2000; Brennan & Clark, 1996; Bard et al., 2008), it is useful to check if the kinds of referential phenomena noted

Attribute	Frequency	Example
COLOR	594	red, green, silver, purple, yellow
SIZE	192	big, medium, small, short, thick, long
SHAPE/FORM	156	heart, circle, ball, square, sphere, bent, twisty
TYPE/MATERIAL	94	foam piece
TYPE/SHAPE	89	heart, square, circle, rectangle
MATERIAL	73	foam, wooden, tinsel, plastic, bronze
SHEEN	22	sparkly, glitter, shiny, luminescent
TEXTURE	16	fluffy, fuzzy, furry
ORIENTATION	12	upside-down, horizontal
PATTERN	3	striped, (with a) pattern
LOCATION	1	"at the bottom of the. . . presentation"

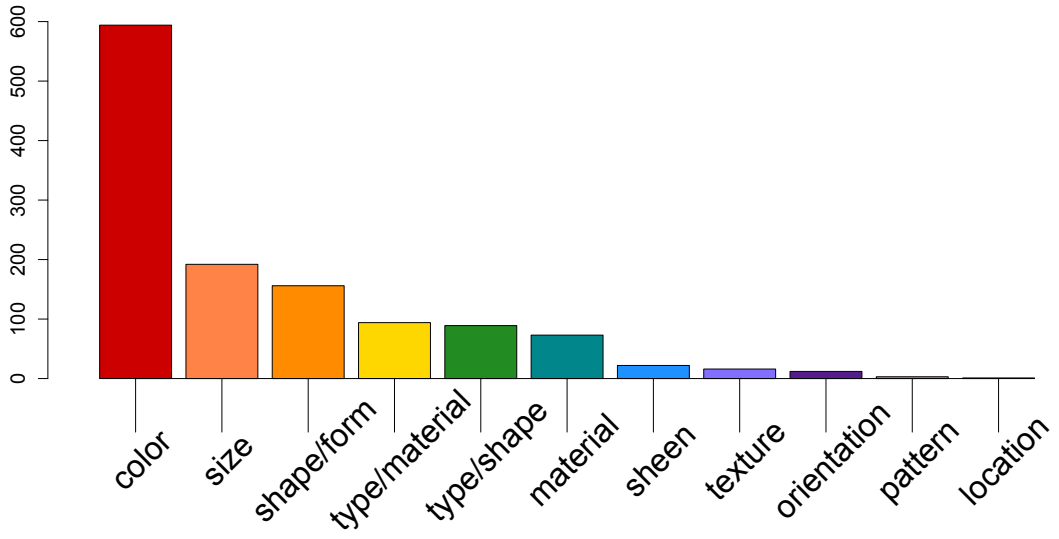


TABLE 2. Craft Corpus attribute frequencies.

in dialogue applies to monologue. If there are similar tendencies, then we have some support that the generalizations of what people do when referring in dialogue applies to what people do when referring in monologue.

Clark and Wilkes-Gibbs (1986) (Tangram Data) identify six types of noun phrases that introduce a referent into dialogue. Of these types, four do not include an interjection from the interlocutor. I do not provide an extensive analysis of this data, but it is worth noting that all four types appear in monologue as well. Several of these involve the speech

stream, and how the uttered phrase fits into separate intonation contours, or *tone groups*. Examples are below.

(1) **Elementary noun phrase:** The most basic form, usually stated in one tone group.

Tangram Data Example: *The guy leaning against the tree.*

Craft Data Example: *The yellow pipe-cleaner that's in a circle.*

(2) **Episodic noun phrase:** Uttered in two or more easily distinguished episodes or tone groups.

Tangram Data Example: *The goofy guy that's falling over... with his leg kicked up.*

Craft Data Example: *Your silver ribbon that's like a twisty S-shape... but it's in a straight line.*

(3) **Provisional noun phrase:** A noun phrase that is at first underspecified and immediately expanded without prompting.

Tangram Data Example: *The one that doesn't look like anything. It's kind of like the tree?*

Craft Data Example: *The smaller pom-pom. It's the uh the white one.*

(4) **Dummy noun phrase:** Stand-in phrase until a better phrase comes, e.g., *whatchamacallit.*

Tangram Data Example: Not found.

Craft Data Example: *A piece of uh I don't know.*

Also frequently reported for dialogue, underspecification and overspecification/redundancy are common. Speakers do not include enough information to uniquely distinguish the referent; they also include more information than is necessary to uniquely distinguish the referent. The extra information provided can either be completely *redundant* – one adjective rules out exactly the same set of distractor objects as another adjective – or

more generally *overspecified*, including an adjective that is made unnecessary for unique identification by another included adjective. Examples are listed in Table 3.

Label	Expression	Note	Reference Type
C15	*the pipe-cleaner*	Unclear which is meant.	U
D8	*the short silver ribbon*	The only ribbon that is short is silver; *short* is minimally distinguishing.	O
A11	*the yellow circular pipe-cleaner*	The only pipe-cleaner this is yellow is circular; *yellow* rules out the same set of pipe-cleaners that *circular* does; either *yellow* or *circular* would be minimally distinguishing.	R
A7	*a sparkly small pom-pom*	The only pom-poms that are sparkly are small; unclear which is meant	U, O

TABLE 3. Underspecification (U), Overspecification (O), and Redundancy (R): Examples from the Craft Corpus. Contrast items shown in Figure 3.

3.4.2. Object Dimensionality. In addition to properties that pick out referents, throughout the data we see reference to the objects' dimensions. This includes reference with SIZE modifiers, and reference to objects' parts, picking out pieces take up different locations within the whole object. This points to an underlying dimensional object representation that may be utilized during reference.

3.4.2.1. *Size Comparisons.* A total of 134 expressions (25.67% of all expressions) refer to size with a vague modifier (e.g., "big", "wide"). Only two references (0.38% of all expressions) use an estimate for a crisp measurement (e.g., "1 centimeter"), and both are produced by the same participant. Vague SIZE modifiers include comparative (e.g, "larger") and superlative (e.g., "largest") forms, which appear in 44 references, and base forms ("large") which appear in 88 references. Two references include both comparative/-superlative and base forms, "the smallest long ribbon" and "a small, the smaller of the

Part-whole modularity	Relative Size	Analogies
"a green pom-pom...	"a red foam-piece...	"a natural-looking piece
with the tinsel on the outside"	which is more square	of pipe-cleaner, it looks
"your gold twisty ribbon...	in shape rather than	a bit like a rope"
with sequins on it"	the longer rectangle"	"a pipe-cleaner that
"a wooden bead...	"the grey pipe-cleaner...	looks a bit like...
with a hole in the center"	which is the thicker one...	a fluffy caterpillar"
"one of the green pom-poms...	"the slightly larger one"	"the silver ribbon
with the sort of strands	"the smaller silver ribbon"	that's almost like
coming out from it."	"the short silver ribbon"	a big S shape."
"the silver ribbon... with the chainmail	"quite a fat rectangle"	"a... pipe-cleaner
detail down through the middle of it."	"thick grey pipe-cleaner"	that looks like tinsel."
11 References, 2.11%	**134 References, 25.67%**	**13 References, 2.49%**

TABLE 4. Frequency and examples for expressions with part-whole relations, size, and analogies.

five...". Further examples of SIZE language are given below, and the frequencies of each are listed in Table 5.

(3.1) Comparative: *"the bigger pom-pom"*

(3.2) Superlative: *"the green largest pom-pom"*

(3.3) Base: *"the large orange pipe-cleaner"*

Comparative/Superlative:	44	(32.35%; 8.43% of all expressions)
Base:	88	(65.44%; 17.05% of all expressions)
Both:	2	(1.47%; 0.38% of all expressions)
Crisp:	2	(1.47%; 0.38% of all expressions)
One or two axes:	37	(27.21%; 7.09% of all expressions)
Overall SIZE:	99	(72.79%; 18.97% of all expressions)

TABLE 5. SIZE modifier breakdown.

Of the references that mention SIZE, 99 (72.79% of the expressions with SIZE in them) use a vague modifier that refers to the overall SIZE of the object ("big", "small", "medium"). The 37 remaining references (27.21%) use a modifier that applies to one or two dimensions. This includes modifiers for *height*, the size along the y-axis ("the short silver

ribbon"), *width*, the size along the x-axis ("quite a fat rectangle"), and *width/depth*, including size along both the x- and z-axes ("the thick grey pipe-cleaner"). The distinction between these kinds of modifiers is further discussed in Chapter 4.

Current approaches to SIZE in REG utilize size measurements that are explicitly given (cf. van Deemter (2006)), or else require SIZE values to be predefined. In a visual domain, almost all participants produce vague SIZE modifiers without sizes or measurements explicitly given; with an input of a visual object presentation, their output includes SIZE modifiers. Such data suggests that natural reference in a visual domain utilizes processes comparing the height, width, and depth of a target object relative to other objects in the set. Indeed, several references in the data set include explicit comparison with the size of other objects:

(3.4) *"a red foam-piece... which is more square in shape rather than the longer rectangle"*

(3.5) *"the grey pipe-cleaner... which is the thicker one... of the selection"*

(3.6) *"the shorter of the two silver ribbons"*

(3.7) *"the longer one of the ribbons"*

(3.8) *"the longer of the two silver ribbons"*

In Example (3.4), height and width across two different objects are compared, distinguishing a square from a rectangle. In (3.5), "thicker" marks the referent as having a larger circumference than other items of the same type. (3.6) (3.7) and (3.8) compare the height of the target referent to the height of similar items.

The use of SIZE modifiers in a domain without specified measurements suggests that when people refer to an object in a visual domain, they are sensitive to its size and structure within a dimensional, real-world space. Without access to crisp measurements, people compare relative size across different objects, and this is reflected in the expressions they generate. These comparisons are not only limited to overall size, but include size in each dimension. This is compelling evidence that objects' three-dimensional structure,

particularly the size of each axis, should play a role in the construction of a referring expression in a visual domain, a finding also noted in work in psycholinguistics (cf. Landau and Jackendoff (1993)). I address how this can be implemented next, in Chapter 4.

3.4.2.2. *Part-Whole Modularity.* The role that a dimensional object understanding has within reference is further detailed by utterances that pick out the target object by mentioning an object part. 11 utterances (2.11%) in the data include an object part and its location within reference to the whole object. Half of the participants make reference to an object part at least once. Examples are given below.

(3.9) *"a green pom-pom, which is with the tinsel on the outside"*

(3.10) *"your gold twisty ribbon... with sequins on it"*

(3.11) *"a wooden bead... with a hole in the center"*

In (3.9), (3.10), and (3.11), parts of the objects are isolated from the whole object with their relative locations specified. This *part-whole modularity* (a term first introduced by Roy and Reiter (2005)) suggests that for an REG algorithm to generate these kinds of references, it must be provided with a model detailing the structure of each object, with the whole object as well as parts in, on, and around it represented. Part-whole modularity is (unfortunately) not explored further in this thesis.

3.4.3. Analogies. The data from this study also provide information on what can be expected from a knowledge base in an algorithm that aims to generate naturalistic reference. Reference is made 13 times (2.49%) to objects not on the board, where the intended referent is compared against something it is *like*. Examples are given below.

(3.12) *"a gold... pipe-cleaner... completely straight, like a ruler"*

(3.13) *"a natural-looking piece of pipe-cleaner, it looks a bit like a rope"*

(3.14) *"a pipe-cleaner that looks a bit like... a fluffy caterpillar..."*

In (3.12), a participant makes reference to a shape property of an object not on the board. In (3.13) and (3.14), participants refer to objects not on the board that typically share a variety of properties with the referent.

Reference to these other items do not pick out single objects, but types of objects (e.g., an object *type*, not *token*). They correspond to some typical idea of an object with properties similar to those of the referent. Rosch (1975) examined this tendency, introducing the idea of *prototype theory*, which proposes that there may be some central, *prototypical* notions of items that arises from stored typical properties for an object category (see Chapter 2 for further review).

A knowledge base with typical object properties could be utilized by an REG algorithm to compare the target referent's properties to typical properties of other objects. Such representations would help guide the generation of reference to items not in the scene, but similar to the target referent. I do not examine analogies further in this thesis, but I do work on the construction of a knowledge base of typical object properties in Chapters 5 and 8, and such a structure may be useful to build analogies in future work.

3.4.4. Speaker Variation. What is also clear from this study is that speakers are varied in the kinds of expressions they produce. It is worth noting that this variation is not normally distributed: There is a tendency to include color, and to include size when there is another object of the same type nearby. We therefore see that it is most common to describe an object by its type and color; this is followed by a preference to describe objects by their size, type and color. For the same referents, we also see underspecification, referring to just the object type; we also see overspecification, with, e.g., both color and size mentioned (see Table 3).

From a modeling perspective (Chapter 7), this speaks to the utility of a stochastic function that captures similar human tendencies. For example, a function that will give higher weight to color and size over material and texture, and so most frequently generating initial reference with color modifiers. I introduce such an approach in Chapter 7.

3.5. Implications: Distinguishing, Describing, and Reference

I have discussed several different aspects of reference in a study where referring expressions are elicited for objects in a real world visual scene. Reference in this domain appears to utilize object forms as they exist in a three-dimensional space and utilizes background knowledge to describe referents by analogy to items outside of the scene. This is undoubtedly not an exhaustive account of the phenomena at play in such a domain, but offers some initial conclusions that may be drawn from exploratory work of this kind.

Before continuing with the discussion, it is worthwhile to consider whether some of the data might be seen as going beyond reference. Perhaps the participants are doing something else, which could be called describing. Clark and Wilkes-Gibbs (1986) claim that on the first trial in their study, directors always *describe* the figure – using an indefinite description – while on the rest of the trials, they *refer* to it – using a definite description. How to draw the line between a distinguishing reference and a description, and whether such a line can be drawn at all, is an interesting question, and I address this in Chapter 2 Section 2.2.1. If the two are clearly distinct, then both are interesting to NLG research. If the two are one in the same, then this sheds some light on how REG algorithms should treat reference.

If what marks whether something is being described or being referred to is the type of determiner used ("the" or "a"), then it is noteworthy that frequency of each type is essentially equal in this data. Is half the data referring expressions and half the data descriptions? Or is one kind reference to an object token, while another reference to an object type? If the participant adds more information in the definite description, e.g., "the shorter ribbon with the sequins running down", does the expression move from being reference to being a description?

This experiment supports the idea that the difference between describing and distinguishing is not so clear in initial reference to visible objects, and may stand as theoretical endpoints on a continuum. *Referring*, then, may be seen as suggested by Clark and

Bangerter (2004), establishing (i) an individual as the referent; (ii) a conceptualization or perspective on that individual. Schematically, referring = indicating + describing; it distinguishes, it describes, may distinguish by describing, and whether the properties used are indicative of one or the other is not necessarily clear.

3.6. Towards an Algorithm

I now turn to a discussion of how the observed phenomena may be best represented in an REG algorithm. I propose that an algorithm capable of generating natural reference to objects in a visual scene should utilize:

(1) a spatial representation; a model of the object's dimensions and relative location of parts.
(2) a propositional representation; a model of non-spatial features such as COLOR and TEXTURE.
(3) a knowledge base of object typical object properties.

3.6.1. Spatial Knowledge. It is perhaps unsurprising to find reference that exhibits dimensional awareness in a study where objects are presented in three-dimensional space. Human behavior is anchored in space, and spatial information is essential for our ability to navigate the world we live in. However, referring expression generation algorithms geared towards spatial representations have largely oversimplified this tendency, keeping objects within the realm of two-dimensions, and leaving SIZE to the basic forms of *small* and *large*.

There has been some useful work done on spatial relations *between* objects. For example, Funakoshi (2004) and Gatt (2006) focus on how objects should be clustered together to form groups. Similarly, one of the strengths of the Graph-Based Algorithm (Krahmer et al., 2003) is its ability to generate expressions that involve relations between objects, and these include spatial ones ("next to", "on top of", etc.). In all these approaches, however, objects are essentially one-dimensional, represented as individual nodes.

Work that does look at the spatial information of different objects is provided by Kelleher et al. (2005). In this approach, the overall volume of each object is calculated to assign salience rankings, which then allows the Incremental Algorithm (Dale & Reiter, 1995) to produce otherwise underspecified reference. Dimensional properties of the referent are not used in constructing the referring expression, but one aspect of the object's three-dimensional shape (volume) affects the referring expression's final form (further work in this vein is detailed in Chapter 2). To my knowledge, the current work is the first to suggest that objects themselves should have a dimensional representation (x, y, and z axes) that guides how the reference is constructed.

This is supported by previous work that shows that we attend to the spatial properties of objects when we view them (Blaser, Pylyshyn, & Holcombe, 2000), and we have purely spatial attentional mechanisms operating alongside non-spatial, feature-based attentional mechanisms (Mishkin et al., 1983; Treue & Trujillo, 1999). These feature-based attentional mechanisms pick out properties commonly utilized in REG, such as TEXTURE, ORIENTATION, and COLOR. They also pick out edges and corners, contrast, and brightness. Spatial attentional mechanisms provide information about where the non-spatial features are located in relation to one another, SIZE, and the spatial interrelations between component parts.

Applying these ideas to this study, an REG algorithm that generates natural reference should utilize a visual, feature-based representation of objects alongside a structural, spatial representation of objects. A feature-based representation is already common to REG, and could be represented as a series of <ATTRIBUTE:value> pairs. A spatial representation is necessary to define how the object is situated within a dimensional space, providing information about the relative distances between object components, edges, and corners.

With such information provided by a spatial representation, the generation of part-whole expressions, such as "the pom-pom with the tinsel on the outside", may be possible. This

also allows for the generation of SIZE modifiers ("big", "small"), for example, by comparing the difference in overall height of the target object with other objects in the scene, or against a stored prototype (discussed below). Relative SIZE comparisons across different dimensions could also be made, used to generate SIZE modifiers such as "wide" and "thick" that refer to one or two dimensional axes. We propose a model for the generation of SIZE modifiers in the next chapter, Chapter 4.

3.6.2. Propositional Knowledge. Referring expression generation algorithms commonly utilize an input that lists properties within the scene. In the Incremental Algorithm, the properties are available in a 'preference order' list. In the Graph-Based Algorithm (Krahmer et al., 2003), they are associated to weights. I find that assuming objects have properties that are available for use during generation fits well with the data; object properties are mentioned explicitly in all of the elicited reference. I can account for the majority of the data by proposing that properties of objects in the scene are utilized to form a referring expression, e.g., that properties of objects in the scene may serve as input to a referring expression generation algorithm. I examine this in further detail in Chapters 7 and 8, proposing a model that builds properties from visual input and then uses these to refer to a target object.

However, there is not always a clear distinction between TYPE and other sorts of properties, such as SHAPE and MATERIAL. This runs counter to the usual assumptions of referring expression generation, where TYPE is given a special treatment separate from other kinds of properties.

For example, one participant mentions "one little green foam heart" while another mentions a "small green heart foam-piece", referring to the same object. In order to account for such variation in the Incremental Algorithm and other models of referring expression generation, these two expressions must be generated from two different sets of properties. One set, to generate "one little green foam heart", would have TYPE corresponding to 'heart' and MATERIAL corresponding to 'foam'. The other, to generate the "small

green heart foam-piece", would have TYPE corresponding to 'foam-piece' and SHAPE corresponding to 'heart'. Such representations would miss the clear generalization that the two expressions are naming the same properties of the same object, and just have different surface forms.

This may be seen as a user model issue. Different subjects have different domain models of the craft domain (which may partially reflect differences in expertise), and this will affect (amongst other things) what the possible values for TYPE are in the referring expressions they produce. SHAPE and MATERIAL, for example, may be realized as a head noun to name an object depending on how familiar the subject is with the object.

3.6.3. Typicality and Analogies. When we use analogies, as in "the pipe-cleaner that looks like a caterpillar", we use world knowledge about items that are not themselves visible. Such an expression draws on similarity that does not link the referent with a particular object, but with a general type of object: the pipe-cleaner is caterpillar-*like*.

To generate these kinds of expressions, an REG algorithm would first need a knowledge base listing typical values of attributes. For example, a banana object might have a typical COLOR of yellow. With typical object properties in the knowledge base, the REG algorithm would need to calculate similarity of a target referent to other known items. This would allow a piece of yellow cloth, for example, to be described as being the color of a banana.

Implementing such similarity measures in an REG algorithm is challenging. One difficulty is that typicality values may be different depending on what is known about an item; a typical unripe banana may be green, or a typical rotten banana brown. Another difficulty will be in determining when a referent is similar *enough* to a stored object to warrant an analogy. Additional research is needed to explore how these properties can be reasoned about. I further explore what affect typicality has on reference in Chapter 5, and develop a knowledge base representation of typical properties in Chapter 8.

Approaching REG in this way follows work in psycholinguistics that posits the existence of some kind of object typicality structure in people's mental representations of objects (Rosch et al., 1976; Wu & Barsalou, 2009, also see Chapter 2 Section 2.2.2), and work in cognitive science and neurophysiology that suggests that expectations about objects' visual and spatial characteristics are derived from stored representations of typical object features in the inferior temporal lobe of the brain (Logothetis & Sheinberg, 1996; Riesenhuber & Poggio, 2000; Palmeri & Gauthier, 2004). Most formal theories of object perception posit some sort of *category activation system* (Kosslyn, 1994), a system that matches input properties of objects to those of stored typical objects, which then helps guide expectations about objects in a top-down fashion.[3] This appears to be a neurological correlate of the knowledge base I propose to underlie analogies.

Such a system contains information about typical object component parts and where they are placed relative to one another, as well as relevant values for MATERIAL, COLOR, etc. This suggests that the spatial and non-spatial feature-based representations proposed for visible objects could be used to represent typical object representations as well. Indeed, how we view and refer to objects appears to be influenced by the interaction of these structures: Expectations about an object's spatial properties guide our attention towards expected object parts (Mackworth & Morandi, 1967) and non-spatial, feature-based properties throughout the scene (Kosslyn, 1994; Itti & Koch, 2001). This affects the kinds of things we are most likely to generate language about (Itti & Arbib, 2005).

3.6.4. Further Implications. Most implemented algorithms focus on *unique identification* of a referent, determining the set of properties that distinguish a particular target object from the other objects in the scene (the contrast set). This view of reference was first outlined by Olson (1970), "the specification of an intended referent relative to a set of alternatives". A substantial body of evidence now shows that contrastive value relative to a set of alternatives is not the only factor motivating speakers' choice

[3]Note that this is not the only proposed matching structure in the brain – an *exemplar activation system* matches input to stored exemplars.

of properties in reference. The phenomena of *overspecification* and *redundancy*, where speakers select properties which have little or no contrastive value and confirmed in this study, was observed in early developmental studies in visual domains (Ford & Olson, 1975; Whitehurst, 1976; Sonnenschein, 1985), later studies on adult speakers in visual domains (Pechmann, 1989; Engelhardt, Bailey, & Ferreira, 2006; Koolen et al., 2011), and confirmed in this study as well. The related phenomenon of *underspecification*, where speakers select a set of properties that do not linguistically specify the referent, has also received some attention, particularly in visual domains (H. H. Clark et al., 1983; Kelleher et al., 2005; Viethen et al., 2008, see Chapter 2 Section 2.2.2.2 for a further discussion).

In this exploratory work, I have found that rather than uniquely identify objects using a set of contrasting properties, participants tended to include detailed information about the objects' parts and analogies to other things that the object is most like. Along with the phenomena of over- and underspecification, this suggests that people select distinguishing properties not just as a function of their discriminatory power, but by how visually salient they are within the scene, how common each property is to mention, and how each property compares to stored representations of similar objects. Additionally, there are a variety of expressions that people produce, with tendencies for some expressions over others. I introduce a new algorithm that takes many of these issues into account in Chapter 7.

I have also found support for a knowledge base used during reference that contains typicality profiles of objects. This opens up the possibility of generating many other kinds of natural references; in particular, such knowledge would allow the algorithm to compute which properties a given kind of referent may be expected to have and which properties may be unexpected. Unexpected properties may therefore stand out as particularly salient. For example, a dog missing a leg may be described as a "three-legged dog" because the typical dog has four legs.

I believe that this perspective, which hinges on the unexpectedness of a property, suggests a new approach to attribute selection. Unlike the Incremental Algorithm, the order in which attributes are examined would not be fixed, but would depend in part on the nature of the referent and what is known about it. I further explore the role of typicality in Chapter 5, examining the properties of MATERIAL and SHAPE. These findings are implemented in the algorithm in Chapter 7.

3.7. Conclusions and Future Work

I have explored the interaction between viewing objects in a three-dimensional, spatial domain and referring expression generation. This interaction has shed light on structures that may be useful in connecting vision in the real world to naturalistic reference. The proposed structures include a spatial representation, a propositional representation, and a knowledge base with representations for typical object properties. Using structures that define the propositional and spatial content of objects fits well with work in psycholinguistics, cognitive science and neurophysiology, and may provide the basis to generate a variety of natural-sounding references from a system that recognizes objects.

One interesting issue I did not explore here is the issue of dual contrast sets. In the study discussed in this chapter, one contrast set is the group of craft items in front of the speaker, and the other contrast set if the group of craft items making up the face. As items in one go down, items in the other go up. The question that comes up about this is whether people will distinguish in terms of the items in the craft face, or in terms of all the craft items available to them. This would make the difference between saying, for example, "Place all the pink fluff balls" and "Place 6 pink fluff balls". I did not look at this effect in great detail, and hope to examine this further in future work.

It is important to note that any naturalistic experimental design limits the kinds of conclusions that can be drawn about reference. A study that elicits reference to objects in a visual scene provides insight into reference to objects in a visual scene; these conclusions

cannot easily be extended to reference to other kinds of phenomena, such as reference to people in a novel. I therefore make no claims about the broader task of reference in this chapter; generalizations from this research can provide hypotheses for further testing in different modalities and with different sorts of referents.

What is clear from the data is that both a spatial understanding and a non-spatial feature-based understanding appear to play a role in reference to objects in a visual scene, and further, reference in such a setting is bolstered by a knowledge base with stored typical object representations. Utilizing structures representative of these phenomena, we may be able to extend object recognition research into object reference research, generating natural-sounding reference in everyday settings.

In the next chapter, I focus on the dimensional representation of objects, building a computational model for the second most common property in this study and one of the primary properties in visual perception: SIZE.

Chapter 4

Size

4.1. Introduction

The exploratory experiment in the last chapter discussed the predominance of SIZE in describing objects. Evidence from other visual description tasks further suggests that SIZE is a salient property, particularly when objects of the same type are in the scene (Brown-Schmidt & Tanenhaus, 2006; Sedivy, 2003). This is further supported by tagging visually descriptive text (Table 1), where we find top words denoting size and color, with *little*, *long*, and *large* appearing in the thirty most common adjectives.[1]

If size-denoting adjectives play a key role in describing the visual world, then we can shed some light on how to generate visual descriptions by examining how size features map to size language. By using features that characterize an object's size – its height and width, for example – we can begin to predict the kind of size words that speakers are likely to use for different objects.

As a first-pass analysis of the problem, consider the pictures in Figure 1 below.

In the pictures of the mice, the mouse in the middle may be *large* in the first picture, but *thin* in the second picture. In the pictures of the tables, table A is taller and wider than table B, so it is true that A is *taller* than B; it is true that A is *wider* than B; it is also true that A is *bigger* than B. All three words may be appropriate to refer to A, but may mean something very different and reflect different properties of a referent.

[1]The fact that size-denoting adjectives are so prevalent in these corpora is likely due to the fact that size words are used both visually and non-visually: Lacking a way to automatically distinguish between the two is further reason to understand what characterizes visually descriptive language. It is for this reason that I use visually descriptive text and not, e.g., the BNC to get a sense of the size adjectives that are used.

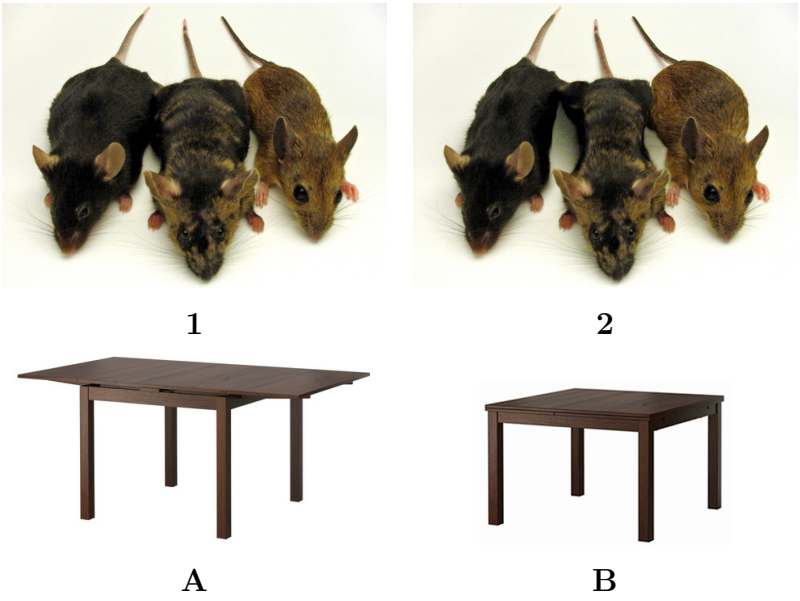


FIGURE 1. Size variations: What are the appropriate lexical forms?

WORD	COUNT	FREQ.	WORD	COUNT	FREQ.
little	1213	0.0292	much	309	0.0074
old	979	0.0236	new	287	0.0069
other	762	0.0183	whole	273	0.0066
more	662	0.0159	few	270	0.0065
good	637	0.0153	**large**	269	0.0065
great	588	0.0142	next	253	0.0061
last	517	0.0124	sure	238	0.0057
such	473	0.0114	better	226	0.0054
own	438	0.0105	**white**	211	0.0051
same	429	0.0103	**black**	210	0.0051
young	424	0.0102	high	203	0.0049
first	393	0.0095	full	181	0.0044
many	379	0.0091	dead	180	0.0043
long	372	0.0090	least	176	0.0042
poor	351	0.0084	dark	175	0.0042

TABLE 1. Top 30 adjectives: Andersen's *Fairy Tales*, Brönte's *Wuthering Heights*, E.T.A. Hoffman's *Devil's Elixir*, Mark Twain's *Life on the Mississippi*, Lewis Carroll's *Through the Looking Glass*, Sheridan le Fanu's *Uncle Silas*. Extracted using the Stanford part-of-speech tagger.

In the following sections, I describe three studies examining the usage of SIZE modifiers. The term "modifier" is often used rather than "adjective" to refer to the variety of surface forms in which SIZE may be realized: as an adjective ("the short box"), relative clause ("that is shorter"), or prepositional phrase ("with less height").

In the first study on SIZE, I seek to better understand the relationship between an object's dimensions and the words used to identify it. I conduct an experiment to elicit size-denoting modifiers from images of real world objects, and evaluate three hypotheses that explore this relationship. These hypotheses test the interacting *height* and *width* features that are involved in the selection of SIZE modifiers, and illustrate when preferences for OVERALL SIZE modifiers ("big", "small") versus INDIVIDUATING SIZE modifiers ("tall", "thin") emerge in different contexts. Additionally, I am able to confirm the Hermann and Deutsch (1976) findings on SIZE preferences, and further build on these results.

In Study 2 (Section 4.4), I expand the first study to an additional 414 participants, and examine how well a machine-learning approach does at predicting among three basic SIZE types. Taking the findings from Study 1 as a starting point, I develop an end-to-end connection from visual features to size language. This incorporates a visual front-end as input to the size classification task, using an image processing technique called SIOX (Friedland et al., 2005). I test whether real world measurements are better predictors of size language than pixel-based (image) measurements, and find that they are.

Study 3 (Section 4.5) further builds on the prior two studies, predicting among six more fine-grained SIZE types. In this study, I compare a full hand-coded SIZE generation algorithm to a decision tree-based binary classification task that predicts the inclusion of each SIZE type, and find that the two approaches perform comparably, predicting well over the majority of SIZE language used by participants. Remarkably, the SIZE generation approaches work even better when tested in a new domain, and I discuss how this work folds into a larger visual description generation algorithm.

The first two studies on SIZE examine how SIZE modifiers are used with just a *single* comparator object in the scene beside the target. In the third study, I apply the approaches developed on this data to a corpus with *several* comparator objects, illustrating its applicability across domains and in more complex scenes. The approaches are easily adapted to handle several comparators rather than just one by using the average height and width of all objects of the same type as the target referent.

Throughout, I will use the terms *h* and *height* to refer to an object's *y*-axis in a three-dimensional space, and *w* and *width* to refer to an object's *x*-axis in three dimensional space.

4.2. Background

4.2.1. Size Research. To my knowledge, a thorough analysis on the use of different size adjectives to refer to an object's size is not available prior to this work. However, previous research does suggest the kinds of features that may influence the selection of SIZE modifier. Landau and Jackendoff (1993) point out that a modifier like "big" selects different dimensions depending on the nature of the object, and tends to be used in cases where an object is large in either two or all three of its dimensions, while modifiers like "thick" and "thin" may be applied when an object extends in a single dimension (see Chapter 2 Section 2.2.2). Hermann and Deutsch (1976) show that when people are presented with an object with two axes of different sizes than a comparator's, they are more likely to refer to the axis with the larger difference. Roy (2002) finds that words like "small" and "large" cluster together, but that "tall" is placed in a separate cluster.

There has been considerable research on the behavior of SIZE modifiers for other purposes, such as the semantics of dimensional modifiers (Bierwisch & Lang, 1989; Eilers, Oller, & Ellington, 1974; Tucker, 1998; Morzycki, 2009) and the acquisition of the meaning of such modifiers (Bartlett, 1976). We also know roughly how to choose between different forms of a size adjective ("larger", "largest") (van Deemter, 2006). A primary

open question this research leaves is whether people distinguish objects by focusing on one single dimension or by combining dimensions, and how these are realized as surface forms. Given information about an object's height and width, it is unclear how it will be referred to.

Utilizing images of real objects to predict the SIZE modifier types used in reference supplies a non-trivial computer vision input while following work in developing computational models that bridge the symbolic realm of language with the physical realm of real world referents (Roy & Reiter, 2005; Tanenhaus et al., 1995). Approaching the task in this way provides detailed information about which visual size features may affect the form of a referring expression, and I discuss the implications of the findings for research in referring expression generation.

Most REG algorithms presuppose that referents are individuated using absolute properties, whose applicability to a referent does not depend on the context in which the referent appears. They therefore do not provide mechanisms for reasoning about how a property may involve interacting features, such as the interaction of an object's height with its width. In Dale and Reiter (1995) and Krahmer et al. (2003), the knowledge base must mark elements as `large` or `small`. Van Deemter (2000, 2004) modifies this procedure by storing actual sizes (e.g., in centimeters) in the knowledge base, making the decision of whether something is larger or smaller context dependent. More fine-grained SIZE modifiers are presumably considered lexical decisions, made by a later module that translates properties into words.

The problem with these proposals is that they do not do justice to the fact that SIZE can involve a *combination* of dimensions; a turtle may be fat, or big, but seldom tall.

This property of reference is not only important for work in referring expression generation that uses SIZE (Kelleher et al., 2005; van Deemter, 2006; Viethen & Dale, 2008), but it offers a clear link between language generation and computer vision techniques that provide detailed information about an object's physical dimensions (Friedland et

al., 2005; Zheng, Yuille, & Tu, 2010). Systematically manipulating the visual feature of size to develop an account of how SIZE is used in reference furthers the goal of developing a grounded semantic core for natural language (Gorniak & Roy, 2004), tying visual perception to linguistic reference.

4.2.2. Machine Learning and Object Description. This exploration into generating SIZE modifiers includes a machine learning component. Although there has not been previous work on machine learning specifically for the generation of SIZE modifiers, there has been previous work on machine learning for broader object descriptions. I therefore provide some background for these approaches.

Previous work on determining the form of an object description using machine learning has created models that predict a wide range of properties, such as the inclusion of COLOR, LOCATION, etc., as well as the overall form of the noun phrase (e.g., personal pronoun, definite description). These approaches utilize a variety of contextual features, such as intentional influences and conceptual pact features (Jordan & Walker, 2005) and syntactic, semantic, and discourse features (Poesio, 2000).

A clear area where machine learning may be useful in building REG models is in predicting different references for different speakers, incorporating the observation that to generate natural reference, one must account for speaker variation (Reiter & Sripada, 2002). In light of this, recent work in REG has begun to use speaker-specific constraints in order to improve the performance of reference algorithms (Fabbrizio, Stent, & Bangalore, 2008). In work most closely related to this work, Viethen and Dale (2010) use a decision tree classifier to predict the set of attributes different speakers will use to refer to geometric shapes. The results are mixed, largely due to the lack of data for many of the proposed classes; however, there is a significant increase in accuracy when speaker identity is included as a model feature.

It is important to note that at both ends of this connection, the problem is reduced to basic levels. The visual input of images is an obvious application for computer vision

that utilizes object recognition. However, object recognition can only return regions of an image where an object is likely to exist, not the specific details of the object's dimensions (Walther, Itti, Riesenhuber, Poggio, & Koch, 2002; Lowe, 2004, see Chapter 2 Section 2.4). To reason about an object's shape, an object segmentation approach is needed, with the general location of the object already specified. Work linking object recognition to object segmentation is still quite new (e.g., Zheng et al., 2010). I therefore compare real world measurements to measurements extracted from semi-supervised object segmentation.

At the other end of the vision-language connection is REG, a well-developed subfield within natural language generation. However, as discussed in Chapter 2 and the Introduction, REG has focused on categorizing which subset of scene attributes may be selected to identify an object. In this study, I take a more fine-grained approach by exploring the use of a single attribute – SIZE – and several of its possible forms. I hope that this research provides a basic foundation from which to raise the complexity at both ends.

I therefore set out to examine how the words proposed to refer to specific axes, like "tall" and "thick", are used differently than words proposed to refer to overall size, like "large" and "small". The first type I will call INDIVIDUATING SIZE modifiers and the second OVERALL SIZE modifiers.[2]

Over the course these three studies, I explored the visual features that can be used to determine size and introduced two approaches to SIZE modifier generation. These are developed for the microplanning stage of a natural language generation system (Reiter & Dale, 2000, see Chapter 1 Section 1.2), generating a SIZE type that directly informs lexical choice and surface realization of a final string.

[2]Note that INDIVIDUATING SIZE modifiers may occasionally pick out more than one axis, e.g., as in the word "thick".

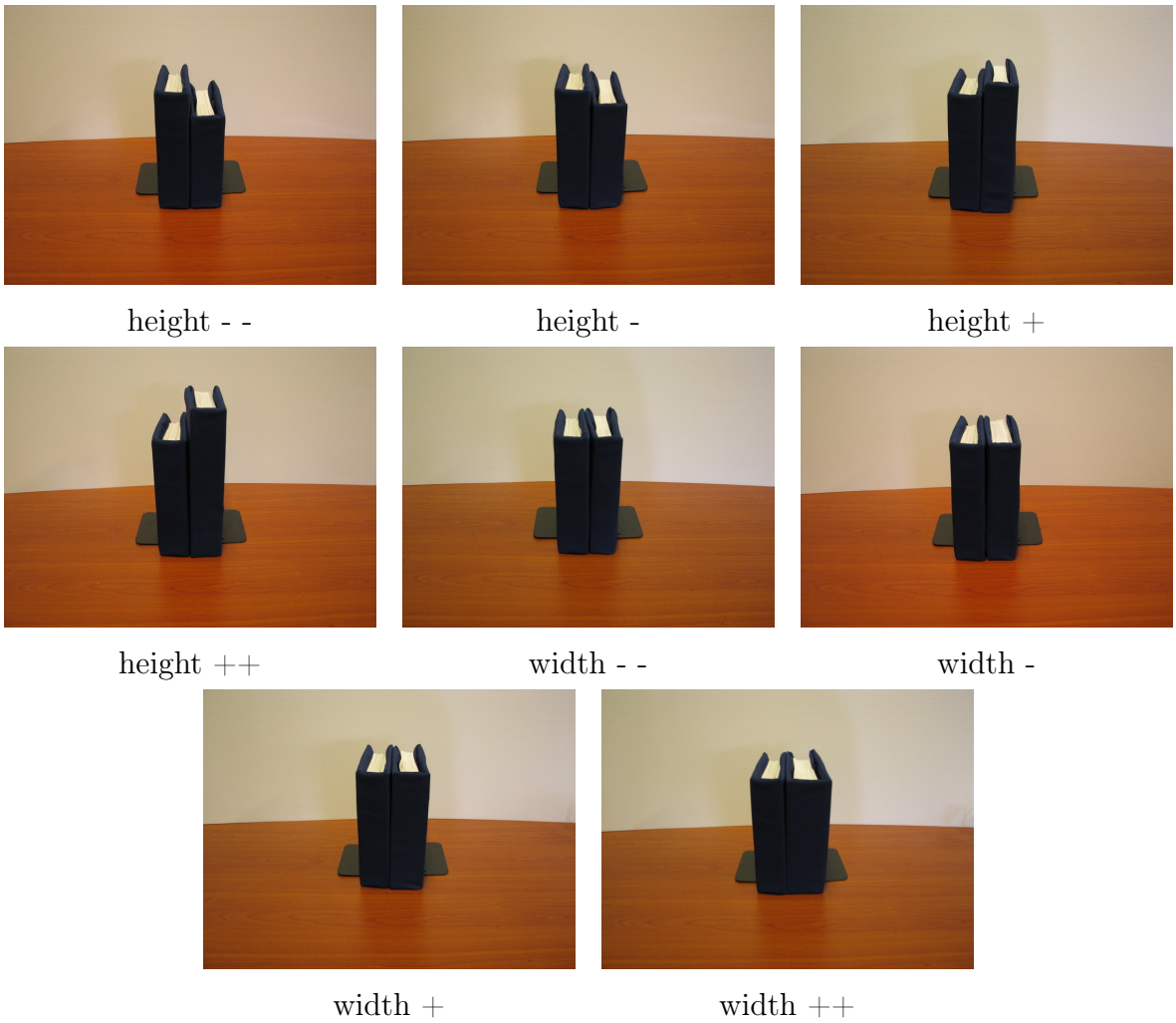


FIGURE 2. Examples of all books stimuli differing on one dimension. Target referent is on the right, and is a lot shorter (height - -), a little wider (width +), etc.

.

4.3. Study 1

In the first study on SIZE, I elicited size-denoting language to images of real world objects. The initial hypotheses were designed to formalize aspects of reference to size that have been implied by earlier work (e.g., Landau & Jackendoff, 1993), but have not yet been systematically tested. This provides a basis from which to design an REG algorithm that refers to an object's size.

I categorize the size language that people use as being either INDIVIDUATING (words like "tall" and "thin"), OVERALL (words like "big" and "small"), or NEITHER (no SIZE modifier), and find that the selection of these modifier types is predictable. I also find strong evidence that the selection of each is brought about by several interacting factors in this domain, including how a target object's physical dimensions differ from another object of the same type, and the relationship between the target object's individual dimensions. Findings from this study are used to inform the initial design of a hand-coded algorithm capable of referring to objects naturally, providing a further link between visual cues and corresponding linguistic forms.

4.3.1. Experiments. The experiments in this study are designed to examine what happens when a referent object is different in size from a comparator object (1) along a single axis; (2) along two axes, in the same direction (both axes larger or both smaller); and (3) along two axes, in opposite directions (one axis larger, one smaller). An example of test stimuli with one set of objects differing along a single axis (the height or y axis and the width, or x axis) are shown in Figure 2. Hypotheses are listed below.

H_1 When a single dimension differs between a referent object and another object of the same type, an INDIVIDUATING SIZE modifier will be produced more often than an OVERALL SIZE modifier.

H_2 When two dimensions differ in the same direction between a referent object and another object of the same type, an OVERALL SIZE modifier will be produced more often than an INDIVIDUATING SIZE modifier.

H_3 When two dimensions differ in opposite directions between a referent object and another object of the same type, an INDIVIDUATING SIZE modifier will be produced more often than an OVERALL SIZE modifier.

It is relatively straightforward to write a deterministic algorithm capturing what we hypothesize people will tend to do when to compare dimensions between two similar objects, and I sketch such an algorithm in Figure 3. Note that some aspects are still

left unspecified, and the algorithm does not address how large a difference must be in order to be salient – clearly, some differences between referent and comparator may be too small to elicit a corresponding modifier. This is an area for future work.

Lines 3 and 6 represent H_2, returning an OVERALL SIZE modifier depending on the differences between dimensions. Lines 4 and 7 roughly represent H_3, and call to a second function motivated by Hermann and Deutsch (1976), LARGESTDIMDIFF, which returns the dimension with the greater difference (if they are equal, the algorithm can randomly select one axis). Lines 5, 8, 9, and 10 represent H_1. The final SIZE modifier structure an OVERALL SIZE modifier (<'over'>), or an INDIVIDUATING SIZE modifier picking out a specific axis (<'ind', 'x'> or <'ind', 'y'>), along with whether the modifier should capture a larger (1) or smaller (0) difference. Thus, for example, (<'over'>, 1) could be realized as "large" or "big", while (<'ind', 'y'>, 0) could be realized as "short".

Fitting this within the larger framework of referring expression generation, this algorithm decides (1) whether or not to include a SIZE modifier; and (2) the general semantic type of the SIZE modifier. For (1), if line 11 is true, the algorithm returns (None, None), showing there is no size difference and no SIZE modifier will be used. In this way, the SIZE algorithm may always called during REG given the heights and widths of objects in the scene, regardless of the discriminatory power of size; there's either a SIZE modifier type returned or no SIZE modifier type returned.

However, I expect that what this algorithm captures is not the whole story, and return to this issue in Section 4.3.5.

The stimuli in this study were photographs of real world objects, and the objects were physically cut and shaped into different sizes.

4.3.2. Method.

4.3.2.1. *Participants.* 95 subjects collected using Amazon's Mechanical Turk (Amazon, 2011) were paid for their participation. 87 of these participants labeled themselves as

Input: Referent height, width (ry, rx)
Average height, width for comparators of referent's type (dy, dx).
Output: SIZE modifier type.

```
SIZEMOD(rx, ry, dx, dy):
 1. axes = <rx, ry, dx, ry>
 2. case (mod, pol) of:
 3.     ry > dy and rx > dx:     (<'over'>, 1)
 4.     ry > dy and rx < dx:     LargestDimDiff(axes)
 5.     ry > dy and rx == dx:    (<'ind', 'y'>, 1)
 6.     ry < dy and rx < dx:     (<'over'>, 0)
 7.     ry < dy and rx > dx:     LargestDimDiff(axes)
 8.     ry < dy and rx == dx:    (<'ind', 'y'>, 0)
 9.     ry == dy and rx > dx:    (<'ind', 'x'>, 1)
10.     ry == dy and rx < dx:    (<'ind', 'x'>, 0)
11.     ry == dy and rx == dx: (None, None)
12.  return (mod, pol)

LARGESTDIMDIFF(<rx, ry, dx, dy>):
  axis = axis with largest difference between r and d (x or y)
  pol = direction of difference (1 or 0)
  return (<'ind', axis>, pol)
```

FIGURE 3. Initial algorithm for generating SIZE modifiers. 1 is used to designate a positive polarity (+) and 0 a negative polarity (-). 'ind' represents *individuating* modifiers, and 'over' represents *overall* modifiers.

"Native" or "Fluent". From this set, I randomly chose a subset of 60 total participants, spread evenly as groups of 20 in each of the three experiments.

4.3.2.2. *Materials.* Several different objects were used to elicit SIZE modifiers. These objects were sponges, boards, books, and brownies. All objects were rectilinear solids, varied along their height and width dimensions. The objects were intermixed with fillers, discussed in further detail below.

Each object appeared to the right of a comparator object of the same type (see Figure 4). The target object could appear in 24 different sizes, systematically varied along height and width axes: larger (++, axis 5/4 size of comparator), a little larger (+, axis 11/10

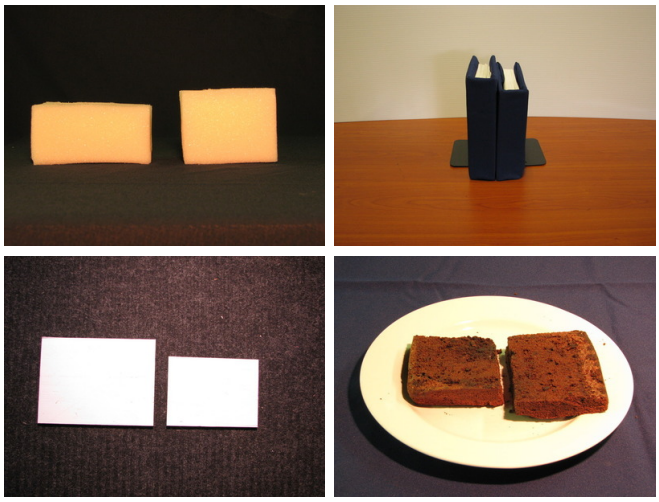


FIGURE 4. Example stimuli: sponges (h++/w- -), books (h-/w0), boards (h- -/w- -), and brownies (h++/w0). Object sizes are annotated as height/width combinations (see Table 2).

Object	**Height**					**Width**				
	++	+	0	-	- -	++	+	0	-	- -
brownies	11.25	9.90	9.00	8.18	7.20	11.25	9.90	9.00	8.18	7.20
sponges	6.25	5.50	5.00	4.54	4.00	12.50	11.00	10.00	9.09	8.00
books	25.00	22.00	20.00	18.18	16.00	6.25	5.50	5.00	4.55	4.00
boards	19.05	16.76	15.24	13.84	12.19	25.4	22.35	20.32	18.47	16.26

TABLE 2. Measurements for objects along each axis (in cm). Object sizes are annotated as height/width combinations, e.g., h++/w++.

size of comparator), no difference (0, axis same size as comparator), a little smaller (-, axis 10/11 size of comparator) and smaller (- -, axis 4/5 size of comparator). Values for these measurements are provided in Table 2. In the rest of this this chapter, these annotations are used to denote height/width combinations for objects. For example, a sponge that is h++/w0 has a height of 6.25cm and a width of 10cm, while a book that is h++/w0 has a height of 25cm and a width of 5cm. A total of 96 images were used for this study, split among three experimental groups, one for each hypothesis.

4.3.2.3. *Design.* I conducted three experiments, addressing each of the hypotheses. The design for each was dimension (2: height, width) x degree of difference (2: small, large) x direction of difference (2: bigger, smaller).

EXPERIMENT 1: DIFFERENCES OF DEGREE, SINGLE DIMENSION. Responses were elicited for objects with height/width combinations of h++/w0, h0/w++, h+/w0, h0/w+, h-/w0, h0/w-, h- -/w0 and h0/w- - (8 conditions). Each target item differed from its comparator item in one dimension.

EXPERIMENT 2: DIFFERENCES OF DEGREE, MATCHING ACROSS DIMENSIONS. Responses were elicited for objects with height/width combinations of h++/w++, h++/w+, h+/w++, h+/w+, h- -/w- -, h- -/w-, h-/w- - and h-/w- (8 conditions). Each target item differed from its comparator item in two dimensions and in the same direction for each; the target item was either bigger overall or smaller overall than the comparator.

EXPERIMENT 3: DIFFERENCES OF DEGREE, DIFFERENT POLARITIES ACROSS DIMENSIONS. Responses were elicited for objects with height/width combinations of h++/w- -, h- -/w++, h++/w-, h-/w++, h+/w- -, h- -/w+, h+/w- and h-/w+ (8 conditions). Each target item differed from its comparator item in two dimensions and in the opposite direction for each; the target item had one axis bigger and one axis smaller than the comparator.

For each experiment, I followed a Latin square design where all participants saw each of the four object types, with two examples per condition (for example, both sponges and brownies for the h++/w- - condition). This yielded 16 experimental stimuli per participant. Each experiment had two subgroups, where one half (10 participants) saw 2 stimuli per condition, and the other half (10 participants) saw the other 2 stimuli per condition.

Stimuli in each experiment were intermixed with the 24 filler pictures, consisting of spatulas, Legos, and shoes. Spatulas appeared in groups of three and Legos and shoes appeared as sets of two. Most objects in filler conditions could be distinguished using part-whole phrases, e.g., “the one with the red Lego” or “the shoe with the laces untied”. In total, each subject provided responses for 40 object pictures. Each picture was 400 pixels wide x 300 pixels high, and could be enlarged to 700 x 525 by clicking on it. Pictures

were presented in random order, and experimental groups were assigned randomly. The beginning of an example Mechanical Turk stimulus set seen by a participant is shown in Appendix C.

4.3.2.4. *Procedure.* Instructions informed participants that they had been chosen as "the thrower", tossing objects down a tube to a person below, and their goal was to clearly identify the object on the right so that the person below could pick it up.

Responses were manually corrected for spelling and normalized for punctuation and capitalization. For each expression, I annotate the modifiers as being INDIVIDUATING (I) – words like "tall" and "thin" – OVERALL (O) – words like "big" and "small" – or NEITHER (N).

Each INDIVIDUATING modifier was annotated by three postgraduates as being a height modifier or a width modifier. I use the annotations from the annotator who had the highest agreement with the other two, with a Cohen's kappa of 0.90 (95% CI, 0.87–0.94) and 0.71 (0.66–0.76). Table 3 lists the vocabulary and modifier types based on this data. Most base modifiers have corresponding comparative (ending in -er) and superlative (ending in -est) forms.

INDIVIDUATING	height:	high long narrow short skinny slender squat tall thick thin
	width:	fat lengthy long narrow skinny slim thick thin wide
OVERALL		big large small

TABLE 3. Size vocabulary.

4.3.3. Results. Results are based on the 320 responses for each experiment. Each response to the test stimuli is counted as either including or not including an INDIVIDUATING SIZE modifier (0 or 1) and including or not including an OVERALL SIZE modifier (0 or 1). Note that the two are not exclusive. For each participant, I sum the total number of responses with each type of modifier. This provides two sets for a two-tailed paired t-test in each of the analyses.

Object, Cond.	Expression	Modifier Types
sponges, h+/w- -	taller sponge	INDIVIDUATING
boards, h- -/w++	the shorter and slightly wider board with a diagonal top side	INDIVIDUATING
boards, h- -/w- -	smaller board	OVERALL
brownies, h+/w-	the most square brownie	NEITHER

TABLE 4. Example expressions for different <object, condition> stimuli. Conditions are composed of different measurements of the height (h) and width (w) axes.

Experiment	INDIVIDUATING	OVERALL	BOTH	NEITHER
1	160 (50.0%)	114 (35.6%)	8 (2.5%)	38 (11.9%)
2	93 (29.1%)	211 (65.9%)	15 (4.7%)	1 (0.3%)
3	226 (70.6%)	28 (8.8%)	20 (6.3%)	46 (14.4%)

TABLE 5. Count and proportion (in parentheses) of responses including either 1+ INDIVIDUATING SIZE modifiers, 1+ OVERALL SIZE modifiers, BOTH, or NEITHER. Statistics below are not based on these raw numbers, but on total number of responses *per participant* that include INDIVIDUATING or OVERALL.

Examples of normalized responses along with corresponding modifier types are given in Table 4. Table 5 provides the counts and proportions of responses that included an INDIVIDUATING SIZE modifier, an OVERALL SIZE modifier, BOTH, or NEITHER for each experiment.

H_1: When a single dimension differs between a referent object and another object of the same type, an INDIVIDUATING SIZE modifier will be produced more often than an OVERALL SIZE modifier.

We do not see a strong trend to include INDIVIDUATING SIZE modifiers, with such modifiers occurring in an average of 8.4 responses per participant (168 responses total; 160 with only INDIVIDUATING, 8 with BOTH), compared to an average of 6.1 responses per participant (122 responses total; 114 with only OVERALL, 8 with BOTH) containing an OVERALL SIZE modifier. The difference is not significant ($t = 1.382$, $df = 19$, $p = 0.183$).[3]

[3]In Study 2, with a larger set of participants, this trend is significant.

H_2: When two dimensions differ in the same direction between a referent object and another object of the same type, an OVERALL SIZE modifier will be produced more often than an INDIVIDUATING SIZE modifier.

We find a strong trend to include OVERALL SIZE modifiers, with such modifiers occurring in an average of 11.3 responses per participant (226 expressions total; 211 with only OVERALL, 15 with BOTH). INDIVIDUATING SIZE modifiers occur in an average of 5.4 responses (108 expressions; 93 with only INDIVIDUATING, 15 with BOTH). The difference in this distribution is significant ($t = -4.914$, $df = 19$, $p < .001$).

H_3: When two dimensions differ in opposite directions between a referent object and another object of the same type, an INDIVIDUATING SIZE modifier will be produced more often than an OVERALL SIZE modifier.

We find that when two dimensions differ in opposite directions, INDIVIDUATING SIZE modifiers are chosen in an average of 12.3 responses per participant (246 expressions total; 226 with only INDIVIDUATING, 20 with BOTH), while OVERALL SIZE modifiers are chosen in an average of 2.4 responses (48 expressions total; 28 with only OVERALL, 20 with BOTH). The difference in this distribution is significant ($t = 8.866$, $df = 19$, $p < .001$).

Based on these results, we can confirm Hypotheses 2 and 3. Overall SIZE modifiers tend to be used when both axes are different from a comparator in the same direction, and INDIVIDUATING SIZE modifiers tend to be used when both axes are different from a comparator in opposite directions. Results are significant at $\alpha = .01$. We cannot reject a null hypothesis in favor of Hypothesis 1; we do not see a significant difference in the distribution of SIZE modifier types when a single axis is different between a target and a comparator. Further factors that may be affecting participant responses are discussed in the next section.

4.3.4. Post-Hoc Analysis. I have illustrated some basic principles of how people use SIZE in reference. However, these experiments also provide much richer information

on how people use SIZE. One immediate question these findings leave is whether it is common to include two INDIVIDUATING modifiers, each referring to a separate axis, when the objects have differences of degree, different polarities across dimensions (Experiment 3). This occurs in a minority of responses (96 responses; mean per participant = 4.8), while it is significantly more common (224 responses; mean = 11.2) to include just one INDIVIDUATING SIZE modifier, an OVERALL SIZE modifier, or neither ($t = -4.292$, $df = 19$, $p < .001$).

We can also confirm the findings in Hermann and Deutsch (1976). Based on responses to Experiment 2 and Experiment 3, in conditions where there is a large difference and a small difference (h++/w+, h+/w++, h++/w-, h-/w++, h- -/w-, h-/w- -, h- -/w+, h+/w- -), if an INDIVIDUATING SIZE modifier is chosen, that modifier will refer to the larger difference more often than the smaller difference (mean for large difference = 3.4; small difference = 2.6, $t = 3.629$, $df = 38$, $p < .001$).

4.3.5. Discussion. This data supports the idea that a difference along two axes in different directions corresponds to SIZE modifiers like "tall" and "thin", and a difference along two axes in the same direction corresponds to SIZE modifiers like "small" and "big". The majority of the data comports with the algorithm sketched in Figure 3, however, this data are probabilistic; the algorithm is not. Assigning probabilities to each of the conditional statements may help to better capture how people use SIZE.

These experiments have also shed some light on some of the other factors that may affect the selection of SIZE modifier. One trend that emerges in the data is the relationship between the selection of INDIVIDUATING or OVERALL SIZE modifier and the ratio between the height and width of the target object itself. Although I did not design the study to test this aspect, the data indicate that the closer the object is to a square shape, e.g., the smaller the difference between height and width, the more likely participants are to use an OVERALL SIZE modifier like "big" or "small". Figure 5 illustrates this trend, where the x-axis is the ratio between the larger axis (height or width) and the smaller axis (height

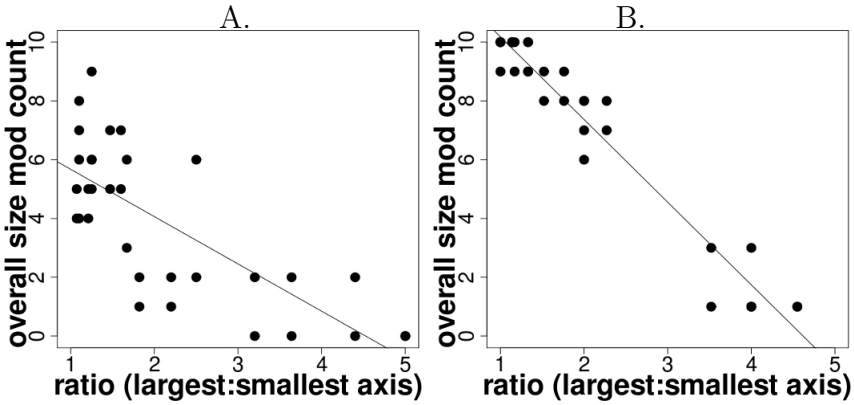


FIGURE 5. Count of OVERALL SIZE modifiers for different height/width ratios in Experiment 1 (A) and Experiment 2 (B), with linear regression. Ratios shown are for the largest axis divided by the smaller axis.

or width) for each stimulus, and the y-axis is the number of responses to the stimulus that include an OVERALL SIZE modifier. In the data from Experiment 2, this trend is quite strong, $r^2 = 0.95$ ($p < .001$). Across conditions with only height or width differing from the comparator object (Experiment 1) – the conditions where we did not see a tendency to use OVERALL SIZE modifiers – there is also a trend, $r^2 = 0.57$ ($p < .001$). Further testing is necessary to examine this effect.

This suggests that the selection of INDIVIDUATING versus OVERALL SIZE modifier may be influenced by the difference in height and width from the comparator object as well as the difference between height and width of the target object itself. INDIVIDUATING SIZE modifiers may be used when only one axis of the target is different from the comparator, however, as the axes of the target itself converge in size, there is a marked increase in preference for OVERALL SIZE modifiers.

We also see a preference to use height modifiers over width modifiers, across the three experiments (mean for height = 6.3, width = 4.7; $t = 4.409$, $df = 59$, $p < .001$). This may reflect that the objects are presented side by side, their heights directly comparable. This brings to light another facet of how the dimensional properties of objects may be reasoned about in a computational model, taking into account a target object's position with respect to a comparator when selecting a SIZE modifier type.

Another possible factor in this trend concerns the confound of type. Different types have different ratios, and also different collocation patterns. Some of these observations may be an artifact caused by the fact that, e.g., "square brownie" is a more common collocation than "square book", and brownies have a size ratio which is close to 1. However, I illustrate in Study 3 that an algorithm that partially uses aspect ratio to decide SIZE modifier type performs reasonably well; the effect that ratio has should be examined further in future work.

4.3.6. Implications for Study 2. This study suggests that the selection of SIZE modifier when referring to real world objects in the presence of another object is influenced by at least two factors:

(1) Whether one or both axes differ from a comparator.
(2) Which axis is the most different from a comparator.

And may be influenced by two further factors:

(1) The location of the target object relative to the comparator.
(2) How similar in size the two axes of the target object are.

With a large enough corpus, we can explore these factors as features in a machine learning approach, and see how well they can predict the kind of size language that people use; this gives us two possible ways to predict SIZE modifiers, via a hand-written algorithm or a classifier, and we can see how each does.

Study 2 examines such a machine learning approach. I scale-up the experiments from this study, expanding to 414 participants. This produces a large corpus of size-denoting expressions, a reasonable enough size to begin training a statistical model. I further add a visual front-end, reading measurements of segmented objects from images; and explore speaker variation, a key concern in generating natural reference.

4.4. Study 2

Study 1 isolated some of the features that may be useful in predicting the size language the speakers will use. The next step is to see if we can *actually* predict that kind of size language that speakers will use based on these features. I find that we can. The corpus from Study 1 becomes the development corpus for Study 2; I train and test on data from a new, larger set of participants.

The last study used images of objects to elicit participant responses. These same images are now used in this study as input to an object segmentation algorithm, and I compare how well we can predict speakers' behavior using the real world measurements of the pictured objects and the image pixel-based measurements. We will see that real world measurements are the best predictors of modifier choice, suggesting that people infer real world size features from images. However, automatically extracted pixel measurements do perform relatively well at predicting modifier choice, offering a potential connection between computer vision and natural language. When speaker identity is taken into account, modifier choice can be predicted with even greater accuracy (around 75%), and the difference between automatically extracted and real world measurements is no longer significant.

The input to the model in this study is therefore the height and width of each object, and the output is the type of SIZE modifier to generate. The SIZE types predicted include INDIVIDUATING SIZE modifiers, corresponding to surface forms such as "tall" and "thin", OVERALL SIZE modifiers, corresponding to surface forms such as "big" and "small", and a type for expressions without SIZE modification (e.g., "the square brownie").

I compare inputs to the model based on real world measurements, image pixel measurements extracted by hand, and image pixel measurements extracted using the semi-supervised SIOX algorithm (Friedland et al., 2005). The semi-supervised approach connects modifier choice to the output of an image processing/computer vision technique known as object segmentation, providing a possible link between natural language and

computer vision. We can see that this approach works well, with an accuracy of 64.95% on unseen test images, but does not perform as well as the models built from real world measurements, which reach 69.44% accuracy. By adding speaker label as a model feature, accuracy from all models improves above 75%, and the difference between the semi-supervised pixel-based measurements and the real world measurements are no longer significant.

I use a decision tree classifier in order to visualize how different features affect the selection of SIZE type. I looked at a range of different models (support vector machines, Naive Bayes classifiers), but found decision trees to perform the best on the development data. Some of the features that emerge with high information gain in these models may be useful in a hand-coded REG algorithm, and I walk through these details in the Results section. The trees built with speaker label also provide a concrete model of speaker variation for this task.

This study therefore makes three primary contributions: (1) a connection between the visual features of a scene and the generation of natural size language; (2) an exploration of visual features that may be useful in further work on human-like REG; and (3) a model of speaker-dependent variation for the SIZE attribute. Both the images and elicited expressions are available at:
http://www.m-mitchell.com/corpora/size_corpus/

4.4.1. Experiment. The design of the elicitation part of this study is identical to the design in Study 1, but scaled up for results from an additional set of 414 participants. I use the same set of objects as in Study 1.

As in Study 1, for each expression, I annotate the modifiers as picking out INDIVIDUATING axes (I) – words like "tall" and "thin" – OVERALL axes (O) – words like "big" and "small"

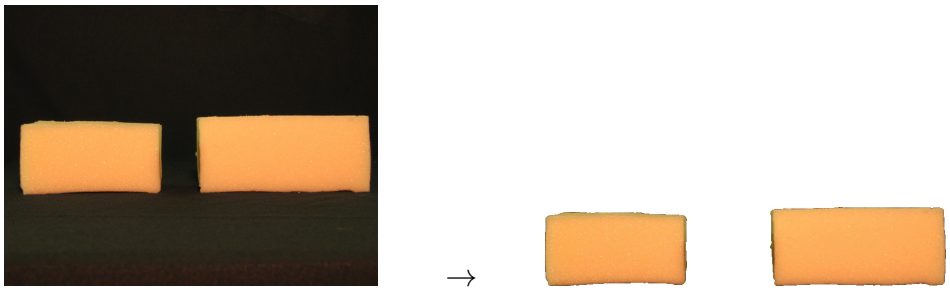


FIGURE 6. Example of original and extracted objects.

– or NEITHER (N). Inter-annotator agreement on a randomly selected 10% of this data is high, Cohen's $\kappa = 0.94$.[4]

In this study, the SIZE modifier types serve as the class labels for each image-based feature vector in the training data. The full list of size-denoting words in this study for each class label is given in Table 6.

Label	Count	Vocabulary
INDIVIDUATING	3307	breadth, broad, deep, elongated, fat, flat, height, high, length, long, low, narrow, short, skinny, slender, slim, squat, stout, tall, thick, thin, wide, width
OVERALL	2614	big, large, little, shrunk, slight, small
NEITHER	703	- -

TABLE 6. Root words for SIZE modifier labels.

4.4.2. Object Segmentation. In addition to the measurements of the real height and width of the objects, I measure the objects' height and width in image pixels. I also extract such information using the SIOX algorithm (Friedland et al., 2005), a semi-supervised method for object segmentation. I explain this algorithm briefly here.

The input for the SIOX algorithm consists of three user specified regions of a given image: known background, unknown region, and known foreground. To notate each region, I manually outline a general selection of the location of each object. The outer region of this selection becomes the known background, and the inner region the unknown region.

[4]729 SIZE modifiers were compared for the agreement score; 5 modifiers only labeled by one annotator are excluded.

By selecting (brushing over) parts of the object, I specify the known foreground. Both known regions are then used in a classification task to identify which sections of the unknown region are background and which are foreground. The resulting output is an outline of the segmented object, separated from the surrounding background.

I then store each of the segmented objects as separate images. With this in place, an image processing tool can be used to extract pixel height and pixel width of each object image. I use CONJURE for this, a command-line based program implemented within ImageMagick (Cristy, Thyssen, & Weinhaus, 2011). Figure 6 shows an example of an image and extracted objects.

4.4.3. Machine Learning. Each of the 96 images represent an <object, condition> stimulus with associated features. There are a variety of size-based visual features available from the heights and widths extracted from the input images, listed in Table 7. These include REFERENT FEATURES, features of the target object alone; COMPARATOR FEATURES, features of the comparator object on the left; and COMPARISON FEATURES, features that store the difference between the referent and comparator. These features may serve as the training/testing data in a machine learning approach where the class label in each instance corresponds to the SIZE type (I, O, or N) used by a particular speaker for a particular image. The classification problem is therefore to use the visual features to predict the SIZE type used by each speaker for each image.

I use C4.5 decision tree classifiers as implemented within Weka (Hall et al., 2009) with default parameter settings. Performance is evaluated using cross-validation, where the set of results from all speakers for each <object, condition> stimulus (each image) is tested against a model trained on all other objects and conditions. Each cross train/test makes up a testing fold, totaling 96 testing folds.

4.4.4. Results. Results are presented in Table 8, listed as the percentage of correct predictions, and in italics, the percentage of testing folds where the predicted type was

#	ID	Description
Referent Features		
1	type	object type
2	ry	height of target
3	rx	width of target
4	rrat	target height:width
6	ryrxdf	target height - target width
5	rsurfar	surface area of target
Comparator Features		
7	dy	height of comparator
8	dx	width of comparator
9	drat	comparator height:width
10	dydxdf	comparator height - comparator width
Comparison Features		
11	ydf	target height - comparator height
12	xdf	target width - comparator width
13	ratdf	target ratio - comparator ratio

Table 7. Visual features extracted from images.

found in the majority of responses. I compare results based on the three kinds of visual measurements:

(1) Automatically extracted image measurements (Auto): the pixel measurements extracted from the segmented objects within the pictures.

(2) Gold-standard image measurements (Gold): pixel measurements measured by hand from the objects within the pictures.

(3) Real World measurements (Real): the actual measurements of the pictured objects.

Accuracy is computed as the number of correct classifications divided by the number of classified instances, over all testing folds. If n is a testing fold in the set of testing folds N, t_i is the true class label of each instance i, and p_i is the predicted class label, then:

	Auto	**Gold**	**Real**	**Oracle**	**Baseline**
Without	64.95%	62.80%	69.44%	75.88%	49.93%
Speaker	*65.63%*	*65.63%*	*75.00%*	*88.54%*	*47.92%*
With	75.33%	77.20 %	76.95%	100%	64.05%
Speaker	*91.67%*	*96.88%*	*95.83%*	*100%*	*71.88%*

TABLE 8. Accuracy across folds.

$$\text{Accuracy} = \frac{\sum_{i \in n \in N} (p_i = t_i)}{\sum_{n \in N} |n|}$$

Accuracy based on the automatically extracted pixel measurements indicates how well the system connecting object segmentation to reference generation performs. Accuracy based on gold standard and real world measurements provide a comparison indicating how well the system performs when the size data are provided manually.

The system connecting object segmentation to natural reference generation (Auto) performs relatively well, predicting 64.95% of response types. The comparison pixel measurement system (Gold) predicts 62.80% of response types, which is not significantly different from the automated approach (paired t-test, $p = 0.4104$).

Interestingly, even though real world measurements may not be clear in photographs, we can see that classification based on these measurements performs significantly better than classification based on the manually or automatically derived pixel measurements (paired t-test, real vs. auto: $p = 0.0363$, real vs. gold: $p = .0188$). This suggests that people are good at reasoning about size in the real world from a two-dimensional image, and the connection between what a computer can see and what it can talk about may be improved with more sophisticated techniques for geometric reasoning.

Since all testing instances in each fold are identical, differing only in class label (the SIZE type), I implement an oracle method to understand the upper bound of this task. This predicts the most common SIZE type in each testing fold, which yields 75.88% accuracy. The results can be compared against a majority baseline that predicts the most common

type from the training data in each fold. Without speaker, the majority class is always I, which is used in 3,307 of the 6,624 instances; 2,614 are O, and 703 are N.

Figure 7 A shows an Auto model built over all data, without speaker labels. We can see that *dydxdf*, the feature for the difference between the comparator's height and width, is selected as having the highest information gain. In other words, the learning approach finds that first splitting up the data based on the value of this feature is the optimal way to distinguish between different SIZE choices; when the model sees a set of visual features for a SIZE choice it has to guess, it will first check whether *dydxdf* is less than or equal to -47 pixels.

Intuitively, the feature *dydxdf* chosen here with the highest information gain here does not make a lot of sense; why the difference between a comparator's height and width would be preferred over something for the referent is not immediately clear. However, it is important to note that the fact that the model finds a particular feature useful to predict human behavior does not mean that that feature is used by people themselves. In particular, information gain tends to prefer variables with a lot of values in the observed data, and this may explain some of the splits it makes. It may also be the case that the model is over-fitting, and this motivates us to remove features in the next section.

In this model, the features related to *ratio* appear as strong predictors of SIZE type. Both the height-to-width ratio of the referent object and the difference in height-to-width ratio between the referent and comparator object are used early on in the trees. This means that features of the target referent itself, as well as features derived from the comparison between referent and comparator, are useful in predicting which label is selected. This suggests that there may be a relationship between the selected SIZE type and how close the height and width of the target object are to one another – for example, when the dimensions are far apart, INDIVIDUATING SIZE modifiers may be preferred, resulting in expressions with words like "tall" and "thin", but when closer together (more square-shaped), OVERALL SIZE modifiers may be preferred, resulting in expressions with words

like "big" and "small". However, as discussed in the last paragraph, what a model finds useful to predict how people refer does not necessarily reflect what a person finds useful when referring. Further testing is necessary to understand if any of the behavior of the decision trees is reflective of human use of these features.

It is interesting that the models are not composed entirely of comparison features, but incorporate features of the referent in isolation, such as its ratio and width. This runs counter to much work in REG, where algorithms usually select features of a referent object based solely on comparison with features of surrounding objects (discussed further in Chapter 2). This data suggest there may also be a benefit in reasoning about the relationship between individual features of the referent object itself before surface realization.

4.4.5. Speaker-Specific Reference Generation. I next add speaker label as a feature in the data and evaluate how well the classifiers perform. This provides a way to distinguish between instances within each testing fold. The trees built using this feature also provide a model of speaker variation.

As shown in Table 8, accuracy improves, and this is significant for all three learned models (Auto, Gold, and Real, $p < .001$). These models outperform a baseline that predicts the majority SIZE type used by each speaker based on the training data in each fold. The Auto models predict 75.33% of the observed SIZE types, and predict the majority type for a testing fold 91.67% of the time. This is not significantly different from the predictions made by the Real models (paired t-test, $t = 1.685, p = 0.095$). The resulting trees have very low depth, tuning decisions to each speaker and then using a small set of individualized features to decide the final SIZE type (Figure 7 B).

4.4.6. Discussion. In this study, I examined how well a small set of objective visual features perform at predicting the type of SIZE modifier selected to refer to everyday objects. I include the size-based features of surface area and height-to-width ratio suggested by Roy (2002) to be correlated with distinct SIZE adjectives. In contrast to earlier work

A.

```
dydxdf <=  -47
|   ratdf <=  -0.097
|   |   xdf <=  4
|   |   |   type = books: 0
|   |   |   type = boards: 0
|   |   |   type = brownies: 0
|   |   |   type = sponges: I
|   |   xdf >  4
|   |   |   ydf <=  7: I
|   |   |   ydf >  7: 0
|   ratdf >  -0.097
|   |   ratdf <=  0.143
|   |   |   rrat <=  0.705
|   |   |   |   dydxdf <=  -49
|   |   |   |   |  rrat <=  0.688
|   |   |   |   |   |   rsurfar <=  41004
|   |   |   |   |   |   |   ydf <=  -14: 0
|   |   |   |   |   |   |   ydf >  -14: I
|   |   |   |   |   |   rsurfar >  41004: 0
|   |   |   |   |   rrat >  0.688: I
|   |   |   |   dydxdf >  -49: N
|   |   |   rrat >  0.705: 0
|   |   ratdf >  0.143
|   |   |   ratdf <=  0.152: I
|   |   |   ratdf >  0.152
|   |   |   |   rx <=  177: 0
|   |   |   |   rx >  177
|   |   |   |   |   rx <=  238
|   |   |   |   |   |   ydf <=  6: 0
|   |   |   |   |   |   ydf >  6
|   |   |   |   |   |   |   ratdf <=  0.245: N
|   |   |   |   |   |   |   ratdf >  0.245
|   |   |   |   |   |   |   |   ydf <=  22: N
|   |   |   |   |   |   |   |   ydf >  22: I
|   |   |   |   |   rx >  238: 0
dydxdf >  -47: I
```

B.

```
dydxdf <=  -47
|   rrat <=  0.674
|   |   spkr = A2E: 2d
|   |   spkr = A2J
|   |   |   ryrxdf <=  -113: 1d
|   |   |   ryrxdf >  -113: 2d
|   |   spkr = A2F: 1d
|   |   spkr = A32
|   |   |   rrat <=  0.561: 2d
|   |   |   rrat >  0.561: 1d
|   |   spkr = A2T: 1d
|   |   spkr = AW5: 2d
|   |   spkr = A37: 2d
|   |   spkr = A3G: 1d
|   |   spkr = A94
|   |   |   ryrxdf <=  -113: 1d
|   |   |   ryrxdf >  -113: 2d
|   |   spkr = A3U
|   |   |   dx <=  205: N
|   |   |   dx >  205
|   |   |   |   ydf <=  8: I
|   |   |   |   ydf >  8: 0
|   |   spkr = AN3
|   |   |   xdf <=  35: 2d
|   |   |   xdf >  35: 1d
|   |   spkr = A34: 2d
|   |   spkr = A1I: 1d
|   |   spkr = A35: 2d
|   |   spkr = A2S
|   |   |   dydxdf <=  -126: 1d
|   |   |   dydxdf >  -126: 2d
|   |   spkr = A19
|   |   |   y <=  152: 1d
|   |   |   y >  152: 2d
|   |   spkr = A3I: 1d
|   |   spkr = A18: 2d
```

FIGURE 7. Pixel-based decision tree without speaker labels (A) and a section of pixel-based tree with speaker labels (B). Decision trees are different across folds.

on machine learning for generating object descriptions, the images are of real objects, the features do not rely on detailed annotation,[5] and the set of predicted classes is kept small. This narrows the machine learning task from earlier related work and avoids data sparsity issues. At the same time, it provides a relatively clear connection between the SIZE aspects of a scene, such as the height and width of a target object, and natural referring expression generation.

[5]In the semi-supervised approach I discuss, the features are extracted from images, but the ability to recognize such features in a scene is limited by how well an object segmentation algorithm works; I control this aspect by looking at clear, uncluttered scenes.

We see that that generating human-like reference to visible, real world objects is possible by reconstructing the problem of REG: Rather than analyzing the SIZE property as a single dimension in feature space (<SIZE:`large`>), it can be analyzed as a multi-dimensional property (<SIZE:[`height:y width:x ratio:z...`] >). In this way, output from a visual analysis may serve as input to a model that selects the most reasonable value (including *neither*) for the given attribute.

Without speaker labels, the models built on real world measurements perform better than the models built on pixel image measurements. This suggests that a connection between language generation and object segmentation can be improved by adding a mechanism to reason about how the two-dimensional image space maps to a three-dimensional real world space.

The models built here point the way to further psycholinguistic work, such as research uncovering other factors that affect the modifier choice made by people (perhaps, for example, cognitive load). Whether the features selected by the decision trees reflect the features humans use when referring to SIZE is an area for future research.

4.4.7. Implications for Study 3. We have seen that we can do reasonably well at predicting among broad SIZE types using a machine learning approach. How well a hand-written algorithm can compare, and whether such approaches can predict more fine-grained SIZE types, remains to be seen.

In the next study, I therefore expand the kinds of SIZE language the models predict, specifying more detailed classes within the two broad SIZE types and exploring further size features. I find that we can successfully predict even finer-grained SIZE types, and the process of refining the hand-written algorithm introduced in Study 1 suggests further features for the machine learning approach introduced in Study 2. Both the hand-written algorithm and machine learning approach perform comparably, and reach high precision/recall when tested in an entirely new domain.

Type		Axis	Polarity
INDIVIDUATING	(<ind, y>, +)	y	+
	(<ind, y>, -)	y	-
	(<ind, x>, +)	x	+
	(<ind, x>, -)	x	-
OVERALL	(<over>, +)	x, y	+
	(<over>, -)	x, y	-

TABLE 9. SIZE types.

4.5. Study 3

Study 1 established that people use SIZE modifiers in predictable ways, isolating features that influence the selection of SIZE modifier type. In Study 2, I began generating different SIZE types from these features, and found that the generated types matched the type used by actual speakers to the same stimulus in well over the majority of cases.

However, the SIZE types I examined were quite broad: *Individuating* SIZE modifiers, which refer to at least one axis, and *overall* SIZE modifiers, which refer to the overall size of the object. In this study, I further refine the broad SIZE types, breaking them into types corresponding to both polarity and axis. Words like "tall" and "big" denote a positive polarity (+), and words like "small" and "thin" denote a negative polarity (-). Words like "tall" and "short" may be used to refer to difference along the y-axis of an object, and words like "fat" and "thin" may be used to refer to differences along the x-axis. The six abstract SIZE types based on these distinctions are listed in Table 2, and a few examples of corresponding surface forms are listed in Table 10. These types may be used to generate different surface realizations from the same underlying semantic form, for example, (<ind,y>, -) may be used to produce adjectives ("the short box"), relative clauses ("that is shorter"), and prepositional phrases ("with less height").

Using this more fine-grained distinction, I predict modifiers in two domains: The Size Corpus established in Studies 1 and 2; and a new domain, the Craft Corpus, discussed in Chapter 3. Since the Size Corpus influenced the design of the hand-coded algorithm

Type	Examples		
(<ind, y>, +)	taller	thicker	longer
(<ind, y>, -)	shorter	thinner	short
(<ind, x>, +)	longer	thicker	wider
(<ind, x>, -)	thinner	shorter	narrower
(<over>, +)	larger	bigger	big
(<over>, -)	smaller	small	smallest

TABLE 10. Top three surface forms for each SIZE category in the Size Corpus.

and the features selected in the machine learning approach, it is interesting to see how well we can predict SIZE types in an entirely new domain. I find that we can do quite well, with above 80% precision and recall.

4.5.1. The Size Algorithm. Study 1 introduced the beginnings of a hand-coded SIZE generating algorithm, but the experiments suggested further features that should be taken into account when generating SIZE. In particular, I found that there may be an effect of aspect ratio on the selection of modifier type. This probabilistic finding can now be incorporated into a full SIZE-generating algorithm, which we detail in Figure 9. This algorithm is a model of the findings suggested from the first study, listed again in Figure 8, and is used when the following preconditions are met:

(1) There is a target referent and one or more comparator objects
(2) Each comparator has two dimensions that can be compared with the target referent's dimensions

As input, the algorithm takes the width and height of the referent (rx, ry) and the width and height of the comparator of the same type or average of the distractors of the same type as the referent (dx, dy). The algorithm outputs one of the SIZE types listed in Table 2.

Lines 3 and 6 of SIZEMOD model the first finding in Figure 8, creating a structure to generate an OVERALL SIZE modifier ('over') with the appropriate polarity. Lines 4 and

1. When two dimensions differ in the same direction between a referent object and another object of the same type, an OVERALL SIZE modifier will be produced more often than an INDIVIDUATING SIZE modifier.
2. When two dimensions differ in opposite directions between a referent object and another object of the same type, an INDIVIDUATING SIZE modifier will be produced more often than an OVERALL SIZE modifier.
3. The closer the aspect ratio of an object, the more likely participants are to use an OVERALL SIZE modifier.

FIGURE 8. Size findings from Study 1.

7 create a structure to generate an INDIVIDUATING SIZE modifier ('ind') referring to the axis with the largest difference, with the appropriate polarity. Here, the modifier type selection reflects the second finding in Figure 8, while the selected axis is chosen based on the conclusions of Hermann and Deutsch (1976).

Lines 5, 8, 9, and 10 are all cases where one axis is different from the comparator and one axis is not. In these cases, following the third finding in Figure 8, I calculate the ratio of difference between the axes (CALCRATIO). This is a stochastic process that models speaker preference for a modifier type as a function of the object's aspect ratio. The closer the ratio of the x / y axes is to 1, the more likely the algorithm is to generate an OVERALL SIZE modifier.

Line 11 handles the case where both the referent and comparator have the same height and width. In this case, no SIZE modifier is generated.

The Size Corpus from Study 1 and Study 2 provides information about size when there is a single comparator of the same type, however, in practice, a referent may be competing against several comparator objects. To address this, the algorithm must compare a referent's height and width against a larger set of heights and widths. A straightforward way to apply such a comparison is to take the *average* height and width of the items in the contrast set. Such an approach has also been suggested by work in vision, which has shown that observers know the mean size of a collection of homogeneous objects quite

Input: Referent height, width (ry, rx)
Average height, width for comparators of referent's type (dy, dx).
Output: SIZE modifier type (See Table 2).

```
SIZEMOD(rx, ry, dx, dy):
 1. axes = <rx, ry, dx, ry>
 2. case (mod, pol) of:
 3.    ry > dy and rx > dx:     (<'over'>, 1)
 4.    ry > dy and rx < dx:     LargestDimDiff(axes)
 5.    ry > dy and rx == dx:    (CalcRatio(axes, 'y'), 1)
 6.    ry < dy and rx < dx:     (<'over'>, 0)
 7.    ry < dy and rx > dx:     LargestDimDiff(axes)
 8.    ry < dy and rx == dx:    (CalcRatio(axes, 'y'), 0)
 9.    ry == dy and rx > dx:    (CalcRatio(axes, 'x'), 1)
10.    ry == dy and rx < dx:    (CalcRatio(axes, 'x'), 0)
11.    ry == dy and rx == dx: (None, None)
12.  return (mod, pol)

LARGESTDIMDIFF(<rx, ry, dx, dy>):
  axis = axis with largest difference between r and d (x or y)
  pol = direction of difference (0 or 1)
  return (<'ind', axis>, pol)

CALCRATIO(<rx, ry, dx, dy>, axis):
  if ry > rx: greater = ry, smaller = rx
  else: smaller = ry, greater = rx
  p = (greater/smaller) - 1
  if p > 1: p = 1
  v = round(100 * p)
  i = random integer between 1 and 100
  if i > v: mod = <'over'>
  else: mod = <'ind', axis>
  return mod
```

FIGURE 9. Size algorithm. 1 is used to designate a positive polarity (+) and 0 a negative polarity (-).

accurately but retain little information about the size of the individual objects (Ariely, 2001). Since SIZE is more common when an item of the same type is in the scene (Brown-Schmidt & Tanenhaus, 2006), it may be suitable for the algorithm to compare size using the height and width average of other items of the same type. This also provides a simple

way to model the size expectations of the referent relative to similar items. I do not test how well this approach works with size outliers; this is a clear area for future work.

4.5.2. Machine Learning. One of the strengths of applying machine learning to this task is that it may be constructed as a series of binary classification problems, where a model is built for each SIZE type. This allows more than one modifier to be generated for each referent, while avoiding issues of data sparsity inherent in training every combination of SIZE as a separate class. The machine learning approach therefore has functionality that the hand-coded size algorithm does not have; it is able to predict sets of modifiers for a referent instead of being limited to a single modifier. This flexibility is a benefit to the machine learning approach over the hand-coded algorithm, and I return to this issue in Section 4.5.5.

To build the models, each expression in the Size Corpus from Study 2 was annotated to mark the SIZE modifiers and their types (Table 2). A random selection of 10% of the dataset was checked for inter-annotator agreement. The annotators found that many of the annotated brownie references picked out the z-axis, the third dimensional axis pointing inwards in the picture; although the images are two-dimensional, both annotators reasoned about the three-dimensional shape to resolve references to all three axes. This is probably especially true for the brownies stimuli due to the angle of the camera, where differences in height may appear to be along the z-axis. In future work, it would be better to control this aspect, perhaps making only two dimensions visible. For this data, I group those modifiers for z- and y-axes together. Inter-annotator agreement was quite high at $\kappa = 0.94$.[6]

As in Study 2, the models are constructed using C4.5 decision tree classifiers as implemented within Weka (Hall et al., 2009), with default parameter settings. I did not find a significant improvement in accuracy on the development set with different pruning methods or normalization. Each feature vector used by the models lists visual size features

[6] 729 SIZE modifiers were compared for the agreement score; 5 modifiers only labeled by one annotator are excluded.

#	ID	Description
REFERENT FEATURES		
1	ry	target height
2	rx	target width
3	rrat	target height:width
4	ryrxdf	target height - target width
5	rsurfar	surface area of target
COMPARATOR FEATURES		
6	dy	comparator height
7	dx	comparator width
8	drat	comparator height:width
9	dydxdf	comparator height - comparator width
10	dsurfar	surface area of comparator
COMPARISON FEATURES		
11	ydf	target height - comparator height
12	yratio	target height / comparator height
13	xdf	target width - comparator width
14	xratio	target width / comparator width
15	ratdf	target ratio - comparator ratio
16	discx	1 if rx > dx; 2 if rx == dx; 3 if rx < dx
17	discy	1 if ry > dy; 2 if ry == dy; 3 if ry < dy

TABLE 11. Expanded visual features for each expression, including features in Table 7. Features 16 and 17 mirror the size algorithm's comparisons.

Type	<ind, y>		<ind, x>		<over>	
	+	-	+	-	+	-
Observed	22	10	3	0	51	43

TABLE 12. Frequency of observed SIZE modifier types in the Craft Corpus.

that characterize each image, such as the size of the referent and comparator's axes, and differences between the two. I also provide a set of features reflecting the comparisons made in the hand-coded algorithm. The feature set is listed in Table 11.

4.5.3. Testing Corpus. To evaluate how well the models perform in a new domain, I use the Craft Corpus from the experiment in Chapter 3. The 2010 experiment is a different task, and differs in several critical ways from the 2011 experiment: (1) It

was conducted in-person, using three-dimensional objects; (2) the referring expressions were produced orally; (3) there were many different objects in the scene, and (4) the objects had a variety of different visual properties: values were different for TEXTURE, MATERIAL, COLOR, SHEEN, etc., as well as SIZE along all three dimensions. Subjects referred to objects as, for example, "the longer silver ribbon", and "small green heart". Table 12 lists the frequency of each observed SIZE type in this corpus.

As discussed above, I adapt the size algorithm to the new domain by taking the average height and width of all comparators of the same type, and comparing the referent against this average. The implications of this are three-fold: (1) Comparisons are limited to those items of the same type; (2) comparisons are limited to those items in an immediately surrounding group; and (3) comparisons are against a general 'gist' of the surrounding scene, instead of individual measurements.

To adapt the classifiers to the new domain, I remove all direct measurement features from training and testing; work on the development set suggests that including all listed features achieves the best precision and recall when training and testing in the same domain, however, when expanding to a new domain, certain features overfit the model to the development domain. This includes features 1 (ry, target height), 2 (rx, target width), 4 (ryrxdf, target height - width), 6 (dy, comparator height), 7 (dx, comparator width), 9 (dydxdf, comparator height - width), 11 (ydf, target height - comparator height), 13 (xdf, target width - comparator width). Removing these features allows the classifiers to build models from relative measurement features alone, and helps minimize overfitting to any one domain.

4.5.4. Evaluation. Before testing on the new domain, I test how well the two approaches do on the Size Corpus. The construction of the size algorithm was informed by this corpus, and so this provides a measure of how well the algorithm does in the domain for which it was designed. The decision trees are evaluated in this domain using leave-one-out validation, where the set of expressions for a referent containing at least

```
discy <= 1: no
discy > 1
|   discx <= 1: no
|   discx > 1
|   |   drat <= 1
|   |   |   xratio <= 0.909: yes
|   |   |   xratio > 0.909
|   |   |   |   discy <= 2: no
|   |   |   |   discy > 2
|   |   |   |   |   rrat <= 0.455
|   |   |   |   |   |   xratio <= 0.910: yes
|   |   |   |   |   |   xratio > 0.910
|   |   |   |   |   |   |   rrat <= 0.413: yes
|   |   |   |   |   |   |   rrat > 0.413: no
|   |   |   |   |   rrat > 0.455: yes
|   |   drat > 1: no
```

FIGURE 10. Example (partial) decision tree, binary classification: Training on Mechanical Turk data, direct measurement features removed, model for inclusion of (<over>, 0). Values in cm.

one SIZE modifier is tested against the models trained on the SIZE expressions for all other referents. An example tree is shown in Figure 10. Features developed from the hand-coded algorithm (features 16 and 17 in Table 11) appear to have high discriminative utility in the trained models.

Unlike the machine learning approach, the size algorithm generates no more than one SIZE type for each referent, although participants may produce several. To understand the upper bound of both approaches, I therefore implement an oracle method for the size algorithm (ORACLE_{alg}) that always guesses the most common SIZE type for each referent, and an oracle method for the classifiers (ORACLE_{tree}) that always guesses the most common set of SIZE types for each referent.

To understand the lower bound, I implement a baseline method that guesses the most common SIZE type and most common set of SIZE types in the training data for each testing fold. The most common set of SIZE types across folds contains a single modifier, making the baseline of the two approaches equivalent.

I evaluate the systems using precision and recall. Since I am comparing the set of predicted modifiers with the set of modifiers that a description contains, it would have been possible to use the DICE metric (Dice, 1945), as has often been done in evaluations

Model	**Mturk precision/recall**	**Crafts precision/recall**
BASELINE	25.7% / 24.5%	16.4% / 16.4%
ORACLE$_{alg}$	80.5% / 72.7%	89.1% / 89.1%
ORACLE$_{tree}$	79.5% / 76.0%	89.1% / 89.1%
SIZE ALGORITHM	69.7% / 63.4%	81.3% / 81.3%
DECISION TREE	65.4% / 65.7%	80.5% / 81.3%

TABLE 13. Precision and recall for models, testing on expressions that contain SIZE. The size algorithm is averaged over 5 iterations.

of REG algorithms (Gatt & Belz, 2008). But DICE does not distinguish between recall (i.e., modifiers that are not predicted but should have been) and precision (i.e., modifiers that are predicted but should not have been), collapsing both of these into one single metric. For my purposes, it will be more informative to separate precision and recall. Given:

$\mathbb{O}_e$ = The set of SIZE modifier types observed in an expression e

$\mathbb{P}_r$ = The set of SIZE modifier types predicted for a referent r

$\mathbb{E}_r$ = The multiset of expressions for a referent r

$\mathbb{R}$ = The multiset of expressions $\mathbb{E}_r$ for each test referent r

$$\textbf{Precision} = \frac{\sum_{e \in \mathbb{E}_r \in \mathbb{R}} \frac{|\mathbb{P}_r \cap \mathbb{O}_e|}{|\mathbb{P}_r|}}{|\mathbb{R}|}$$

$$\textbf{Recall} = \frac{\sum_{e \in \mathbb{E}_r \in \mathbb{R}} \frac{|\mathbb{P}_r \cap \mathbb{O}_e|}{|\mathbb{O}_e|}}{|\mathbb{R}|}$$

Table 13 shows how well the different systems perform. Testing instances are limited to those that contain a SIZE modifier. The second column lists precision and recall on the Size Corpus. The difference in results between the two systems is not statistically significant.

The third column of Table 13 lists how well the systems do when tested on the new domain, the Craft Corpus. The precision and recall values here are identical for the systems that generate one modifier because almost all SIZE expressions in the Craft Corpus contain just one modifier. This also allows a more direct comparison between the two systems, as both the lower bounds (BASELINE) and upper bounds (ORACLE) of the two systems are equal.

As discussed in Section 4.5.3, both systems are adapted slightly for the new domain. The size algorithm uses the height and width *average* of items that are the same type as the referent. The decision trees are trained on the full Size Corpus, and when the models are built from all of the features listed in Table 11, precision/recall on this task is 44.1%/48.1%. However, once the classifiers are adapted to the subset of relative measurement features, there is a large jump for both measures.

The two systems perform similarly. The size algorithm achieves just over 81.3% precision and recall, while the machine learning approach reaches 80.5% precision and 81.3% recall, and the differences between the two methods are not statistically significant. Oracle accuracy is higher by around 8%, suggesting that both systems are reasonable, and further work may want to finesse the kinds of size information that each uses.

4.5.5. Discussion. It is interesting that both systems perform better in the new domain. Both were built based on typed reference to one of two rectilinear solids in a two-dimensional photograph, and still produce reasonable output to spoken reference to one of several three-dimensional objects with different shapes in a much more descriptive task. The two systems likely perform better on the Craft Corpus than the one they were developed on because in the Craft Corpus, almost all expressions contain just one SIZE modifier (only one expression had more).[7]

The machine learning approach does poorly when it uses the same set of features in both domains, however, by removing those features that may lead to overfitting – the

[7]This was "the smallest long ribbon", which both models fail to predict.

direct measurements of individual objects, which vary across the different domains – it dramatically improves in the new domain. The difference in precision and recall between the two systems is not statistically significant, with values above 80%.

A notable difference between the two systems is that the machine learning approach can predict any number of SIZE modifiers, while the size algorithm is limited to predicting one modifier (or none). The upper and lower bounds are the same for both in the Craft Corpus discussed here, however, the classifiers' ability to predict when several SIZE modifiers will be included may help extend this method in other domains.

One immediate question that arises from this work is how to move from abstract SIZE type to surface form. For some modifiers, this will be relatively straightforward, but for others, e.g., using (<over>, +) to generate the phrase "the second largest one", further functionality must be in place to reason about individual sizes of objects in the contrast set.

Both systems may be developed further by modeling speaker variation. As shown in Study 2 and in previous work (Viethen & Dale, 2010), adding speaker label as a feature within the decision tree models improves performance. Creating a more concrete way to handle speaker variation may lead to more natural output.

In the size algorithm, speaker variation may be applied several ways. Currently, the algorithm's CALCRATIO function decides which of the two broad SIZE modifier classes to generate by using a random number generator. This was implemented based on speaker variation in cases where the aspect ratio of an object approaches 1 (Figure 8). A similar technique may be applied throughout the algorithm, where a prior is assigned to various decisions based on an analysis of how speakers behave. Another method could apply slightly different versions of the algorithm to different speaker models, where some more detailed aspects of the algorithm are varied for different speaker profiles – for example, placing a preference on height over width within a threshold of axis size similarity.

4.6. Conclusions and Future Work

I have presented two methods for generating SIZE modifiers. Both utilize the dimensional aspects of objects in a scene to decide among six broad SIZE categories, which may be used to inform the selection of SIZE modifier in a realized surface string. Both work relatively well when generating SIZE modifiers for two-dimensional images of three-dimensional rectilinear solids, and are extensible to a new domain of real world three-dimensional objects with irregular shapes.

One of the next clear steps in developing the hand-coded size algorithm is to add functionality for generating sets of modifiers. I would also like to explore different features and the effect they have on the overall accuracy of the different approaches. I hope to address modifiers that pick out specific configurations of multiple axes, e.g., "stout" may be realized from {(<ind, x>, +), (<ind, y>, -)}. Methods for reasoning about the distance and relative orientation between the target object and its comparators may guide which axis is referred to, and the systems should be further expanded to real world objects by adding mechanisms to handle a third z-axis. A better understanding of when a difference along an axis is small enough not to be salient would help connect these approaches more closely to a visual input, placing constraints on when the outlined cases apply.

To broaden generation to a new domain, in Section 7.5 I took the height and width average of same-type objects. It is an open question whether this approach works well when there are clear size outliers, and I note this for future work. It would also be useful to use the decision trees discussed in this chapter to cluster speakers, and generate individual speaker variation by generating from particular speaker clusters. In this approach, one could make a tree for each speaker, and then cluster similar trees together. It may also be useful to explore further classification approaches beyond SVMs and decision trees, e.g., k-nearest neighbor, which may be better at predicting size modifier preferences based on similarly sized objects in the training data.

I hope to address other kinds of properties of real world referents using a similar methodology, in particular, reasoning about the inclusion of spatial prepositions between objects. It seems intuitive that some of the same features defining size of an object could be used along with distance between objects to understand the perception of their spatial relations and how those relations are described; this has also been suggested in recent work, e.g., Kelleher and Costello (2009). By further defining when different properties are used, how distinct properties interact, and the features affecting their realization, I hope to continue to expand the methods to generate naturalistic reference.

Chapter 5

Typicality: Shape and Material

5.1. Introduction

Consider the picture below. What is this a picture of? You may say "a dog", or perhaps "a golden retriever". You may also say it is "a *three-legged* dog". This can be contrasted with references that sound considerably more marked – it seems odd to say this is "a *two-eared* dog" or "a dog *with a nose*". So why does *three-legged* sound fine, while *two-eared* and *with a nose* do not?

This chapter focuses on the role of *typicality* in reference to real world objects. I test whether changing the typicality of an object's properties affects reference to it, and whether we can predict how an object will be referred to by comparing its specific attribute-values against a knowledge base of stored, typical attribute-values for the object type. I examine this effect using the attributes of MATERIAL and SHAPE, which commonly appear in descriptions of visible objects (see Chapter 3). This is an area that has received little attention in work on referring expressions generation, and I hope to make some initial conclusions that can provide a basis for further research.

What is *typical* about an object is flexible and will not necessarily be identical across participants, which makes predicting behavior based on a list of typical features problematic. However, if mental representations of typical object features do play a role in object description, then we may be able to approximate typicality using the most frequently named features for objects.

To establish the most frequently named features for objects, we use semantic feature production norms. Semantic feature production norms provide a set of common properties for basic-level concepts, and are collected to explore conceptual representations such as typicality (Rosch & Mervis, 1975) and semantics (Wu & Barsalou, 2009). We use McRae's norms (McRae, Cree, Seidenberg, & McNorgan, 2005; McRae, 2011) (see Table 1), which to our knowledge is the largest source of production norms to date. McRae's norms were collected by providing participants with 10 blank lines for each basic category and asking them to list features for each, such as physical (perceptual) properties (how it looks, sounds, smells, feels, and tastes), functional properties (what it is used for and where and when it is used), and other information, such as encyclopedic facts (e.g., where it is from).

I consider *atypical* values to be those not listed by any participant for the object. I group features into the categories of SHAPE and MATERIAL, which correspond to a subset of the "external_surface_property"/"external_component" labels in the norms (for SHAPE) and all of the "made_of" labels (for MATERIAL). The study is primarily run on English-speaking North Americans, the same general cultural group that established the norms. For those features that McRae's norms do not provide, I rely on what features were available.

In testing typicality in a domain of real world objects, one issue that immediately arises is the *interconnectedness* of different attributes. For example, the MATERIAL attribute often entails COLOR and TEXTURE, among others. An object made of wool is often fuzzy or rough (TEXTURE values), while an object made of wood is often tan or brown, and

for everyday objects, tends to be smooth (COLOR and SMOOTHNESS values). Ideally, participants would refer only to those attributes that I vary, SHAPE or MATERIAL; but they may instead refer to interconnected attributes, calling a woolen bowl "coarse" or "flexible", or a plastic coin a "fake" coin.

Another issue that arises using complex objects is that of lexical choice. This is particularly clear for SHAPE, where it may be difficult to lexicalize or describe various shape manipulations. Coins with a flowered shape may be called "flowery" or "ruffled" perhaps (see Figure 2); but neither of these descriptors are as common as a word like "round", and this may affect whether the descriptor is included. Some shape manipulations may also be realized as a prenominal modifier ("*octagonal* mug"), while others may be realized using a more syntactically complex postnominal modifier ("ruler *with holes in it*"). This undoubtedly also affects whether or not a shape difference will be described. Further complicating the issue is that an object's shape is often indicated by its name (Markman, 1989; Landau & Jackendoff, 1993), and so when presented with an object with an atypical shape, subjects may think it is a fundamentally different object from the one intended (see Figure 1).

A further issue arises from visual saliency. Recent research suggests that MATERIAL is not available pre-attentively and is inefficient for guiding attention (Wolfe & Myers, 2010), while SHAPE may be much more accessible when scanning the scene. Indeed, the study discussed in Chapter 3 suggests that SHAPE modifiers may be preferred to MATERIAL modifiers, a conflating factor in comparing the frequency with which each attribute is named when atypical.

FIGURE 1. Bowl, Sugar Bowl, Creamer, Teacup, Mug, Pitcher:
Similar objects with different shapes tend to have different names.

FIGURE 2. The production of a modifier may be affected by how lexicalizable or common it is. Here, an adjective like "ruffled" or "flowery" (on the right) is less common than an adjective like "round" (on the left).

Such issues make isolating individual object attributes in a controlled study much more difficult than it may seem at first blush, and may be part of the reason why studies using a range of complex real world objects are few and far between. In this chapter, I examine typicality in the interplay of both the MATERIAL and SHAPE attributes, and find that subjects tend to prefer *atypical* over typical SHAPE values when describing objects, but do not find strong evidence of this for atypical MATERIAL values. In the hope of informing future research in this area, I also provide a discussion of the process of collecting materials for this study, some of the difficulties encountered, and the solutions to them.

5.2. Background

This study follows the idea that the concepts we access when viewing an object are informed by previous experiences we have had with other objects of the same type. Such experiences give rise to mental representations of the object that may include exemplars, examples of specific instances in which we have previously viewed the object before (Medin & Schaffer, 1978; Wu & Barsalou, 2009; Frassinelli, 2010); and prototypes, a generalized notion of what the object is typically like (Rosch, 1975; Rosch et al., 1976, further details in Chapter 2 Section 2.2.2).

Theories of object recognition posit that the visual representation of an object is matched to structural descriptions and functional attributes stored in long term memory; visual

processes encode the SHAPE, COLOR, TEXTURE, etc., of an object to match that visual input to an object category (Logothetis & Sheinberg, 1996; Humphreys, Price, & Riddoch, 1999; Riesenhuber & Poggio, 2000, further details in Chapter 2 Section 2.3). Word production emerges from a coupling of these categorical concepts with the articulatory motor system active during object naming (Kosslyn, 1980; Levelt, Roelofs, & Meyer, 1999).

If stored features of an object category are accessed when viewing a specific instance of that object, then those features may influence object naming. For example, if the MATERIAL *metal* is a typical property of spoons, then a wooden spoon may be noticeably "wooden", and described as such more often than a metal spoon would be described as "metal".

This study tests such an intuition, presenting participants with atypically-featured objects and measuring the influence these features have on object descriptions. We will see some evidence that typicality affects how an object is referred to, and the results suggest further processes at play. The findings from this study may be useful to create richer, more natural and descriptive visual descriptions.

5.3. Material Collection

In keeping with previous chapters and the goals of this thesis, I choose naturalistic, everyday, inanimate objects for the study. The initial list of possible objects included all inanimate objects from McRae's norms that could fit on an experiment table, and this set was narrowed down by availability and my abilities to control the visual properties of the objects.

Pilot work for this study suggested that the test objects had to match as closely as possible in COLOR and SIZE, while being clearly typical in one other attribute (MATERIAL or SHAPE) and clearly atypical in the other. Without this control, COLOR and SIZE attributes tended to be preferred for distinguishing the objects; this further reflects the

idea that these two attributes are particularly salient for visual reference (cf. Chapters 3 and 4).

For example, a ruler, made of white paper (an atypical MATERIAL for a ruler) contrasted with a wooden ruler (a typical MATERIAL) was called "the *white* ruler" rather than "the *paper* ruler". Such attribute interconnections made the process of collecting objects particularly challenging. Objects with different shapes still had to be roughly the same size, and objects with different materials still had to be the same color. To meet these demands, many objects had to be hand-crafted by professionals or else acquired after sifting through and returning to several different stores, often buying objects for which a matching test pair could not be found or created.

I aimed to sample a variety of materials and shapes, which also made finding appropriate objects difficult – having a large subset of atypical MATERIAL be clay, for example, which is easy to manipulate and therefore easy to use to create test objects, may not tell us about typicality, but about clay. I therefore had to work with tools capable of cutting hard plastic, metal, wood, etc., with enough precision to keep the manipulated objects from standing out as the obvious test objects.

Further issues arose when an object with an atypical value was called by some other name. For example, a coin made of zinc could be called "the dark coin" and scissors with jagged edges could be called "pinking shears". Many objects considered from McRae's norms were not used because manipulating the shape or material was likely to elicit interconnected attributes.

The final set of test objects are listed in Table 2, along with their typical and atypical values for SHAPE and MATERIAL.

5.4. Annotation

A hallmark of annotating materials from real world objects is that there may not be agreement in all cases as to the attribute being described. Is "unsharpened" a realization

OBJECT	SHAPE			MATERIAL	
bowl	curved (7)	round (19)		ceramic (5)	plastic (8)
box	rectangular (7)	square (21)		cardboard (16)	wood (7)
envelope				paper (14)	
key				metal (20)	
mug				ceramic (10)	
ruler	rectangular (6)	straight (6)	flat (5)	plastic (12)	wood (17)
screws	flat_head (6)			metal (17)	

TABLE 1. McRae's Norms: Visual features for test objects. Production Frequency, the number of participants out of 30 who listed this feature, is in parentheses. I add typical SHAPE for *envelope*, *key*, and *mug* based on what was most common in stores where the objects were purchased. The final set of objects and properties are listed in Table 2.

	Typical		Atypical	
Object	**SHAPE**	**MATERIAL**	**SHAPE**	**MATERIAL**
bowl	round	ceramic	flower	wool
box	square	cardboard	heart	clay
envelope	rectangle	paper	square	foam
key	rounded head	metal	square head	wood
mug	round	ceramic	octagonal	metal
ruler	rectangle	wood	with holes	paper
screw	flat circular head	metal	oval head	plastic

TABLE 2. Typicality Study: SHAPE/MATERIAL objects.

of a PROCESS attribute? Does "poky" correspond to SHAPE? I develop the following set of attributes to annotate the data, listed in Table 3. A second annotator other than the main author was given a random subset of 20 experiment test objects, and told to annotate each expression as to which attributes it contained from those shown in Table 3, with the given examples. To check inter-annotator agreement for the two properties of SHAPE and MATERIAL, I treat each as binary categorical variables. Cohen's κ is very good for SHAPE (κ=.894) and good for MATERIAL (κ=.798). Disagreements were over whether "metallic" in "the non-ribbed metallic cup" was a MATERIAL or a TEXTURE, whether "heart" in "a heart-shaped box" was a SHAPE or a TYPE, and whether "silver" in "a silver round cup" was a COLOR or a MATERIAL.

ANALOGY	"like a cat"	SHAPE	"square"
COLOR	"blue"	SHEEN	"shiny"
FILL	"empty"	SIZE	"little"
FLEX	"flexible"	SUBJECTIVE	"weird"
FORM	"open"	SUBTYPE	"ball point"
HARDNESS	"hard"	TEXTURE	"rough"
INTENSITY	"dark"	TYPE	"box"
LOCATION	"close to me"	USE	"for oil"
LUMINESCENCE	"bright"	WEIGHT	"light"
MATERIAL	"copper"	**Examples of phrases conveying more than one attribute**	
OPACITY	"clear"		
OTHER	"other", "another"	ANALOGY:SHAPE	"shaped like a P"
PART-WHOLE	"with a slot on top"	TYPE:SHAPE	"diamond"
		PROCESS:ANALOGY	"that opens like a purse"
PROCESS	"adjustable"		

TABLE 3. Attributes considered and example surface forms.

5.5. Method

My goal is to discover whether an atypical attribute-value is more likely to appear in a description than a typical attribute-value. I test this for two attributes, SHAPE and MATERIAL.

5.5.1. Participants and Design. Thirty native English speaker with normal or corrected vision were recruited through word of mouth and online ads, 17 males and 13 females, aged 20-55. Trials were arranged in a repeated measures design. Participants were assigned randomly to one of two groups, **Atypical SHAPE** or **Atypical MATERIAL**. In the Atypical MATERIAL group, subjects referred to objects with a typical SHAPE and an atypical MATERIAL. In the Atypical SHAPE group, subjects referred to objects with a typical MATERIAL and an atypical SHAPE. Each participant referred to an equal number of fillers and seven target objects, in a randomized order. Each target

item could be minimally distinguished by one attribute – SHAPE or MATERIAL. Four male subjects and one female subject were randomly excluded to balance gender, leaving 6 female and 6 male subjects in each group. This held-out set was later used to replace two subjects identified as outliers (see Section 5.6).

FIGURE 3. Objects used in study.

5.5.2. Materials. Subjects were presented with an array of different objects, primarily office-type objects, as shown in Figure 3. Amongst these objects were SHAPE/MATERIAL pairs for the seven target objects – bowls, boxes, envelopes, keys, mugs, rulers, and screws (see Table 2). Most filler objects[1] could be distinguished by type (head noun) alone. A few filler objects of the same type could be distinguished by SUBTYPE, SIZE, MATERIAL, and FORM. The full list of filler objects are listed in Table 4, along with attribute-values. Objects of the same type that could only be uniquely identified with additional disambiguating modifiers are grouped together.

5.5.3. Procedure. This study follows a director-matcher paradigm, where the director ("the speaker", the participant) sees pictures of objects arranged on a grid and has

[1]Filler objects also serve as *distractor* objects in the traditional REG sense (Dale & Reiter, 1995).

battery	ball	(FORM:spiky, MATERIAL:plastic)
coin	*ball*	(FORM:smooth, MATERIAL:styrofoam)
bracelet	clip	(SUBTYPE:binder-clip)
c-clamp	clip	(SUBTYPE:paperclip, SIZE:big, MATERIAL:metal, FORM:ribbed)
cube*	clip	(SUBTYPE:paperclip, SIZE:small, MATERIAL:metal)
fork	clip	(SUBTYPE:paperclip, COLOR:yellow, SHAPE:triangle, MATERIAL:plastic)
funnel	*clip*	(SUBTYPE:paperclip, COLOR:yellow, INTENSITY:dark, SIZE:small)
paintbrush	clip	(SUBTYPE:hair-clip)
pencil	comb	(COLOR:black)
rolling-pin	comb	(COLOR:red)
rubber-band	*pen*	(COLOR:black, SUBTYPE:mechanical,finepoint)
salt-shaker	*pen*	(COLOR:blue, SUBTYPE:refillable,fine point,rollerball)
scissors	pen	(COLOR:blue, SUBTYPE:bic,ballpoint)
sphere*	pen	(COLOR:purple, SUBTYPE:felt)
staple-remover	pushpin	(COLOR:white)
stapler	*pushpin*	(OPACITY:clear)
toothpick	*pushpin*	(MATERIAL:metal, PART-WHOLE:SHAPE:flat)
	pushpin	(MATERIAL:metal, PART-WHOLE:SHAPE:sphere)

TABLE 4. Filler objects. Objects for which reference was not elicited are in italics. Objects that could not be distinguished by type alone are listed on the right, alongside possible distinguishing attribute-values.
* These objects were varied by COLOR/SIZE as part of a separate pilot experiment.

to instruct the matcher ("the mover", an assistant) to put them in the same positions. Each picture contains five objects: four fillers and one test object, in different positions across images (see Figure 4).

Subjects sat at a table opposite another student assisting in the study. The experimenter sat at the head of the table, reading a book. Between the participant and the assistant was a large set of everyday objects (rulers, envelopes, pins, etc., as shown in Figure 3). Facing the subject was a laptop, whose screen was not viewable to the assistant. The laptop displayed images of different objects laid out on a grid (as in Figure 4), and the subject could scroll through the images by clicking a button. Subjects were instructed to explain to the assistant how to reproduce each picture. Full participant instructions are available in Appendix D.

The twelve participants in the **Atypical SHAPE** group saw eight pictures (1 practice + 7 test) that included atypical SHAPE stimuli. The twelve participants in the **Atypical MATERIAL** group saw eight pictures (1 practice + 7 test) that included atypical MATERIAL stimuli. Between each trial, objects were put back into their original positions by the experimenter. Subjects were recorded directly to the laptop, and the data was transcribed and anonymised.

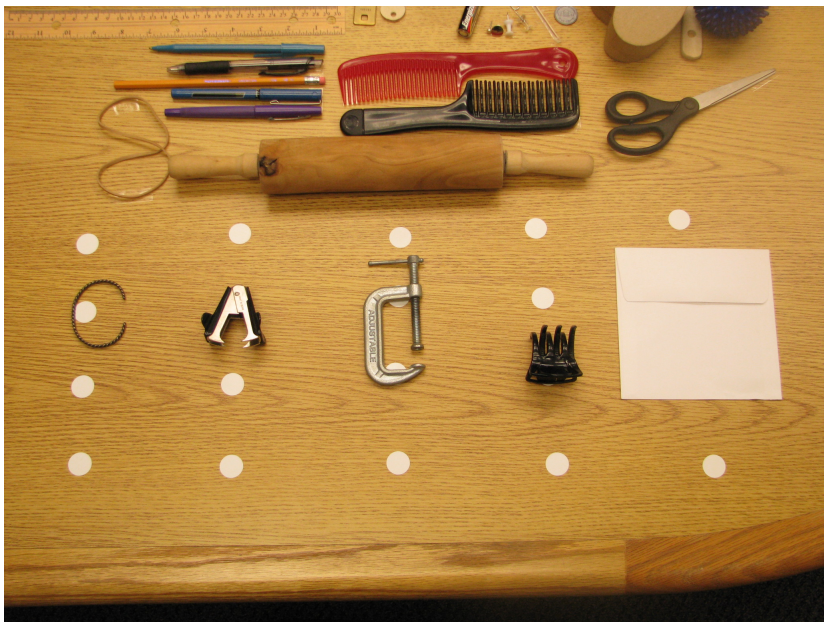


FIGURE 4. Example stimuli, Atypical SHAPE group. Here, the target object is the square envelope.

5.6. Results

5.6.1. Annotation and Outliers. Some example expressions for each of the test objects and some of the fillers are given in Table 5. Example modifiers for the MATERIAL and SHAPE attributes are given in Table 6.

To check for possible outliers, I look for participants who are more than two standard deviations from the mean. In each group, I calculate the average number of references with SHAPE, and the average number of references with MATERIAL. Participants whose total number of references with SHAPE or MATERIAL are more than two standard deviations from the mean for that property are identified as possible outliers. No subjects in

Object	SHAPE GROUP	MATERIAL GROUP
bowl		
	"the brown flower shaped bowl"	"the brown cloth bowl"
box		
	"the heart shaped box"	"the brown square clay box"
envelope		
	"the cd case"	"a white envelope"
key		
	"the key with the square um whatever you call it i-"	"the wooden key"
mug		
	"the um silver cup that's octagon shaped"	"a metal cup"
ruler		
	"a ruler. . . a ruler with geometric shapes"	"the uh the flat ruler"
screw		
	"the black small screw with the flat head to pick it up with"	"a little black screw"
some fillers	"the large silver paper clip" "the uh plastic fork"	"the black comb" "the white thumb tack"

TABLE 5. Example expressions: Test objects and some fillers.

Object	SHAPE	MATERIAL
bowl	"flowery", "flower-shaped"	"felt", "cloth"
box	"heart-shaped", "heart"	"clay"
envelope	"square"	"padded-looking*"
key	"with a square um handle"	"wooden"
mug	"octagonal", "that is not round at the top it's a hexagon"	"metal", "tin"
ruler	"with geometric shapes, that has little holes punched out of it"	"thinner*"
screw	"squished-headed", "with the flat head"	"metal", "flat head*"

TABLE 6. Atypical SHAPE and MATERIAL: Example surface forms. *Material for these objects not mentioned or incorrect.

the Atypical MATERIAL group were identified as possible outliers, and two subjects in the Atypical SHAPE were identified as possible outliers. The data for these two subjects (one male, one female) were removed and replaced with gender-matched data from the held-out set.

5.6.2. Analysis. For each group, I measure whether there is a statistically significant difference in the selection of SHAPE versus MATERIAL modifier. This is measured using a standard paired t-test with object type as a repeated measure. For each participant, I sum the number of test object expressions containing a MATERIAL modifier and the number of test object expressions containing a SHAPE modifier. This provides two paired vectors to compare.

The t-test shows that there is a slight statistically significant difference between the the selection of SHAPE and MATERIAL in the Atypical SHAPE group at $\alpha = .05$ ($t = 3.0268$, $df = 11$, $p = 0.01152$), but not in the Atypical MATERIAL group ($t = -1.8171$, $df = 11$, $p = 0.09651$). In the Atypical SHAPE group, an average of over 4 out of the 7 objects include a SHAPE modifier, and an average of 3 out of 7 objects contain a MATERIAL modifier. In the Atypical MATERIAL group, an average of over 2 out of 7 objects include a SHAPE modifier, while over 3 include a MATERIAL modifier (see Table 7).

This suggests that atypicality affects different attributes differently. People are more likely to pick out *atypical* over *typical* SHAPE properties when referring to objects, but we do not have evidence that this effect is significant for MATERIAL properties.

These findings may reflect that SHAPE is a preferred attribute to MATERIAL; this may cause the selection of an atypical MATERIAL attribute to be about as likely as a typical SHAPE attribute, and explain the difference between the two groups. Further testing is necessary to examine this possibility.

	Atypical SHAPE	**Atypical MATERIAL**
Expressions with SHAPE	4.42	2.33
Expressions with MATERIAL	3.00	3.42

TABLE 7. Average number of object references (out of 7) with SHAPE, MATERIAL modifiers per participant.

Interestingly, when MATERIAL was included in a reference in the Atypical MATERIAL group, it was often incorrect (Figure 5). The plastic screw was called "metal", the paper ruler was called "wooden". A similar tendency to refer to incorrect MATERIAL types emerges in the Atypical SHAPE group, where the ceramic mug painted silver was called "metal" or "steel", the ceramic bowl was called "plastic", and the cardboard box was called "wooden" (see Figure 6). The Atypical SHAPE group was dominated by SHAPE references, however, and those were usually correct; an exception again for the mug, which was called "octagonal" (correct), "hexagonal" (incorrect) and "septuplet" (incorrect).

As can be seen in Figure 5, the ruler in particular in the Atypical MATERIAL group gave rise to incorrect MATERIAL modifiers – it was printed on paper with a wood print, so it was called "wooden". Most participants in the Atypical MATERIAL group did not use MATERIAL modifiers for the envelope and the screw, which may be due in part to the fact that the screw was painted black and so was not clearly plastic; and the envelope was made of foam, which may have not been clear without physically touching the object.

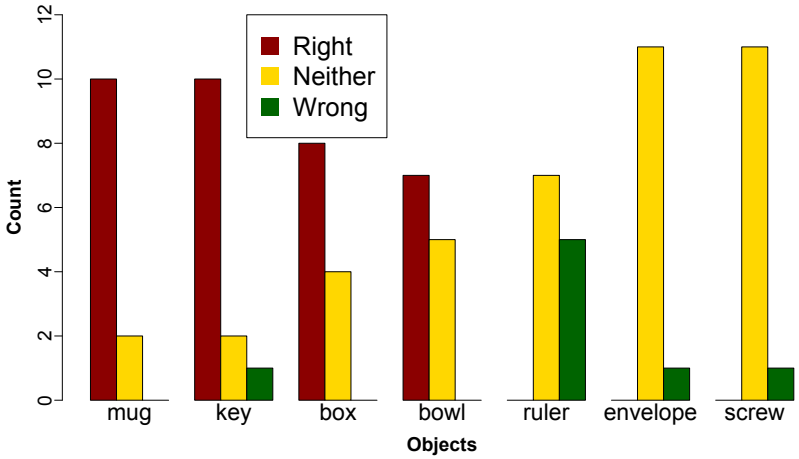


FIGURE 5. **Atypical MATERIAL Group.** Number of participants who included MATERIAL modifiers that were right, wrong, or did not use a MATERIAL modifier at all (neither) for atypical MATERIAL items. The ruler tends to evoke incorrect MATERIAL modifiers. The ruler, envelope, and screw had no correct MATERIAL modifiers at all.

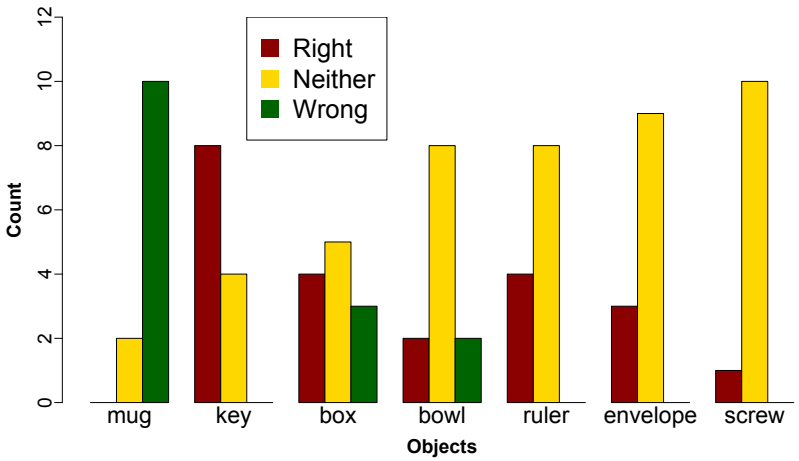


FIGURE 6. **Atypical SHAPE Group**. Number of participants who included MATERIAL modifiers that were right, wrong, or did not use a MATERIAL modifier at all (neither) for atypical SHAPE items. The mug, box, and bowl tended to evoke incorrect MATERIAL modifiers. The mug (painted silver) had no correct MATERIAL modifiers at all.

Some examples of expressions in the Atypical MATERIAL group that do not include MATERIAL modifiers at all are given in Table 7.

Ruler	Envelope	Screw
"the ruler"	"the white envelope"	"the screw"
"ruler that's flatter"	" the uh weird padded looking envelope thing"	"black flat head screw"
"the darker tan ruler"	"long rectangular envelope"	"the screw with the flat head"

FIGURE 7. Examples of references without MATERIAL in the Atypical MATERIAL group. We see underspecified references and references describing the object's size.

Object	SHAPE	MATERIAL	COLOR	SIZE	PART	Other
bowl-material	14.0%	33.0%	**38.0%**			14.0%
bowl-shape	**46.0%**	17.0%	25.0%	13.0%		
box-material	12.0%	32.0%	**36.0%**	8.0%	4.0%	8.0%
box-shape	**58.0%**	37.0%	5.0%			
envelope-material	22.0%	6.0%	**33.0%**	22.0%		17.0%
envelope-shape	25.0%	15.0%	**35.0%**		5.0%	20.0%
key-material	13.0%	**67.0%**	7.0%			13.0%
key-shape	20.0%	**45.0%**	25.0%			10.0%
mug-material	**43.0%**	**43.0%**	4.0%			9.0%
mug-shape	**52.0%**	43.0%	4.0%			
ruler-material	6.0%	**31.0%**	13.0%	**31.0%**		19.0%
ruler-shape	**40.0%**	20.0%	5.0%	5.0%	10.0%	20.0%
screw-material	30.0%	4.0%	**37.0%**	19.0%	4.0%	7.0%
screw-shape	**37.0%**	5.0%	21.0%	16.0%	5.0%	16.0%

TABLE 8. Percentages of different modifier types for objects. Most likely modifier types in bold.

5.6.3. Post-Hoc Analysis. Examining the spread of different modifier types across objects and groups, as shown in Table 8, an interesting trend emerges. Objects are predominantly described with SHAPE, MATERIAL, and COLOR modifiers, but note that those objects in the Atypical MATERIAL group that are not dominated by MATERIAL modifiers are instead dominated by COLOR modifiers. This is shown for bowl, box, envelope, and screw. Interestingly, the objects in this condition that *do* have a high proportion of MATERIAL modifiers – the key, mug, and ruler – were generally referred to with a MATERIAL modifier for which its colors were typical. The key and ruler were

called "wooden",[2] and the mug was called "metal". These objects have correspondingly low proportions of COLOR modifiers.

Those objects with the fewest MATERIAL modifiers in the Atypical MATERIAL group – screw and envelope – are those objects whose material was not visually clear. The envelope was made of foam, and the screw was made of plastic, but colored black. The screw could also be referred to by its subtype, "flat head screw", which some participants picked out (and further speaks to the difficulty of controlling complex properties of real world objects). Not a single person got the materials for these objects correct (see Figure 5), although a couple mentioned an incorrect material, and a few subjects commented that they could not tell what the envelope was made of. For these objects, the colors did not give a clue as to the material; it is possible that MATERIAL was therefore a dispreferred attribute.

The remaining objects, the bowl and the box, have a relatively high proportion of *both* MATERIAL and COLOR modifiers in the Atypical MATERIAL group. For these objects, the material appears to be visually clear and the colors are not particularly tied to the material – both the bowl's material (wool) and the box's material (clay) are dyed rather than being the material's natural color. We therefore may observe relatively high proportions for both of these attributes because the materials are (1) visually clear and (2) atypical, while (3) the colors are not typical for the materials.

The observed interaction between COLOR and MATERIAL is in line with visual work that has pointed out that the two attributes are correlated (Liu, Sharan, Adelson, & Rosenholtz, 2010; Berg et al., 2011). When an object's colors are typical for a MATERIAL, COLOR may be dispreferred and tends not to be produced, while MATERIAL is produced instead. And so we tend to call a wooden bead "wooden" rather than "tan" or "brown". COLOR and MATERIAL are *interconnected*.

[2]As mentioned above, the "wooden" designation was incorrect for the ruler; it was made of paper with a wood-grain print. The pilot studied suggested that a plain paper ruler would be called "white" rather than by its material.

In the Atypical SHAPE group, both the envelope and the key contain fewer SHAPE modifiers than another type of modifier. The atypically square envelope was commonly called "the CD case" (it was not a CD case), a label suggested by the envelope's FORM and SHAPE, and pointing to the interconnected relationship between SHAPE/FORM and TYPE. For the atypically square key, subjects tended to refer to the fact that it was metal – which, in designing the study, I took to be typical of keys – rather than referring to its shape. It may be that its square head was not atypical *enough* to elicit SHAPE modifiers (it was not mentioned in McRae's norms, but may still be relatively typical); it may be that there was an effect from the atypical shape applying to a part of the object (its head) rather than to the whole object; or there may be a general preference for including materials like metal and wood in a description, even when they are typical. Teasing out these details is a clear area for future research.

5.7. Discussion

5.7.1. Findings. This study has suggested that atypicality is a function of the object, the attribute, and the attribute-value. In these tests, we see that adjusting the typicality of the MATERIAL attribute does not show a significant tendency to include material-denoting modifiers, while adjusting the typicality of the SHAPE attribute does show a significant tendency to include shape-denoting modifiers. The reasons for the difference between the two attributes may be due to the fact that SHAPE is preferred overall to MATERIAL. SHAPE may also be more visually salient: MATERIAL is not available pre-attentively and is inefficient for guiding attention (Wolfe & Myers, 2010), while SHAPE may be much more accessible when scanning the scene.

An important issue that I did not address is what is preferred between SHAPE and MATERIAL. If both are equally typical for an object, and SHAPE is preferred, then the results may be explained by the fact that the Atypical SHAPE condition did nothing to discourage the preference for SHAPE, while the Atypical MATERIAL condition did. This still suggests that typicality plays a role, but it is less clear that atypical MATERIAL

is dispreferred; it may be that the preference for SHAPE interacts with a preference to mention an atypical MATERIAL, giving rise to approximately equal distributions of SHAPE and MATERIAL for these kinds of objects.

Another complicating factor is that it seems that MATERIAL is not always visual. Several attributes may be used to determine MATERIAL (COLOR, SHEEN, etc.), and so this appears to be a more complex attribute that may not always be clear without physically handling the object. I find that many of the mentioned materials are incorrect – a silver ceramic mug is called "metal", a wood-printed paper ruler is called "wooden" and a shiny brown ceramic bowl is called "plastic". The white envelope made of foam is not once referred to by this material, but subjects instead pick out its TEXTURE ("weird-textured"), FORM ("padded-looking"), or suggest that it is another object ("the clutch", "eyeglass case"). Similarly, the black screw made of plastic is not once referred to by this material; subjects instead incorrectly refer to the material as "metal", or else refer to it as a more specific subtype of screw, "flat head". Further, there seems to be an interaction between MATERIAL and COLOR, with COLOR modifiers being less common than MATERIAL modifiers for objects whose colors are suggestive of a specific material.

5.7.2. Implications. These findings have interesting implications for a referring expression algorithm. It is not enough to judge whether a visual attribute-value is atypical or not; it must also be judged whether that value is visually clear, and whether other properties suggest another interconnected attribute-value (which may not actually be true of the referent, as when subjects used incorrect modifiers). For SHAPE, we see people using a subtype of the basic level class ("CD case") as well as mentioning the atypical SHAPE. For MATERIAL, if the interconnected attribute of COLOR is not clearly indicating, then a MATERIAL modifier may not be produced, or else produced, but incorrect. On the other hand, if the material is not colored, or is its typical colors, then MATERIAL may be preferred over COLOR.

5.8. Conclusions and Future Work

This study has sampled a handful of objects for referring to the typicality of two attributes, SHAPE and MATERIAL. We see a strong effect for SHAPE, but we do not see a strong effect for MATERIAL. It is still an open question how the findings from this study generalize to other types of objects, or other kinds of attributes. For now, I have developed support for the idea that *in some cases*, atypicality influences what a person will refer to in an object.

The difficulties interpreting these results stem partly from the inherent difficulty of working with real-life objects. Previous work in corpora-building for REG, such as the GRE3D3 Corpus (Viethen & Dale, 2008) or the TUNA Corpus (van Deemter et al., 2006), has largely minimized this issue by choosing simple or deliberately normed objects. In this study, I must make disparate objects as identical as possible for a variety of visual properties, and am limited by my physical abilities to do so.

The benefit of this is that we get to see how people refer to real life objects, presented in-person, which is indispensable in developing an algorithm that generates human-like reference to visible real world objects. This opens up a variety of details that have not yet been researched before, such as the role of *interconnection* between properties and what kinds of properties people tend to pick out when referring to visible objects.

Building on results, it appears that MATERIAL is often tactile as well as visual – many subjects did not get the MATERIAL correct, and their references reflected a misconception based on how the object looked. Perhaps this would have been avoided if we had added tactile sensations to the experiment. It would be interesting to examine this same experiment, but incorporating a tactile as well as a visual modality.

In future work, I hope to examine whether SHAPE is preferred over MATERIAL when all else is equal, and how gradations of atypicality for an attribute affect reference; some values may be more atypical than others, and thus more likely to be included in a final description. It would also be useful to tease out the details of interconnection, and what

this tells us about how people refer in visual domains. The further factor of how cultural notions of typicality affect reference has not been addressed here, but is likely to effect how people refer.

Findings from this study support the idea that a knowledge base with typical attribute-values may be used during referring expression generation to generate human-like reference (discussed in Chapters 2 and 3). In addition to guiding what is remarkable and what is unremarkable, as discussed in this chapter, such a knowledge base could also be used to guide the selection of head noun (TYPE) based on the recognized attributes; and guide the selection of similar objects that the referent object may be compared to, e.g., to create an analogy (Chapter 3).

In Chapter 7, I introduce an algorithm utilizing such a knowledge base to guide the generation of natural reference. Following some of the discoveries in this study, the algorithm treats different attributes differently, with COLOR and MATERIAL processed as interconnected attributes, and atypicality affecting the selection of modifier.

Chapter 6

Color

6.1. Introduction

Although I do not focus on COLOR in this thesis, it is evident that this is a very important property in reference in a visual domain. In each experiment in Chapters 3, 4, 5, COLOR emerged as one of the top properties. This makes sense from the perspective of vision, since COLOR is the property first processed by the visual system, one of the most basic properties in the visual cortex, and a key property for guiding attention when viewing a scene (Treisman & Gelade, 1980; Wolfe & Myers, 2010). This is also in line with previous research that has shown that people favor COLOR when referring (Belke & Meyer, 2002; Sedivy, 2003; Brown-Schmidt & Tanenhaus, 2006; Koolen et al., 2011; Arts, Maes, Noordman, & Jansen, 2011) and often use COLOR redundantly, when a final expression would be equally distinguishing without it (Pechmann, 1989; Viethen, Goudbeek, & Krahmer, 2012). Although I do not run an additional experiment on COLOR, let us at least briefly discuss it, looking at what the literature and the previous work in this thesis tells us, to understand the role it plays in reference to visible objects.

Color receptors are at the forefront of visual perception, preceding later areas that respond to properties such as SIZE and SHAPE (see Chapter 1). There is increasing evidence that COLOR is used in early levels of visual processing to help facilitate shape recognition (Wurm, Legge, Isenberg, & Luebker, 1993; Yip & Sinha, 2002), and an object's color may be an intrinsic component of the visual representation, retained in long-term memory (Naor-Raz, Tarr, & Kersten, 2003) and used to identify objects (Tanaka & Presnell, 1999; Therriault, Yaxley, & Zwaan, 2009). This suggests that a part of the reason why COLOR is so common in referring expressions is that it is useful in low level vision as well

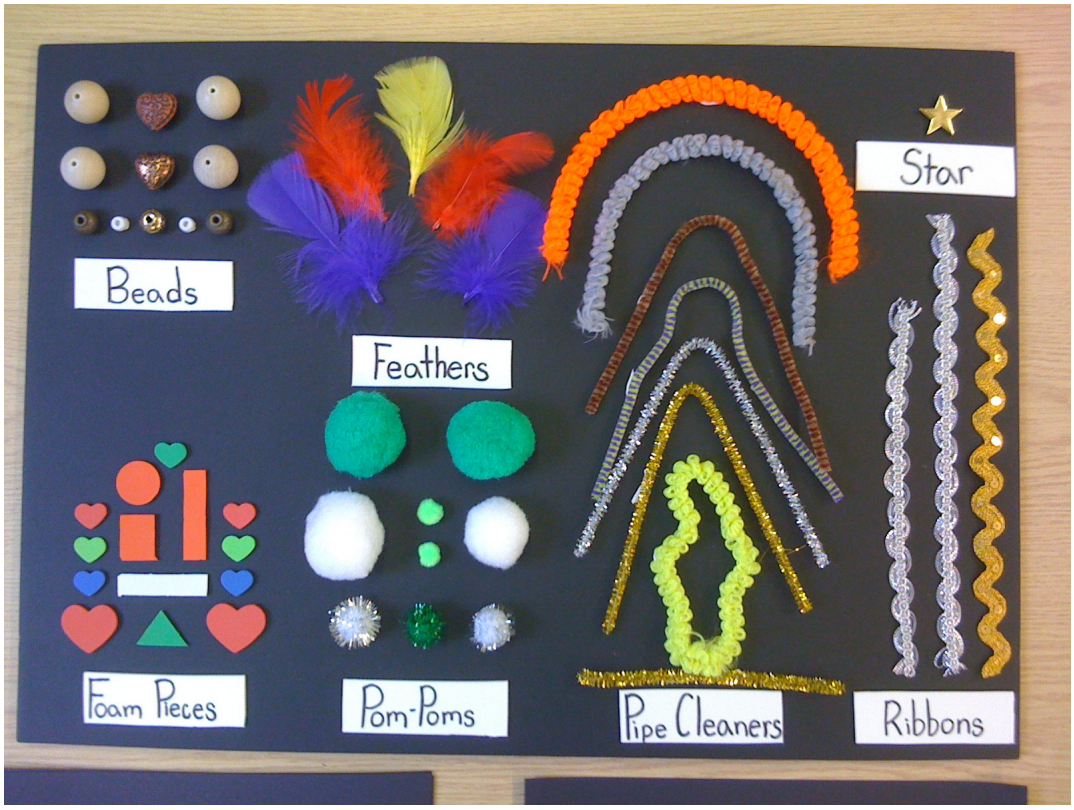


FIGURE 1. Objects used in Craft Corpus.

as higher level object categorization. Not only is COLOR a visually salient property, but an integral part of what we remember about objects.

Previous approaches to referring expression generation have suggested treating COLOR as a special property, placed at the beginning of the Incremental Algorithm's preference order (van Deemter, Gatt, van der Sluis, & Power, 2012) or assigned a cost of 0 in the Graph-Based Algorithm (Viethen et al., 2008). The basic idea behind such approaches is to allow for COLOR to be redundantly included in a final expression. To go a bit further towards understanding how people use COLOR in referring expressions, it is therefore useful to look at when COLOR is *not* used. Below, I go through the corpora I have collected throughout the thesis to examine what this tells us about exceptions to the tendency to include this attribute.

6.2. Craft Corpus

The first corpus I examine is the Craft Corpus from Chapter 3. The objects used in this study are shown in Figure 1. All references to pick out items on the craft board are annotated, represented as visual attribute-value sets (e.g., "COLOR:red TYPE:ball" for "red ball"). The frequencies for each attribute are listed in Table 1, repeated from Chapter 3. As can be seen, COLOR is the dominant attribute that people use, appearing 594 times in 1,842 expressions and more than tripling the next most common attribute, SIZE.

To understand when COLOR is likely not to be included, I look at the distribution of attributes for each referent. For references to singular items, I find that COLOR is consistently not the most common attribute for the wooden bead. For this referent, MATERIAL is more common than COLOR. A possible reason for this is that a MATERIAL modifier like "wooden" conveys more information than a COLOR modifier like "brown". Because wood tends to be brown, COLOR is implied by MATERIAL; the MATERIAL modifier suggests both the MATERIAL and its accompanying typical COLOR. This is again the issue of interconnected properties discussed in Chapter 5, shown in the preference for MATERIAL wood over COLOR brown.

6.3. Size Corpus

This corpus contains expressions elicited to images of object pairs, as discussed in Chapter 4. Object pairs were all rectilinear solids and included books, brownies, boards, and sponges. In all cases, the colors of the two objects were identical; objects were manipulated so that their sizes were different along their x and y axes. This also affected their shape, so that they were either rectangular or square.

This corpus has not been fully annotated for visual attributes other than SIZE. I therefore take the top 100 most frequent words in the corpus and annotate these to gather the list of visual attributes below. Note that because these are annotated without the surrounding

Attribute	Frequency	Example
COLOR	594	red, green, silver, purple, yellow
SIZE	192	big, medium, small, short, thick, long
SHAPE/FORM	156	heart, circle, ball, square, sphere, bent, twisty
TYPE/MATERIAL	94	foam piece
TYPE/SHAPE	89	heart, square, circle, rectangle
MATERIAL	73	foam, wooden, tinsel, plastic, bronze
SHEEN	22	sparkly, glittery, shiny, luminescent
TEXTURE	16	fluffy, fuzzy, furry
ORIENTATION	12	upside-down, horizontal
PATTERN	3	striped, (with a) pattern
LOCATION	1	"at the bottom of the. . . presentation"

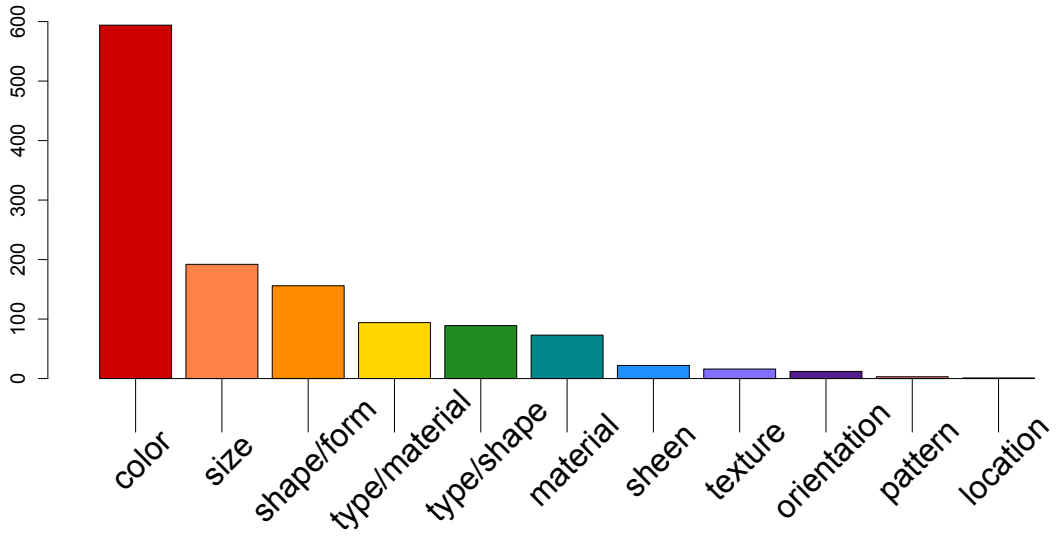


TABLE 1. Craft Corpus attribute frequencies.

context, the value for NUMBER may be inflated; "one" frequently occurs as a head noun, not as a modifier.

We can see that the property of SIZE, an independent variable in the study, and the related property of SHAPE emerge as the most frequent attributes; these are followed by COLOR. In contrast to the previous corpus, COLOR is therefore not the most common property. Instead, we find that in scenes with two objects that have identical colors but differ in size, SIZE emerges as the common attribute. This is in line with findings that

Attribute	Frequency	Example
SIZE	6360	smaller, taller, larger
SHAPE	1571	rectangular, square
COLOR	394	white, black, blue
NUMBER	165	one, two
LOCATION	74	right, top
PROCESS	68	cut, burnt

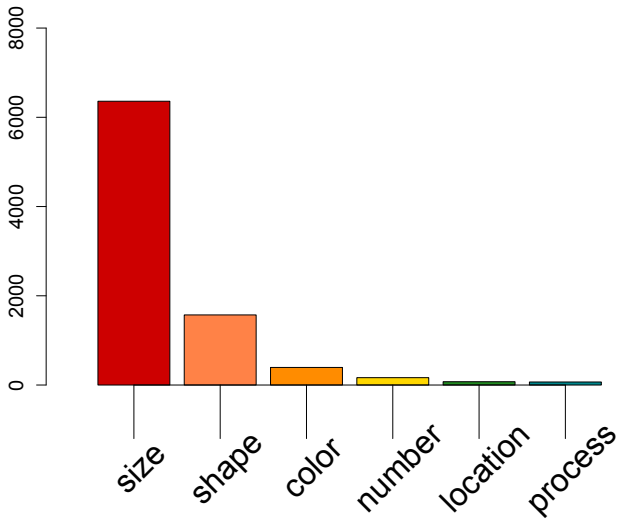


TABLE 2. Size Corpus attribute frequencies.

COLOR is less likely to be used in scenes with relatively little color variation than in scenes with a large amount of color variation (Koolen et al., 2011), and suggests that one factor affecting whether COLOR is selected is a function of the number of objects and the number of object colors.

6.4. Size Corpus Fillers

These are the fillers of the Size Corpus, and include legos, spatulas, and shoes. This corpus has not been fully annotated, and so I again take the top 100 most frequent words in the corpus and annotate these – because the words are annotated without the surrounding context, some values may be inflated. In this corpus, the value for LOCATION seems quite high; *high top* shoes were a stimulus in the fillers, and *top* emerges as a

frequent word, which I annotate as LOCATION. Results are shown in Table 3. We again find that COLOR is the most frequent attribute, occurring more than three times as frequently as the next most frequent attribute SIZE.

Attribute	**Frequency**	**Example**
COLOR	8490	green, blue, black, yellow, white, brown
SIZE	2472	smaller, shorter, taller, bigger
NUMBER	1803	one, two, three
SHAPE/FORM	1540	round, rectangular, slotted
MATERIAL	664	plastic, wooden, lace
LOCATION	630	top, bottom, middle

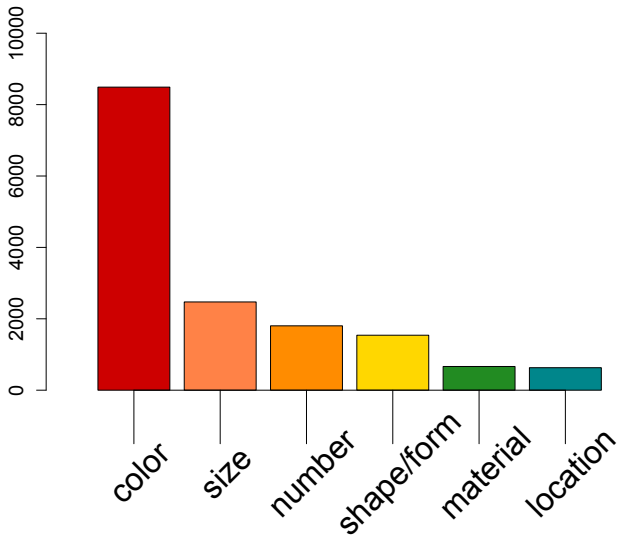


TABLE 3. Size Corpus fillers attribute frequencies.

These findings from the fillers sub-corpus serves as an interesting contrast to the findings from the test stimuli in the Size Corpus. In both sub-corpora, two objects are presented, placed next to one another (with exception to the spatulas, which appeared in groups of three). There is neither a wide range of objects nor a high degree of color variation, and yet we see a similar pattern of results to the highly varied Craft Corpus. Work by Koolen et al. (2011) has shown that scenes with low color variation lead to significantly fewer COLOR modifiers than scenes with high color variation, but in Koolen's study, low variation had *no* COLOR variation: All objects were the same color. With only two

objects of the same type but different visual properties, the marked preference for using COLOR modifiers is evident. Perhaps scenes with low variation – where *some* objects are different colors – will show a large jump in the number of expressions using the COLOR property. Regardless, the pattern of results in the Craft Corpus and the Size Corpus suggests that there is some effect of the contrast set on the selection of COLOR. Whether this is due to visual salience relative to the color of other objects in the scene (the "gist" of the scene and the "pop-out" effect (Treisman & Gelade, 1980); see Chapter 2), direct comparisons with the color of other objects, or some other mechanism entirely, is less clear.

6.5. Typicality Corpus

I again look at attribute frequencies, this time from the Typicality Corpus introduced in Chapter 5. Results are shown in Table 4. The typicality experiment involved manipulations of SHAPE and MATERIAL properties; these are the most frequent in the corpus. Following these, COLOR properties are the next most frequently mentioned.

Examining the object references from the Typicality Corpus, we see that COLOR tends not to be included when the material is metal or wood, in line with the findings from the Craft Corpus discussed above. Here, the scene has relatively diverse objects with a variety of colors, and yet we do *not* see that COLOR is the most common property. Comparing this with the Size Corpus fillers above, we see that when the scene has just a few objects of the same type, differing in several properties including COLOR, COLOR is preferred; when the scene has many objects of different types, but objects of the same type do not have different colors, COLOR is less preferred.

One way to explain this trend is to posit that COLOR is chosen partially by direct comparison processes, specifically comparing a target object against other objects of the same category. This suggests that the selection of COLOR may be affected by both the overall color variation of the scene as well as the contrast with objects of the same type.

Attribute	Frequency	Example
SHAPE/FORM	95	hexagonal, circular
MATERIAL	87	metal, wooden
COLOR	68	black, white, brown
PART-WHOLE	40	with a lid, with the blade
SIZE	23	thin, thick, skinny, little
USE/PROCESS	17	carved, molded
INTENSITY	10	darker, lighter
ANALOGY	9	. . . like a butterfly
TEXTURE	3	hairy, fuzzy
LOCATION	3	here, . . . on top
SUBJECTIVE	2	plain, weird

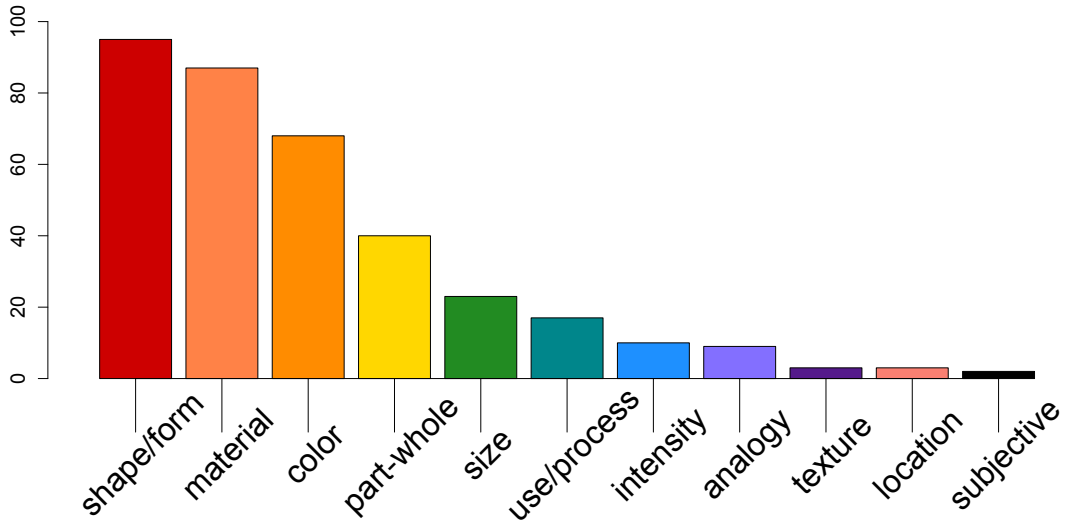


TABLE 4. Typicality Corpus attribute frequencies.

This corpus also provides further evidence that COLOR is interconnected to MATERIAL, particularly when its value is *metal* or *wood*.

6.6. Conclusions

In the highly diverse Craft Corpus and the relatively uniform fillers sub-corpus of the Size Corpus, we found that COLOR was exceedingly preferred. In the even more uniform Size Corpus and the Typicality Corpus, we found less evidence that COLOR is a preferred

attribute, instead seeing a predominance of the attributes being studied in building each corpus. The commonality between the Size Corpus and the Typicality Corpus is that both had objects of the *same type* with the *same color*, while objects in the Craft Corpus and the fillers sub-corpus of the Size Corpus had objects of the *same type* with *different colors*. This suggests that in addition to the pre-attentive role that COLOR may have in visual reference, comparison processes of a target object against another of the same type may also play role.

A clear effect on the selection of COLOR when referring to real world objects is the influence of *interconnection*: Some natural materials, like metal and wood, are inextricably linked to their typical color, and in these cases MATERIAL rather than COLOR is often used to refer to the object. Indeed, the interconnection between COLOR and MATERIAL is reflected in our lexicon, with words like "copper", "silver", and "gold" used for both a material and a corresponding color. It seems reasonable for an REG algorithm to factor in what an object's color may be interconnected to, specifically for MATERIAL properties.

We also see some trends that preliminarily suggest that both comparison processes and overall "gist" knowledge takes place when naming objects in a scene and referring to their color. This is well-supported by visual evidence that shows that we use color to guide attention (Wolfe & Myers, 2010) when looking at a scene pre-attentively, and highly contrasting colors can "pop-out" in their environment (Treisman & Gelade, 1980). To apply this to an algorithm, an initial pre-attentive, "gist"-based representation of the scene that affects the selection of different properties could precede more fine-grained comparison processes, contrasting the target object against other objects of the same type.

The ideas that COLOR is a salient property, interconnection plays a role particularly for COLOR and MATERIAL:metal/MATERIAL:wood, and at least two separate processes are involved in naming, one based on "gist" and one based on direct comparisons, are applied in the algorithm introduced in Chapter 7.

Chapter 7

The Visible Objects Algorithm

7.1. Introduction

In this chapter, I give the problem of generating natural reference to visible objects center-stage, proposing a process that creates semantic structures for object description assuming a correct visual input. This input provides evidence for a wide range of visual properties, including height and width features for the detected object. This separates the problem of vision from the problem of language, allowing us to address two critical issues in generating human-like language from a visual input as computer vision research advances: (1) Specifically what visual input should aim to provide, given the current state of the art; and (2) Methods for generating human-like reference with a perfect visual input.

7.2. Generating Human-Like Reference

Previous algorithms in referring expression generation have largely been deterministic, with a primary goal of creating one expression type to uniquely identify a referent. Throughout this thesis, I have questioned determinism and the goal of unique identification in generating human-like language. I have also discussed how speakers produce several different expressions for the same referent, and how their reference may be *descriptive*, including properties because they are visually prominent – like color and size – rather than because they distinguish the referent from a set of competitor objects. This type of reference is similar to the conversational or verbal account of reference rather than the literary account of reference common to earlier work. Having now set the stage, in this chapter I propose alternative strategies for generating reference. The algorithm I

FIGURE 1. Speaker variation is a large part of referring expression generation. For example, sixty-eight subjects referred to this object in thirty-four different ways (Mitchell, 2008).

introduce uses non-determinism to capture speaker variation, and operates with a goal of describing what is salient about the referent object following ideas from visual processing. Before turning to a larger discussion of the algorithm, I will briefly summarize these points.

7.2.1. Non-Determinism and Speaker Variation. In earlier work (2008), I collected 68 references to a ball that was orange and bumpy (see Figure 1). By letting participants refer to the object however they would like, there were a total of 34 distinct expressions – an average of one unique reference for every two people. However, these were not normally distributed; 13 participants said *red ball*, 6 said *ball*, 3 said *orange bumpy ball* and (for example) 1 said *dog toy*. With this sort of variation, making generalizations about reference in any domain is difficult. We still do not know when properties are included in a description independently of one another, what scene-specific, task-specific, and person-specific aspects result in particular properties being chosen, and what factors are involved in determining whether a description is satisfactory. As discussed in Viethen and Dale (2009), much of the data on referring expression generation does not warrant very strong claims about the nature of referring expression generation at all. What is clear is that people generate different expressions for the same object, but have certain preferences, such as including color modifiers. In this algorithm, we begin to capture this variation for the first time.

A possible way to model this is to make an algorithm *non-deterministic*, generating structures for a wide range of expressions, and giving preference to those properties known

to be preferred by people. This approach is taken in this algorithm, where different attribute sets are stochastically generated based on prior likelihoods (further detailed below). With this method, different referring expressions may be generated with varying probabilities.

7.2.2. Salient Properties, Overspecification, and Underspecification. Approaches to referring expression generation usually focus on *uniquely identifying* the referent, including properties that rule out all other distractors. However, we do not have evidence that this is what people do. In fact, the tendencies for overspecification and redundancy – where properties that are not distinguishing are included (Sonnenschein, 1985; Pechmann, 1989; Koolen, Gatt, Goodbeek, & Krahmer, 2009) (see Figure 2, and Chapter 3 Table 3 for a further discussion) – and the phenomenon of underspecification – where speakers fail to uniquely identify the referent (H. H. Clark et al., 1983; H. H. Clark & Wilkes-Gibbs, 1986; Viethen & Dale, 2008, see the study from Chapter 5) – suggest that ruling out distractors is *not* what people do.

An alternative idea is that properties are selected because they are salient for the speaker (Horton & Keysar, 1996; Bard, Hill, Arai, & Foster, 2009). Such tendencies have been attested in dialogue, but have not been thoroughly examined in monologue (but see Chapter 3); when generating initial reference in conversation, speakers may not spend a great amount of cognitive effort considering the perception of the hearer (Keysar & Henly, 2002), and may "blurt out" their reference (Ferreira & Swets, 2002) before they have even begun scanning the alternatives (Pechmann, 1989). This suggests that a reference generation algorithm that aims to produce natural initial reference should not try to find an optimal subset of properties, but to model what speakers find important when first introducing an object into discourse. In particular, we know that COLOR is a salient attribute in many cases, interacting with MATERIAL, and that SIZE also plays an important role. Further properties may be selected based on prior likelihoods from

a corpus of reference to visible objects – what we have learned as common to mention from past experiences.

7.2.3. Parallel Processing. We know that the visual system contains two distinct cortical pathways operating in parallel, the so-called "what" and "where" pathways (Mishkin et al., 1983). The dorsal ("where") pathway processes locations, sizes, distances, orientations, and spatial properties in three-dimensional space, while the ventral ("what") pathway detects edges, regions of common color, texture, and geometric properties (Kosslyn, 1994).

It is conceivable that using a similar distinction between two parallel pathways may be useful for generating natural language. My underlying assumption is that visual perception and language production may be seen as connected, with the separate visual pathways influencing pathways for language production. As such, we can separate "where" properties – size, orientation, location, etc. – from "what" properties like color, shape, and material during generation. With such a structure, there may be competition between two parallel pathways, where different properties are more likely to be added the more quickly they are processed; because properties are added partially as a function of the length of the expression constructed so far, the speed at which each property is processed in the two parallel paths may affect the final surface form.

The algorithm introduced in this chapter is developed to address these issues, incorporating the non-deterministic size algorithm introduced in Chapter 4 and bringing together the different mechanisms that I have proposed to be underlying different visual descriptors. This includes a knowledge base of typical properties (discussed in Chapters 3 and 5), and separate processes for absolute and relative properties (discussed in Chapters 2, 4, and 6). Likelihood estimates that are used to determine what expression is generated may be learned from a corpus of descriptive expressions, in order to model general human tendencies in describing objects; it may also be learned from a corpus from a single person, in order to model a single person's referential tendencies.

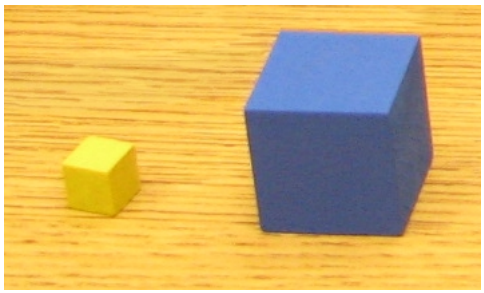


FIGURE 2. One cube, Two cube, Big cube, Blue cube? Many speakers would call the object on the right a *big blue cube*, although either *big* or *blue* can clearly and uniquely identify the referent.

Before continuing, a comment on notation: The algorithm uses *properties*, *attributes*, and *values*. A visual *property* is an attribute-value pair. An *attribute* is a general visual class like COLOR; a *value* is the value for that class, like `red`. A property may therefore be COLOR-`red`, made up of the attribute COLOR with the value `red`.

7.3. Attributes Considered

The algorithm introduced in this chapter first establishes a *semantic value* for an attribute. This value is derived given a multidimensional feature vector for the attribute. For example, given the COLOR attribute and a feature vector representing the RGB values of all the pixels in a target object, a single RGB *semantic value* may be selected to represent the overall color of the object, such as <217, 70, 0>.[1] Semantic values are used to create *lemmas*, representations of the semantic properties of a word without any phonological information specified (Levelt, 1989; Schriefers, 1992; Bock & Levelt, 1994) (see Chapter 2). In our example, <217, 70, 0> could be used to create a lemma for "red" if the object is hair, or a lemma for "orange" if the object is juice.

I make a broad distinction between two kinds of properties, simple and complex. *Simple* properties (SP) include the absolute property of COLOR (with associated values) and the relative properties of SIZE, LOCATION, and ORIENTATION (with associated values).

[1]This color:

These correspond to those properties that are first processed by the visual system (cortical areas V1 and V2) and help guide attention (Wolfe & Myers, 2010, see Chapter 1 for review). *Complex* properties (CP) are those hypothesized to be analyzed in the ventral stream ("what pathway") and include MATERIAL, TEXTURE, PATTERN, and other absolute properties.

I establish such a separation in order for the algorithm to process these two kinds of properties differently. This mirrors activity in the visual pathway, where simple properties feed forward to more complex properties (Mishkin et al., 1983; Riesenhuber & Poggio, 1999; Itti & Koch, 2001), and follows works in computer vision, where recognizers for different visual attributes such as material are built using low level features such as color and edge orientations (Farhadi et al., 2009; Kulkarni et al., 2011).

In the proposed algorithm, simple properties are analyzed first, beginning with the simultaneous processing of COLOR and SIZE. My hypothesis is that the first properties to be cognized in the visual system are likely to be the first considered in an expression for a visible object. This is a possible explanation of why COLOR is so common in referring expressions (see Chapter 3, 4, 5); it is perhaps not a coincidence that color is the first property discriminated by the visual system (see Chapter 1).

A list of visual properties intended for this algorithm is available in Table 1. The list is not an exhaustive list of visual properties, and excludes, e.g, COUNT, FLICKER, MOTION, more complicated forms of location (e.g., "on top of X"), etc. A few of the listed properties are non-obvious and require some explanation: FORM-SHAPE is intended for forms composed of smaller shapes, such as "spiky" (composed of spikes) or its converse, "smooth"; FORM-OBJECTS is intended to be forms composed of smaller objects, e.g. "hairy" (bits of hair) or "feathery" (with feathers); OTHER SENSE corresponds to other senses, e.g., an object's FEEL may be "soft", and this may be used in a visual description. USE/PROCESS is a category for properties that are derived from a process (e.g., "braided") or speak to the use of the object (e.g., "empty", denoting that the object is used to hold something).

Attribute	Example Surface Form
SIMPLE PROPERTIES	
COLOR	"red", "dark"
SIZE	"big"
ORIENTATION	"sideways"
LOCATION	"on the right"
COMPLEX PROPERTIES	
MATERIAL	"ceramic"
OPACITY	"transparent"
SHAPE	"square"
PATTERN	"striped", "speckled"
SHEEN	"shiny"
SUBJECTIVE	"pretty"
FORM-SHAPE	"flat", "spiky", "smooth"
FORM-OBJECTS	"hairy", "feathery"
OTHER SENSE	"soft"
USE/PROCESS	"open", "braided", "empty"

TABLE 1. List of visual properties under consideration.

As mentioned above, each visual property is intended to be a feature vector. For example, LOCATION may be represented as x, y, z coordinate features; and SIZE, the one attribute I have addressed in great detail, may be represented as height and width features (see Chapter 4). This allows the algorithm to create semantic attribute-values based on several interacting features at once. For example, <over, +> for SIZE (Chapter 4). Once created, each semantic value may be selected to become a lemma, and so included in the final description. For example, <over, +> may become the lemma for surface forms like "large" and "big".

Since head noun selection can be chosen from a visual property – people speak to the red *square* or piece of *foam* (SHAPE and MATERIAL properties, respectively: See Chapter 3, Section 3.6.2), the algorithm creates the lemma for the object head noun last. This allows the flexibility to create a head noun from a previously selected property in future work.

7.4. Main Ideas

Given an object and a scene, the algorithm constructs an identifying description. I define an *object* to be a single intended inanimate referent. I define a *scene* to be a visual representation of objects within the focus of attention. I define an *identifying description* to be a description to identify an object, which may or may not uniquely distinguish it. To the best of my current understanding, an identifying description corresponds to Donnellan's (1966) *referring expression*, Searle's (1969) *identifying description*, Appelt's (1985) expressions with *identification intention*, and Clark and Bangerter's (2004) *referring* (see Chapter 2).

The goal behind the algorithm is to identify the referent with human-like variation using its known properties; the goal is not to uniquely identify the referent. Some of the basic ideas behind the algorithm are as follows:

(1) Inclusion of an attribute is a function of description length and prior likelihood for the attribute.
(2) The final expression is not produced deterministically.
(3) Atypicality has an effect on property inclusion.
(4) Some attributes are interconnected.
(5) Both parallel and incremental processes can be used.

Ideas (1) and (2) follow from the fact that people occasionally underspecify or overspecify when they refer, either including more conceptual information than necessary to identify a referent or else failing to uniquely identify the referent upon initial reference. By using likelihood values to non-deterministically add properties, I aim to create an algorithm that well approximates natural human variation. Including attributes as a function of how many attributes have already been selected follows the finding that people rarely include more than three modifiers in a noun phrase, and are much more likely to include two (or fewer) (Mitchell, Dunlop, & Roark, 2011; Berg et al., 2011). This is a way

of implementing the cognitive load that may affect how many properties people use to describe an object.

I define likelihood estimates in the variables α_x, β_y, δ_y and γ, where x represents an attribute and y represents a property (attribute-value pair). α_x represents the prior likelihood of including the attribute x. β_y represents the typicality of property (attribute-value pair) y for the referent. γ is a penalty function given the length of the constructed identifying description r. δ_y is a measure of bottom-up visual salience of the property y, and is not currently implemented; all objects are taken to be equally visually salient. In following work, I aim to tune these estimates more precisely using, e.g., expectation maximization.

For now, I begin by defining the stochastic function that determines whether a lemma is included in the identifying description. Given an incomplete identifying description d and a property y with attribute x, the function returns the probability of adding a lemma for the property given d, $p(y \rightarrow lemma|d)$. This is calculated as the linguistic and visual salience of the property $s_{x,y}$ times a length penalty γ, plus the atypicality of $s_{x,y}$. I define $s_{x,y}$ to be the prior likelihood of attribute x (α_x) multiplied by the visual-salience of the property y (δ_y):

$$s_{x,y} = \alpha_x \times \delta_y \tag{7.1}$$

Typicality values for β_y increase the more typical a property is; because I want to make less typical things more likely to be mentioned, I add the atypicality measure (1 - β_y) weighted by the remainder of the probability space:

$$(1 - \beta_y) \times (1 - (s_{x,y})) \tag{7.2}$$

The penalty based on description length γ should be inversely proportional to the length of the description, and so I define γ as $\frac{1}{|d|}$ multiplied by some constant g, and multiply this into the linguistic and visual salience:

$$s_{x,y} \times \gamma \tag{7.3}$$

This leaves the following stochastic function:

$$p(y \rightarrow lemma|d) = s_{x,y} \times \gamma + (1 - \beta_y) \times (1 - (s_{x,y})) \tag{7.4}$$

$$= \alpha_x \times \delta_y \times \gamma + (1 - \beta_y) \times (1 - (\alpha_x \times \delta_y)) \tag{7.5}$$

A lemma for a property (attribute-value) y is therefore added as a function of the linguistic salience of its attribute (the relative frequency with which the attribute is included in descriptions), the property's visual salience (not currently implemented), and the property's atypicality for the object. The current formulation is admittedly crude. I am here detailing the broader ideas I am working towards; in evaluation (Chapter 8), I use a simpler approach, without a δ_y measure of bottom-up visual salience:

$$p(y \rightarrow lemma|d) = \alpha_x \times \gamma + (1 - \beta_y) \times (1 - \alpha_x) \tag{7.6}$$

Without typicality values available (β_y) (true in some conditions of the evaluation), this becomes:

$$p(y \rightarrow lemma|d) = \alpha_x \times \gamma \tag{7.7}$$

Both functions 7.6 and 7.7 are used for evaluation in Chapter 8. If these initial approaches are promising, I hope to further develop functions using these features.

Ideas (3) and (4) above follow the findings in the typicality study in Chapter 5. (3) is only weakly supported for SHAPE; we did not find a significant effect of typicality for MATERIAL. Therefore, in evaluation (Chapter 8), I examine a version of the algorithm that does not take typicality into account. (4) became evident while constructing and piloting the study in Chapter 5: visual properties can be *interconnected* with other properties. Shape entails typical forms, material entails typical colors. Given an object made out of tin, one can usually assume the object will be silver. That is, the MATERIAL attribute *tin*

entails the COLOR attribute *silver*, or MATERIAL(*tin*) → COLOR(*silver*). People appear to use a false inference for these kinds of interconnections – a heuristic – for example, calling a mug that is silver "metal" (even though it is ceramic). Similar interconnections hold for MATERIAL/OPACITY and SHAPE/FORM (see Chapter 5). To my knowledge, this tendency has not yet been addressed in work in referring expression generation. A preliminary list of attributes and interconnected values (represented as words) is given in Table 2.

Idea (5) is inspired by hypotheses for visual processing that suggest there is a ventral ("what") stream operating in parallel to a dorsal ("where") stream. The idea of parallel processing in language production is not new. Tests of language production processes have found that speakers speak and plan simultaneously, using both a *horizontal* and *vertical* aspect, with various levels operating in parallel as well as sequentially (Ferreira & Swets, 2002). Lexical access to the lemmas of content words of a noun phrase may proceed in parallel, although access to the noun may take longer than access to the adjective (Schriefers, 1992). Applying this to the algorithm, there may be competition between different properties in the separate pathways, such that those processed first are more likely to be generated; because there is a length penalty, properties that take longer to process are less likely to be added. For this algorithm, I do not handle differences in speed (I assume that all properties are processed equally quickly, and this is especially important for COLOR and SIZE, which are both generated simultaneously without length penalty), but in future work implementing different speed differentials for different properties may lead to further human-like variation.

The implementation of incrementality in this algorithm follows more directly work in Pechmann (1989), but is a different view than that taken in the Incremental Algorithm (Dale & Reiter, 1995); the incremental process here operates over attributes and objects rather than operating over object attributes alone (see Chapter 2, Sections 2.2.2 and 2.2.3).

Attribute	Interconnected	Example
COLOR	MATERIAL	material(*tin*) → color(*silver*)
OPACITY	MATERIAL	material(*glass*) → opacity(*clear*)
SHEEN	MATERIAL	material(*aluminum*) → sheen(*shiny*)
FEEL	FORM	form(*hairy*) → feel(*soft*)
FORM	SHAPE	shape(*star*) → form(*with spikes*)

TABLE 2. Example interconnected attributes.

As a final detail, the size algorithm introduced in Chapter 4 is called directly, as a function within the full algorithm. This is used to determine whether generating a SIZE modifier is appropriate, and if so, which axes it should pick out.

7.5. The Algorithm

The algorithm constructs an identifying description, a description to identify a referent for a hearer in a verbal (as opposed to literary) setting (see Chapter 2), given an object and a scene that both speaker and hearer can see. The algorithm gives priority to properties that are visually and linguistically salient, to non-deterministically generate descriptive initial reference. From a computer vision input, an object can be a set of pixels with RGB values and (x, y, height, width) coordinates in an image, corresponding to a single inanimate referent. A scene is a set of such objects within an image.

7.5.1. Assumptions. The algorithm requires the following functionality and system knowledge.

Provided by System:

- α_x, a prior likelihood for the inclusion of attribute x.
- β_{val}, a measure of the typicality of an attribute-value val for a given object category returned from find_category.

- δ_{val}, a measure of bottom-up visual saliency or "pop-out" of the given attribute-value *val* (not currently implemented in evaluation).

- g, a parameter set at run-time specifying the weight to give to the description's length in the stochastic decision process. Used to create γ. Default set to 5.

- CP, an ordered list of complex properties (CP) accessible at all times; attributes are available from Table 1. In evaluation, ordered by corpus frequencies.

- SP, an unordered list of simple properties (SP) accessible at all times; attributes also available from Table 1. In evaluation, ordered by corpus frequencies.

- fixate(scene, obj), a function that returns a visual representation of the object obj in scene. This is represented as a set of visual properties each associated to a multi-featured vector of features, e.g., histogram values for luminance and hue (see Figure 4).

- scene, a list of objects in the visual scene under consideration, where each object is represented as a set of attributes, and each attribute associated to a vector of visual features for that attribute (such as hue, luminance and saturation values for the attribute COLOR). The order of objects in the list will correspond to the order in which they are fixated.

- do_attribute(obj, scene, att), a function that returns a value for an attribute given the visual characteristics of the object, calling different processes for each attribute.

  For example, for SIZE, this would be the output of the algorithm introduced in Chapter 4, e.g., <over, +> (see Figure 6).

- lemma(att, val, category), a function that returns a linguistic specification of the word or phrase to be generated for an attribute-value, optionally given the category of the object.

For example, for the SIZE value <over, +>, this could return "big" with a prenominal specification (see Figure 7). If available, object category may be used to help determine the lemma (for example, color lemmas may depend on the object type).

- lemma(att, val, category, dval), same as above, but may additionally create lemmas based on comparison to a comparator object value dval (for example, to create "the one that is not X").

- KB.find_category(obj), a knowledge base function that returns a propositional representation of the range of typical attribute-values for the given object obj along with its category.
  For example, if kept very simple and semantic values are represented as words, this may provide an object representation from McRae's norms (see Figure 5). Once accessed, this is available at all times.

- KB.interconnected(att), a knowledge base function that returns a list of the typical interconnected attributes for a given attribute att.
  For example, for the attribute COLOR, this would return MATERIAL (see Table 2).

- KB.implies(att, val, i_att, ival), a knowledge base function that checks if an interconnected attribute-value is implied by the current attribute-value.
  For example, using simplified attribute-values (not RGB values), COLOR *tan* implies MATERIAL *wood*.

The algorithm itself defines values with the following variables and functions:

Functions and Variables Defined by Algorithm:

- r, a list that stores the *identifying description* being created. Each member of the list is a lemma for an attribute-value.

- γ, a penalty for the inclusion of a lemma given the length of the expression constructed so far.
- att; i_att, an attribute such as COLOR, SIZE, etc.
- val; ival; dval semantic values for a given attribute, such as <217, 70, 0> for COLOR. These are simplified in evaluation to just be, e.g., *red*.
- known_attributes, a dictionary accessible at all times, which stores attributes and values as they become known in the algorithm.
- refer(obj, scene), a high-level function that calls to functions to find the object category, create lemmas for the identifying description, and returns the final identifying description.
- analyze_SP(obj, scene, r), a function that analyzes COLOR and other simple properties that the COLOR value implies (such as MATERIAL values) in parallel with SIZE, LOCATION, and ORIENTATION.
- check_interconn(obj, s, att, val, r), a function that searches for further attribute-values of the referent r that are implied by the given attribute att and value val. For example, if COLOR is *tan*, then MATERIAL *wood* will be found.
- add_lemma(att, val, length(r)), a function that stochastically adds a lemma for the given attribute-value, using α_{att}, cat.β_{val} if available, and the length of the description constructed so far.
- add_lemma(att, val, length(r), dval), same as above, but used near the end of the algorithm, when the target object is compared to other comparator objects in the scene. This allows the function to additionally use the comparator object's attribute-value (dval) to determine the lemma – for example, generating "not the red one".

- analyze_CP(obj, scene, r), a function that analyzes complex attributes, such as MATERIAL, TEXTURE, etc. (See Table 1.)
- incremental_obj(obj, scene, r), a function that incrementally scans the objects in the scene, finding objects of the same type that have different attribute-values than the target referent; these may then be added to the identifying description in the add_lemma function.
- throw_dice(α_{att}, cat.β_{val}, δ_{val}, len), the stochastic function that determines whether a lemma should be created, and discussed in greater detail in Section 7.4. Uses α_{att}, γ (a penalty based on the description's length), cat.β_{val} if typicality is being analyzed (otherwise this value is set to 1.0), and δ_{val} (in the current implementation, this is also 1.0 across the board).

7.5.2. Pseudocode. Below is the pseudocode for the algorithm in Figure 3, starting with **func refer**. The full identifying description is stored in r.

7.5.3. Inputs and Outputs. I assume the presence of some main function that provides functionality for fixating on objects, and given a particular target referent object (indexed by, e.g., an object ID), calls refer. The previous fixation function should return the visual representation of this object, represented as a bounding box with size (height, width) and minimum coordinates (x, y). Following visual processing, we can characterize the region of the object by the RGB values of its pixels and the distribution of hue h, luminance l, saturation s. Corners c, and edges e, as detected by, e.g., a canny edge detector, may also be represented here. I require that this function return an *obj* object, with data structures corresponding to these features. An example is given below for a hypothetical function fixate (Figure 4).

Once the object has a basic visual representation from this function, refer begins the process of generating an identifying description. Throughout this process, lemmas are

```
01 func refer(obj, scene):
02   r = <>
03   // Parallel process 1
04   cat = KB.find_category(obj)
05   // Parallel process 2
06   r = analyze_SP(obj, scene, r)
07   r = analyze_CP(obj, scene, r)
08   // End parallel processes
09   r = incremental_obj(obj, scene, r)
10   r += <cat.type>
11   return r

12 func analyze_SP(obj, scene, r):
13   // Parallel process 1
14   att = 'COLOR'
15   val = do_attribute(obj, scene, att)
16   known_attributes[att] = val
17   r += check_interconn(obj, scene, att, val, r)
18   r += add_lemma(att, val, length(r))
19   // Parallel process 2
20   for att ∈ <'SIZE', 'LOCATION', 'ORIENTATION'>:
21     val = do_attribute(obj, scene, att)
22     known_attributes[att] = val
23     r += add_lemma(att, val, length(r))
24   return r

25 func check_interconn(obj, s, att, val, r):
26   i = <>
27   for i_att of KB.interconnected(att):
28     ival = do_attribute(obj, s, i_att)
29     known_attributes[i_att] = ival
30     if KB.implies(att, val, i_att, ival):
31       i += add_lemma(i_att, ival, length(r))
32   return i

33 func analyze_CP(obj, scene, r):
34   for att ∈ CP:
35     if att ∉ known_attributes:
36       val = do_attribute(obj, scene, att)
37       known_attributes[att] = val
38     else:
39       val = known_attributes[att]
40     r += check_interconn(obj, scene, att, val, r)
41     r += add_lemma(att, val, length(r))
42   return r

43 func incremental_obj(obj, scene, r):
44   for d in scene:
45     dobj = fixate(scene, d)
46     dcat = find_category(dobj)
47     if dcat.type == cat.type:
48       for att ∈ CP ∪ SP:
49         dval = do_attribute(dobj, scene, att)
50         val = known_attributes[att]
51         if dval != val:
52           l = add_lemma(att, val, length(r), dval)
53           if l not in r:
54             r += l
55   return r

56 func add_lemma(att, val, len, dval=None):
57   l = <>
58   if cat:
59     if dval:
60       l = lemma(att, val, cat.type, dval)
61     else:
62       if throw_dice(α_att, cat.β_val, δ_val, len):
63         l = lemma(att, val, cat.type)
64   else:
65     if throw_dice(α_att, 1.0, δ_val, len):
66       l = lemma(att, val)
67   return l

68 func throw_dice(α_x, β_y, δ_y, len):
69   if len == 0:
70     γ = 1.0
71   else:
72     γ = 1/(len * g)
73   weight_function = α_x * δ_y * γ + (1-β_y) * (1 - (α_x * δ_y))
74   n = random number between 0 and 1
75   if n < weight_function:
76     return True
77   return False
```

FIGURE 3. Algorithm for generating identifying descriptions. The bottom function **throw_dice** represents the stochastic decision process used throughout the algorithm to decide whether or not to add a lemma to the description.

$$\begin{array}{ll} \textsf{fixate} \text{ input:} & \text{image} \\ & \text{bounding box:} \quad \begin{array}{cc} size & coordinates \\ \left[63, 63\right] & \left[1, 10\right] \end{array} \\ \textsf{fixate} \text{ output:} & obj \\ & \begin{bmatrix} hue : h \\ luminance : l \\ saturation : s \\ corners : c \\ edges : e \\ RGBpixels : r \end{bmatrix} \end{array}$$

FIGURE 4. Function fixate input and output.

created, and so can be associated to words and generated on-the-fly; such an extension would allow the process to be interrupted, e.g., by a hearer in a dialogue.

The function refer runs two parallel processes. In the first process (line 04), find_category returns a stored representation cat of what this object is and what it typically looks like.[2] If a known category exists, this supplies a type (cat.type) that may be used to realize a head noun for the object, and a list of typical properties for the object. If it does not exist, cat is False. In this case, a placeholder word (e.g., "thingie") may be generated. An example category is given below in Figure 5, taken from McRae's norms.

Attribute	**Value**
TYPE:	bowl
COLOR:	different colors
SHAPE:	round
MATERIAL:	plastic, ceramic
FORM:	curved
USES:	eating, soup, food, liquids, eating cereal, holding things, mixing
FOUND IN:	kitchen
ASSOCIATED WITH:	spoon

FIGURE 5. Example category for bowl. Adapted from McRae's norms (McRae, 2011).

[2]This is similar to Rosch's (1976) notion of a prototype and Wu and Barsalou's (2009) notion of a situated concept, although I do not expect this to directly correspond to any cognitive model.

In the second parallel process (lines 06–07), visual attributes of the object begin to be associated to semantic values and non-deterministically added to the identifying description r as lemmas. First simple properties are analyzed in analyze_SP, then complex properties in analyze_CP.

Function analyze_SP takes as input a vector-based representation of the objects in the scene and the ID of the target referent (See Figure 8). It runs two parallel processes: one for the absolute property of COLOR (lines 14–18) and one for the relative properties SIZE, LOCATION and ORIENTATION (lines 20–23). Inspired by Pechmann (1989), these latter properties that require comparison processes are analyzed incrementally.

Stepping through the first parallel process in analyze_SP, the function do_attribute (line 18) returns the color value for the object as a function of its hues, luminances, saturations, and RGB values. See Figures 6 and 7 for an example. A lemma for this value is not immediately added. First, the check_interconn function is called in line 17. For each attribute that color typically suggests (line 27), if the color value implies another attribute-value true of the referent (line 30), this implied attribute-value is possibly associated to a lemma (line 31). For example, if the object is made of wood, and the value for COLOR is *tan*, then MATERIAL *wood* may be added to the description before COLOR *tan* is.

Note that this means that both a lemma for a property and a lemma inferentially related to a property can be included by the algorithm, and this may result in redundant inclusion of properties; a MATERIAL lemma may be included because it is suggested by COLOR, and it may also be included as part of analyze_CP. This redundancy is intentional. During evaluation, I ignore (do not generate) repeat properties, but this method of redundant generation opens the possibility of generating redundant properties if desired.

Stepping through the second parallel process in analyze_SP, values for SIZE, LOCATION, and ORIENTATION are created. This comparison process is based on the average visual impression of the scene (its "gist") and not comparison against individual objects. In the

do_attribute input:	object location in image, image, SIZE
do_attribute output:	<over, +> (from algorithm in Chapter 4)
do_attribute input:	object location in image, image, COLOR
do_attribute output:	Average RGB value: <240, 30, 180>

FIGURE 6. Example of function do_attribute inputs and outputs, for COLOR and SIZE.

lemma input:	<over, ->, SIZE
lemma output:	prenominal: <small, smaller, little . . . >
	postnominal: <that is little, that is smaller . . . >

FIGURE 7. Example of function lemma input and outputs.

case of SIZE, this means that the target object's dimensions are compared against the average height and width of other objects of the same type in the scene, as in Chapter 4. In the case of ORIENTATION, this could correspond to which orientation "pops out" (Treisman & Gelade, 1980) compared to average surrounding orientations; and in the case of LOCATION this is a general placement within the image (e.g., *right*, *left*, *top*, *bottom*) rather than location relative to another object or group of objects. For each value, a lemma is stochastically added using the add_lemma function. After analyze_SP is complete, it returns the identifying description constructed so far, r (line 06) (See Figure 8).

The algorithm then begins analyzing further properties in analyze_CP (lines 33–42). The input to this function is identical to analyze_CP, but the identifying description r is partially constructed following the output of analyze_SP. For each of the attributes in the ordered list CP, the algorithm calls to a function to return the attribute-value and checks for interconnected attributes (lines 38–43). A lemma for each attribute-value is stochastically added to r (line 44), with the probability of adding a further lemma quickly diminishing as the length of the identifying description increases. After analyze_CP is complete, it returns the identifying description constructed so far, r (line 10). At this point, the algorithm should have found the object's category (if available) from line 07.

analyze_SP input:

obj: $obj1$

scene:

$$\left[\begin{array}{l} obj1 \\ \left[\begin{array}{l} hue : h \\ luminance : l \\ saturation : s \\ corners : c \\ edges : e \\ RGBpixels : r \\ boundingbox : \quad size \quad coordinates \\ \qquad\qquad \left[63, 63\right] \quad \left[1, 10\right] \end{array}\right] \\ \\ obj2 \\ \left[\begin{array}{l} hue : h2 \\ luminance : l2 \\ saturation : s2 \\ corners : c2 \\ edges : e2 \\ RGBpixels : r2 \\ boundingbox : \quad size \quad coordinates \\ \qquad\qquad \left[80, 100\right] \quad \left[30, 80\right] \end{array}\right] \\ \\ obj3 \\ \left[\begin{array}{l} hue : h3 \\ luminance : l3 \\ saturation : s3 \\ corners : c3 \\ edges : e3 \\ RGBpixels : r3 \\ boundingbox : \quad size \quad coordinates \\ \qquad\qquad \left[60, 75\right] \quad \left[100, 150\right] \end{array}\right] \end{array}\right]$$

r: $<>$

analyze_SP output:

<COLOR:red, SIZE:small>

FIGURE 8. Example of function analyze_SP input and output.

This ends the parallel processes. Each object in the scene is then incrementally fixated on in incremental_obj (lines 43–55). If an object is found that is of the same type as the target object (line 47), then any attribute-values in which the two objects differ (line 51) are stochastically added to the identifying description (line 52).[3] The add_lemma function additionally gets the value of the comparator object as well as the target object, allowing the lemma function to optionally create negated comparison statements ("the one that is not red").

When incremental_obj is complete, the object category is added to the identifying description (line 10) and the algorithm is finished.

7.5.4. An Example. To see how this algorithm works, we will consider an example in detail. Running the algorithm a large number of times can give us a distribution over several possible outputs; for the sake of example, let us consider a likely pass through the algorithm. Suppose the task is to create a referring expression for obj in the scene scene (Figure 9). A previous step has provided the algorithm with visual properties of the area where the object is located, as in Figure 10.

Once refer is called, it initializes r to an empty set. find_category and analyze_SP are then both called in parallel, and the algorithm begins searching for a stored representation of the object while it analyzes the simple visual properties of COLOR and SIZE. For this color, there is no interconnected MATERIAL, and no typical COLOR for the object category. α_{color} has a high value since COLOR tends to be included in description, and there is no penalty for including another lemma because there are no lemmas yet created in this expression. The algorithm therefore has a high probability of adding a lemma for the COLOR value to r; for this example, the adjective *red*.

[3]Note that the outcome of the incremental scanning of objects will depend on the order in which the items are viewed, and object fixation patterns during free viewing are notoriously difficult to predict (Underwood, Templeman, Lamming, & Foulsham, 2008; Griffin & Bock, 2000). The effect of this is that expressions may be longer, and include redundant information, if an object of the same type is in the scene. Such an approach is also in line with previous research that shows an attribute such as colour is more likely to be included when scene variation is high (Koolen et al., 2011). Luckily, this scanning is relatively late in the algorithm, and so will not have as strong an effect as earlier processes on the selection of attribute-values.

FIGURE 9. Example object and scene.

$$
\begin{array}{c}
obj \\
\left[\begin{array}{l}
hue: \\
\qquad \left[\begin{array}{l} R: <181:1596, 180:1303, 179:1301 \ldots > \\ G: <29:1524, 27:1449, 23:1395 \ldots > \\ B: <32:1628, 31:1528, 28:1472 \ldots > \end{array}\right] \\
luminance: \\
\qquad\qquad \left[\begin{array}{l} 10:.20 \\ 20:.40 \\ 30:.40 \\ 40:.50 \end{array}\right] \\
\vdots
\end{array}\right]
\end{array}
$$

FIGURE 10. Partial visual representation of object, represented as a series of histograms, low level visual features.

Simultaneously, the object's size is analyzed; its height and width may be within the realm of typical for this object, but there are several other objects in the scene with which we can compare an average height and width, calling the size algorithm from Chapter 4. This returns <over, ->. α_{size} and length(r) again yield a relatively high probability, and so a lemma will likely be added to r, for this example, *small*. LOCATION is next analyzed, followed by ORIENTATION, and values for these may be stochastically added as well. There is some competition between the two parallel processes: sometimes color lemmas may be processed faster than size, and vice versa, which could affect the length of the description and the likelihood of generating one of the other. For the current implementation, I assume that the length penalty for two properties on mirrored sides of the parallel process are equal; the length from one process does not affect the length penalty of the other until both are completed and the next properties are considered.

Complex properties are next analyzed in analyze_CP, and the algorithm first finds the MATERIAL attribute to be a typical value for this object. Because $\alpha_{material}$ yields a relatively high probability, meaning MATERIAL tends to be included in expressions and β_{val} yields some probability, since the typicality of the MATERIAL value is not the highest, the algorithm may add a lemma for this value to the expression. However, at this point the expression is likely to be relatively long (including adjectives for COLOR and SIZE), which is penalized by γ; and so it will only create the adjective *plastic* some of the time, e.g., if there are not already 3 adjectives selected. With 3 adjectives created, the length-based penalty will make it very unlikely the algorithm will create further words. If the generated expression is still relatively short and no other attributes are selected to be included, then scanning the rest of the objects in the scene (line 46) may return a further modifier marking a difference between the referent and other objects, for example, a realization of the SHEEN value, *less shiny*.

The algorithm will therefore likely generate structures for several different expressions with different frequencies. If we run the algorithm 1,000 times, we can get a distribution

over several possible forms, e.g.,:

Example Surface Form	**Example Frequency**
red fork	.70
red fork in the middle	.08
small red fork	.14
small red fork at the top	.04
red plastic fork	.02
small red plastic fork	.01
plastic red fork that is less shiny	.001
(etc...)	

As in the human data, these outputs are not normally distributed, with a preference for short phrases with color modifiers.

7.6. Discussion

In this algorithm, I attempt to optimally bring together together previous work with findings in this thesis. In analyze_SP, I follow the findings in Chapters 3 and 4 that the properties of COLOR and SIZE are particularly salient for referring in visual domains, and implement this by following work in vision by defining two parallel processes, one for the "what" pathway, defining COLOR, and one from the "where" pathway, defining SIZE, ORIENTATION, and LOCATION (Kosslyn, 1994; Murata, Gallese, Luppino, Kaseda, & Sakata, 2000). In future work, implementing different speed differentials for different properties may lead to further human-like variation. For example, the speed at which MATERIAL is processed as an interconnected property to COLOR may affect whether or not LOCATION is included; for now, COLOR (and interconnected properties) are processed as simultaneous with SIZE, affecting the length penalty for complex properties and LOCATION and ORIENTATION only after they have been processed. This would allow something like

COLOR to sometimes 'win' over SIZE, affecting the length parameter in the decision for whether or not to include SIZE in the description.

Further drawing from work in vision and computer vision, I define COLOR in terms of a multi-featured space of hues, luminances, etc. (Farhadi et al., 2009), and follow the experiments and models developed in Chapter 4 by defining SIZE in terms of a multi-featured space of dimensions of the object and other objects in the scene.

In check_interconn, I follow the findings from Chapter 5 that certain attributes are interconnected, and in particular, color is interconnected with material; the algorithm therefore is less likely to include a color modifier and more likely to include a material modifier if the color suggests the material. In analyze_CP, I follow traditional NLG methods by analyzing properties incrementally, but add a specification of which visual attributes that this applies to.

In incremental_obj, I follow Pechmann (1989), but differ from traditional NLG methods, allowing objects themselves to be analyzed incrementally.

For all of these functions, a stochastic decision-making processes is used to reflect the speaker variance we find throughout the thesis. This captures the notion that people tend to include atypical properties of objects, established in Chapter 5, and the fact that the longer the expression, the less likely it is to be produced.

I have not here addressed reference to more complicated spatial relations, such as topological or projective relations (Kelleher & Kruijff, 2006), part-whole relations, sets, or objects in video rather than still images. I hope that the current work provides a strong basis from which to further research in these areas.

In the next chapter, I evaluate how well this algorithm performs in several corpora of visible objects.

Chapter 8

Visible Objects Algorithm: Evaluation

8.1. Introduction

This chapter provides several evaluations of the algorithm introduced in Chapter 7. The proposed algorithm, which I will call the Visible Objects Algorithm, was designed to approximate human variation within a verbal (non-literary) setting introducing an object into the discourse within a group of visible, real world objects. To understand how well the algorithm performs and how it compares to the state of the art, I compare the proposed algorithm against implementations of two algorithms commonly used in referring expression generation: the Incremental Algorithm (IA) (Dale & Reiter, 1995) and the Graph-Based Algorithm (Graph) (Krahmer et al., 2003). (See Chapter 2 for a review of these algorithms.) Neither the Incremental Algorithm nor the Graph-Based Algorithm were developed specifically for the domain of reference to visible objects, but instead were intended to be general purpose algorithms, and I return to this issue in Section 8.5; to my knowledge, no REG algorithm has been developed from scratch especially for reference to visible objects (although both the IA and Graph have been built upon for visual domains; see Chapter 2). These evaluations therefore serve to introduce a new approach to generating referring expressions, creating specific algorithms for specific modalities.

I first establish that the Visible Objects Algorithm is reasonably effective at generating human-like expressions and is competitive with implementations of previous algorithms by evaluating all algorithms on two well-known REG corpora, the GRE3D3 corpus (Viethen & Dale, 2008) and the TUNA corpus (van Deemter et al., 2006). Because the algorithm is non-deterministic, I run it a number of times and compare its generated

sets against the observed human expressions. Both corpora contain expressions elicited to images of computer-generated objects, and so provide a reasonable starting point for evaluating reference to visible objects. To explore how well the algorithm approximates human reference to *real world* objects, I additionally evaluate on a third corpus of real objects sitting on a table, the Typicality corpus introduced in Chapter 5. For all algorithms, I evaluate the selection of referent attributes. Lexical choice, word order, and attribute values are not taken into account.

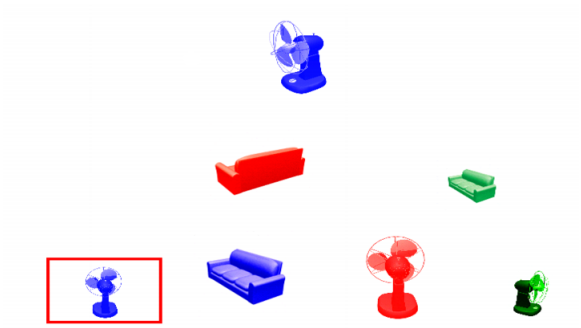
Table 1 lists the algorithms and corpora used in this chapter. Example objects from the GRE3D3 corpus, the TUNA corpus, and the Typicality corpus are shown in Figure 1. I will first briefly summarize how each corpus highlights different aspects of the Visible Objects Algorithm, and then discuss the evaluation measures I use (Section 8.3) and implementation details for each of the algorithms (Section 8.4). Further details about the corpora and input/output for the algorithms is available within each evaluation section. Section 8.6 details evaluation on the GRE3D3 corpus, Section 8.7 details evaluation on the TUNA corpus, and Section 8.8 the Typicality corpus.

Algorithms Used in Evaluation	**Corpora Used in Evaluation**
The Incremental Algorithm	GRE3D3 Corpus
The Graph-Based Algorithm	TUNA Corpus
The Visible Objects Algorithm (proposed)	Typicality Corpus

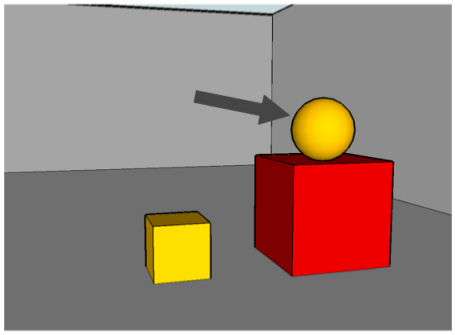
TABLE 1. Algorithms and corpora in evaluation.

8.2. Overview of Corpora

The GRE3D3 corpus is useful for testing some of the basic ideas behind the proposed algorithm: the selection of COLOR and SIZE in two separate processes and the use of prior likelihoods and description length to select attributes for the description. The latter attribute selection process is an alternative to selecting attributes based on whether they rule out comparator objects; ruling out comparator (or *distractor*) objects is a main attribute selection criterion in the IA and Graph. The GRE3D3 corpus therefore serves



TUNA corpus



GRE3D3 corpus

Typicality corpus

FIGURE 1. Example items from corpora.

as a simple, semantically transparent corpus for evaluating the most commonly discussed attributes in generating reference to objects (COLOR and SIZE) for all algorithms.

The TUNA corpus allows us to extend the evaluation slightly farther, adding the typicality aspect of the proposed algorithm, and specifically looking at the typicality of COLOR. The Visible Objects Algorithm's choice to include a lemma for an object's color in this domain is based not only on the length of the description created up to that point and the prior likelihood of color being included in the final expression (as in the GRE3D3

corpus), but may also be based on how typical the color of the object is. The TUNA corpus also contains expressions elicited to black and white images of people, but because this thesis focuses explicitly on reference to *inanimate objects* (which people may process differently than animals and people – see Chapter 2), I do not include this section of the TUNA corpus. The TUNA furniture sub-corpus therefore serves to again test the common attributes of COLOR and SIZE, but in a different domain; in this domain, the objects are more complex objects images of furniture rather than the simple geometric objects in the GRE3D3 corpus, and may introduce issues of typicality. This allows us to understand how the algorithms fare with both simple and complex objects.

The final corpus in these evaluations, the Typicality corpus, allows us to make a leap towards the realm of real world objects and address typicality more directly, using real objects on a table with one typical and one atypical property. It also allows us to evaluate on the selection of *complex* properties: Recall that the vision-inspired approach I propose makes a distinction between the simple absolute property of COLOR, the simple relative properties of SIZE, LOCATION, and ORIENTATION, and the complex properties of SHAPE, MATERIAL, etc. The previous two corpora use a small set of properties which I have called *simple*, and most of the properties are relative (SIZE, LOCATION/ORIENTATION). The Typicality corpus adds the complex properties of SHAPE and MATERIAL, which have not received a great deal of attention in work on reference generation, in addition to the more common attributes of COLOR, SIZE, etc.

The GRE3D3 and TUNA corpora contain expressions elicited to simple computer-generated objects, and the Typicality corpus contains expressions elicited to real world objects within a more complex task. These evaluations therefore shed some light on the breadth of the algorithms' capabilities, evaluating on both semantically transparent domains (GRE3D3, TUNA) and more complex real world domains (Typicality).

Another reason it is useful to look at all three corpora is because of the possible difference in the registers of the corpora: it is an open question whether TUNA and GRE3D3

may be considered corpora of literary expressions or corpora of more conversational or verbal expressions (see Chapter 2). One thing this evaluation tests is whether the proposed algorithm, which uses a more verbal/conversational account of reference, does better at capturing the observed expressions in these corpora than algorithms like the IA and Graph, which make a literary assumption. In contrast to TUNA and GRE3D3, expressions were produced orally in the Typicality corpus, with a real hearer present, viewing the same scene; it is therefore useful to see how the algorithms perform across these different corpora. It may be the case that IA and Graph do better at the more literary TUNA and GRE3D3 corpora, whereas the proposed algorithm does better at the clearly verbal Typicality corpus. In fact, we find that the proposed algorithm does as well as or better than Graph and IA on all three corpora, suggesting that overall, a conversational, descriptive view of initial reference may be more suited to how speakers introduce referents in visual domains.

As has been detailed in previous chapters, modifier inclusion is not normally distributed. This is clear examining expressions in the REG corpora, where we find that there are preferences for some expressions over others (for example, short expressions containing color modifiers). To evaluate, we must measure how well the expressions produced by the various algorithms match the observed distribution.

8.3. Evaluation Measures

Throughout this chapter, I discuss evaluation using *attribute sets* for a referent. A *human-produced* attribute set is the set of attributes annotated for a human's referring expression for a referent, such as we find in the GRE3D3, TUNA, and Typicality corpora. A *predicted* attribute set is the set of attributes predicted by an algorithm for a referent. I first represent each member of an attribute set as a triple $x : y : z$ where x is the object ID, y is the attribute, and z is the value. In evaluation, I only look at attributes of the target referent and treat values as boolean. An example is shown in Table 2.

Example Expression	Corresponding Attribute Set	Evaluated Attribute Set
the red ball	tg:type:ball tg:color:red	tg:type:1 tg:color:1 tg:size:0 tg:location:0

TABLE 2. Example human-produced expression and corresponding attribute sets for evaluation with attributes TYPE, COLOR, SIZE, and LOCATION.

8.3.1. Background. It is not immediately obvious how to evaluate a stochastic algorithm. Previous evaluation of REG algorithms have used measurements such as Uniqueness, Minimality, Dice (Belz & Gatt, 2008), Accuracy, String-edit distance, BLEU, NIST, and ROUGE (Gatt, Belz, & Kow, 2009; Reiter & Belz, 2009). *Uniqueness* is the proportion of attribute sets generated by a system that identify the referent uniquely, and *Minimality* is the proportion of attribute sets that are both minimal and unique. As my goal is to generate natural, human-like reference, and humans occasionally underspecify (e.g., do not always identify the referent uniquely, at least by using only object attributes), these metrics are not as useful for the evaluations as the others.

Accuracy and *Dice* measure the proportion of attribute sets generated by a system that match the corresponding corpus attribute sets. Accuracy is a measure of the proportion of attribute sets that match perfectly, and Dice is a measure of the overlap between attributes within two sets. These are useful for measuring how well a system's output matches a human's output, and I therefore use these metrics in the evaluation. Since the proposed algorithm is stochastic, this introduces a problem in using Dice, as the group of observed attribute sets for a referent and the group of predicted attribute sets for a referent must somehow be aligned. I discuss this in further detail below.

String-edit distance (also known as *Levenshtein* distance) measures the minimal number of insertions, deletions and substitutions required to transform a system's output attribute set to the reference attribute set. In these evaluations, the number of attributes in the system's output attribute set is always equal to the number of attributes in the human-produced attribute set, making String-edit distance proportional to Accuracy.

Attribute Set Produced by Algorithm	**Attribute Set Produced by People**	**Surface Form**
tg:type:fan tg:size:small	tg:type:fan tg:size:small	*small fan*
tg:type:fan tg:size:small	tg:type:fan tg:size:small tg:rel_location:lm,above lm:type:desk lm:color:green	*small fan above a green desk*

TABLE 3. Example attribute sets produced by algorithms and people. Aspects of the human-produced descriptions that I do not address are shown in grey. Object attributes are treated as binary, included or excluded.

BLEU, *NIST*, and *ROUGE* are n-gram based string comparison measures commonly used in evaluating machine translation systems and measure the amount of overlap between the output string and the reference string. I have not experimented with using these last methods. A further method has recently been used for the purpose of evaluating a stochastic algorithm in van Gompel et al. (2012). This measures the likelihood that the algorithm will predict the corpus of attribute sets observed in the participant's data as the probability density function for the predicted distribution. Further details are provided below.

8.3.2. Method. In order to compare the stochastic Visible Objects Algorithm to the deterministic algorithms, I evaluate all algorithms in two ways. I further provide qualitative comparisons between the observed attribute sets and the predicted attribute sets using the distributional method reported in van Gompel et al. (2012).[1] In all evaluation methods, attributes are treated as *boolean*: I do not evaluate on the chosen value, but whether the attribute has been selected or not. Ignoring the attribute's value is especially important in the Typicality corpus, where participants often report an incorrect value.

In the first evaluation method, which I will call MaxAlign, I measure both Accuracy and Dice. Because the proposed algorithm is stochastic, how accurate its predictions are

[1]Further details provided by discussions with the authors.

depends on how its predicted attribute sets are aligned to the observed attribute set. In MaxAlign, I thus find the optimal alignment between the two corpora, yielding the algorithm's maximum Accuracy/Dice score for the evaluation corpus. For the corpus of observed attribute sets I and the corpus of generated attribute sets J, I seek to find the best alignment x out of all possible alignments X between the corpora. The alignment score for two attribute sets i, j is calculated as the number of attributes i_p and j_p from the attributes $p \in P$ that are both included or both excluded, normalized by the number of properties being evaluated. In other words,

$$MaxAlign = \arg\max_{x \in X} \sum_{(i,j) \in x} Alignment(i, j)$$

where

$$Alignment(i, j) = \sum_{p \in P} \frac{(i_p \wedge j_p) \vee (\neg i_p \wedge \neg j_p)}{|P|}$$

A given set of properties such as <a, b> would therefore be treated as <a, b, not c, not d>. Calculating the alignment score over the number of evaluated properties (|P|) has the nice mathematical property of making *Alignment* equal to other common metrics for evaluating a model, including *Accuracy*, *Dice*, *Precision*, and *Recall*. It also allows us to limit each evaluation to the set of properties that have been annotated in each corpus. Note that because IA and Graph are deterministic, finding an optimal alignment is trivial. Because the proposed algorithm is non-deterministic, we run it five times for each scene, and calculate the average score for each.

It is an open question whether the MaxAlign evaluation is fair: Because the proposed algorithm is stochastic and the other algorithms are not, a higher score for the proposed algorithm is due to more alignment options. In the second evaluation method I address this issue, comparing how well the Visible Objects Algorithm's *most likely* predicted attribute set compares with the IA and Graph's predicted attribute set.

The second evaluation method, which I will call Maj, measures whether the most frequent predicted attribute set out of all the predicted attribute sets corresponds to the observed

majority attribute set. Given a target referent t, we can define the boolean-valued variable $Match_t$, where:

$$Match_t = \begin{cases} 1 & \text{if most frequently predicted} = \text{most frequently observed} \\ 0 & \text{otherwise} \end{cases}$$

Calculating these scores over several evaluation targets $t \in T$, we can obtain the proportion of targets that have a majority match:

$$Maj = \frac{\sum_{t \in T} Match_t}{|T|}$$

This is a simple way to fairly compare the output of deterministic and non-deterministic algorithms. This evaluation method is limited to those observed attribute sets that include only attributes under consideration. Parts of an attribute set that include a description of a relatum are excluded from the analysis (e.g., for the phrase "sphere on top of the red cube", annotations related to "red cube" are ignored). There are no ties in the predicted sets, but in the case of a tie in the observed data (more than one attribute set with the highest frequency), I count a match if any of the members of the tie match the most frequent predicted attribute set.

In the third method, Frequency Prediction (FreqPred), I determine the likelihood that the proposed Visible Objects Algorithm will produce the observed corpus of attribute sets. In other words, from the distribution over outputs produced by the algorithm, this method provides the likelihood that the algorithm will produce the observed corpus frequencies. This allows us to see how well the distribution of attribute sets predicted by the algorithm for a referent reflects the variation in people's reference. The nice thing about a distributional analysis is it shows us what kinds of expressions the algorithm fails to predict; it provides a qualitative analysis of the attribute sets that the algorithm is not predicting, or not predicting enough of.

For the Graph-Based Algorithm and the Incremental Algorithm, this likelihood of predicting the variation in the observed corpus is always 0.0; people do not all produce the same attribute set for a referent in any of the corpora, and neither algorithm predicts more than one attribute set for a given referent. For the Visible Objects Algorithm, I produce a model that defines a multinomial probability distribution d over k attribute sets by running the algorithm 1,000 times and estimate the likelihood for each attribute set using maximum likelihood estimation. I then calculate the likelihood that the observed data – the corpus of human-produced attribute sets – would be predicted by the model as the probability density function (pdf) for the Visible Object Algorithm's predicted distribution. In other words, given the observed data x composed of n participants' attribute sets, we can calculate $\mathrm{p}(x|d,n)$ as:

$$\mathrm{p}(x|d,n) = \frac{n!}{x_1!...x_k!} d_1^{x_1}...d_k^{x_k}$$

where $x = (x_1, ..., x_k)$ gives the number of each of k attribute sets for n participants with estimated likelihood d_k for each. Put more simply, this measures the likelihood that we'll see each of our observations as frequently as we do, if we are generating from the underlying probability distribution of the algorithm.

As in the majority evaluation, this evaluation is limited to those observed attribute sets that include only attributes under consideration, and parts of an attribute set that include a description of a relatum are excluded from the analysis. There is a large degree of variation between each referent using this approach, and I report values for the highest and lowest probabilities in each corpus along with qualitative results showing the frequencies of predicted and observed attribute sets.

For each corpus I evaluate on, I report MaxAlign, Maj, and FreqPred using x-fold cross validation. In each fold, I estimate parameters for the algorithms using the human-produced expressions for all referents but one; the held-out referent then serves as the test item. For MaxAlign, I run the algorithms to produce as many expressions as are observed, and for Maj I run the Visible Objects Algorithm 1,000 times. I repeat this so that each

referent in each corpus is a test item once, and report the average scores over all folds. For FreqPred, I run the Visible Objects Algorithm 1,000 times, repeating this so that each referent in each corpus is a test item once, and report some of the highest and lowest scoring test scenes. The proposed algorithm is a bit of a black box; individual aspects of the algorithm, such as parallelism or the selection of specific properties, still need to be evaluated. However, by evaluating the algorithm using several different metrics, I hope to capture whether it is a reasonable competitor to IA and Graph while for the first time capturing speaker variation.

8.4. Implementations

8.4.1. The Incremental Algorithm. The version of the Incremental Algorithm I use is available from the NLTK[2] (Bird, Loper, & Klein, 2009). This algorithm requires that the following be provided:

(1) A preference order list (PO) specifying the order to iterate through the attributes.
(2) The attributes and values (properties) of all objects in the context set.

For (1), the problem of finding the best preference order for the Incremental Algorithm was explored in detail by van Deemter et al. (2012). Looking at the TUNA furniture sub-corpus, they find that an order with COLOR followed by SIZE yields the best results. ORIENTATION is the only other attribute considered in this domain, and changing its position before or after SIZE does not have a significant impact on performance. Other approaches have decided preference orders based on corpus frequencies (Koolen, Krahmer, & Theune, 2012), which also find a similar pattern of COLOR before SIZE.

In my implementation, I determine the preference order from corpus frequencies, using x-fold cross-validation to determine the attribute frequencies in each training fold. What I find is in line with van Deemter et al. (2012) – the preference order has COLOR followed

[2]https://github.com/nltk/nltk_contrib/blob/master/nltk_contrib/referring.py retrieved 1.Aug.2012

by SIZE and then other properties like ORIENTATION, for the TUNA corpus. I report on two versions of the Incremental Algorithm, one which places TYPE – corresponding to the head noun of the referent – as the first attribute in the preference order, and one which places it as the last. We will see that the algorithm performs optimally with TYPE placed at the end of the preference order.

For GRE3D3 and the TUNA furniture corpus, providing (2), the attributes and values of all objects in the context set, is straightforward; these corpora were built around computer-generated objects with simple, easily distinguished properties. (Example input to the algorithms for these corpora are shown for each of the evaluations, Figure 2 for GRE3D3 and Figure 7 for TUNA.) For the Typicality corpus, with real world visible objects, the problem is more complex.

The difficulty of defining attribute values for the Typicality corpus is especially evident for the attributes of COLOR and SIZE. As discussed in Chapter 4, the Incremental Algorithm requires that an object's SIZE be specified as, e.g., *small* or *large* before referring expression generation begins; similarly, an object's COLOR must be predefined. However, appropriate values for COLOR and SIZE are somewhat subjective, and it is unclear when an attribute-value for an object can be said to be significantly different enough from another. For example, are the two bowls shown in the Typicality corpus image in Figure 1 the same color? What about the rulers? For size, the question of size in comparison to *what* arises. For example, the pushpins are small relative to the other objects in the scene, but a normal size for a pushpin. The bowls are large relative to the scene, but a normal size for a bowl. The boxes are small for boxes, but medium-sized relative to the scene. It is unclear if size designations should be specified in comparison to other items in the scene, in comparison to other items of the same type, or in comparison to the size of the table. These decisions are fundamental to the algorithm, because the inclusion of an attribute-value is dictated by whether it is *different* from other items in the distractor set. With real world objects, the question is often whether the difference for an attribute-value is *different enough* (and this may vary person to person).

In order to evaluate the Incremental Algorithm on a real scene, we must make decisions for these issues before we can run it. I therefore decide to encode the bowls, mugs, envelopes, and screws as the same color; the rulers as *light tan* and *tan*; and the boxes as *light brown* and *dark brown*. Because the generation of a size modifier for an object is significantly more likely when there is another object of the same type in the scene (Brown-Schmidt & Tanenhaus, 2006), I create size values by comparing the objects of the same type. This method worked well for generating SIZE modifiers in Chapter 4. From this, I settle on using *medium* for all SIZE values, with exception to the envelopes; the envelope on the right in Figure 1 is labeled as *big* and the one on the left is labeled as *small*. To aid comparison across the different algorithms, I use the same COLOR and SIZE designations for the Graph-Based Algorithm, and the same COLOR designations in the Visible Objects Algorithm, discussed below. Example input to the algorithms following these decisions is shown in the evaluation using this corpus, Figure 9.

8.4.2. The Graph-Based Algorithm. The version of the Graph-Based Algorithm that I use is available from Viethen et al. (2008). This algorithm requires that the following be provided:

(1) A set of cost functions for each edge.
(2) The attributes and values (properties) of all objects in the context set.
(3) A preference order for deciding between attributes in the case of a tie.

As originally introduced, a graph-based approach to REG is a framework for implementing several different kinds of algorithms; the cost functions define the algorithm. Viethen et al. (2008) develop an approach using frequency information to assign costs, with the most frequent properties given a cost of 0 (free), the rarer properties a cost of 2 (expensive), and all other properties a cost of 1. This approach was shown to work well: this version of the Graph-Based Algorithm was the best performing system in the 2009 REG Challenge (Gatt et al., 2009).

Theune et al. (2011) find that using only two costs (0 and 1) achieves even better results, using k-means clustering (with k=2) over the relative frequencies of attribute-values to decide the cost. Following this method, I first determine the frequency with which each property (attribute-value pair) was mentioned for a target object in the training data, relative to the number of target objects with this property. Then I create a cost for each property (either 0 or 1) using k-means clustering with the Weka toolkit (Hall et al., 2009).

To briefly explain how this approach works, the k-means clustering algorithm partitions n observations into k clusters (S_1 to S_k) by assigning each observation to the cluster with the nearest mean. The total within-cluster sum of squares W is minimized by the function:

$$W = \arg\min_x \sum_{i=1}^{k} \sum_{x_j \in S_i} ||x_j - \mu_i||^2$$

where μ_i is the mean of the points $x_i \in S_i$. For Graph, each observation n corresponds to an attribute-value pair, and relative frequency is the only dimension of the vector. μ_i is thus the mean relative frequency of the properties in cluster S_i. The clusters are then ordered by their means and numbered. Costs are defined as follows:

$$\forall x_j \in S_i, cost(x_j) = i - 1$$

For (2), providing the properties of all objects in the context set, the same problem of what the attribute-values should be in a real world visual domain arises for the Graph-Based Algorithm just as it did for the Incremental Algorithm. I therefore use the same attribute-values for the Graph-Based Algorithm as for the Incremental Algorithm.

For (3), a preference order for deciding between attributes, I follow the same method as for the Incremental Algorithm. I obtain a preference order from corpus frequencies, using x-fold cross-validation to determine the attribute frequencies in each training fold and ordering the attributes in descending frequency.

8.4.3. The Visible Objects Algorithm. This algorithm is available through github.[3] Introduced in Chapter 7, the algorithm is written to take in a visual scene represented as a pixel-based image. To understand how the algorithm compares with the state of the art in REG, I am primarily interested in evaluating the aspects of the algorithm that handle the selection of attributes. This requires a stripped-down version of the algorithm, without a fixate function (line 02) returning a visual representation of the object from the image; this in turn requires that the find_category (line 07) function returns the category of the object using something other than visual information.

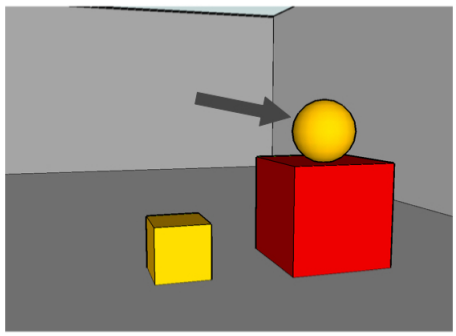
To isolate the selection of attributes from the rest of the algorithm, I begin with a "gold-standard" representation of the scene, written as object identifiers followed by their properties, including a TYPE attribute, comparable to the input for the Incremental and Graph-Based Algorithms. An example input for all systems in the GRE3D3 domain is given in Figure 2. Details of the annotated attribute-values are provided in Section 8.6.1.

The algorithm is therefore changed in the following ways:

- Instead of using a visual fixate function, an object representation is simply accessed from the dictionary scene using the object ID; this returns a set of properties corresponding to the specified object.
- The KB.find_category function accesses the stored category information about the object (typical properties) from the knowledge base based on the object's TYPE.
- The do_attribute function, which is intended to analyze features of the object within the scene and return a semantic representation for the attribute's value (e.g., <240, 30, 180> as a value for COLOR) instead returns the gold-standard attribute value from the scene representation (e.g., *green* for COLOR).
- The function lemma can then just return the attribute value.

[3] https://github.com/itallow/VisibleObjectsAlgorithm.

Scene:



GRE3D3 Scene Input to the Incremental and Graph-Based Algorithm:

tg	color:yellow	size:small	type:ball	location:right-hand	rel_location:lm,on-top-of	rel_location:obj3,right-of
lm	color:red	size:large	type:cube	location:right-hand	rel_location:tg,below	rel_location:obj3,right-of
obj3	color:yellow	size:small	type:cube	location:left-hand	rel_location:tg,left-of	rel_location:lm,left-of

GRE3D3 Scene Input to the Simplified Visible Objects Algorithm:

tg	color:yellow	size:(63,63)	type:ball	location:right-hand	rel_location:lm,on-top-of	rel_location:obj3,right-of
lm	color:red	size:(345,345)	type:cube	location:right-hand	rel_location:tg,below	rel_location:obj3,right-of
obj3	color:yellow	size:(70,70)	type:cube	location:left-hand	rel_location:tg,left-of	rel_location:lm,left-of

FIGURE 2. Example input to the algorithms: GRE3D3 Scene 7. On the left is the object ID, followed by a vector of properties, represented as attribute-value pairs. For SIZE, the Visible Objects Algorithm is one step closer to the raw visual input, using the height and width (height,width) of the object. Evaluation measures the selection of the attribute (not its value).

- We also do not need to store any of the object's attribute values in the known_attributes dictionary, because it becomes trivial to get them as many times as we'd like from the object representation itself.

A list of these changes is available in Table 4. All attribute values are provided as gold-standards except for SIZE, for which we may get a bit closer to the visual input, defining just the pixel height and width and bringing in the hand-written algorithm for SIZE from Chapter 4.

One aspect of the lemma creation process in left unimplemented: The role of visual salience (δ) in the selection of a property. This factor (as well as others, such as conversational salience) clearly play a role in generation; for now, I focus on how well the algorithm performs when just attribute likelihoods (α) and typicality likelihoods (β) are

Function	Change
main	added; a given object's properties indexed from a scene dictionary provided as input.
KB.find_category	returns KB object category given object TYPE rather than visual information.
do_attribute	removed; gold-standard attribute value instead accessed directly from dictionary for all attributes except SIZE. SIZE is determined using SizeMod, the algorithm from Chapter 4.
lemma	removed; lemma now identical to val.
known_attributes	removed; attribute values accessed directly from obj.
scene	changed to a representation of the object identifiers and their associated properties; akin to the representation of a *context set* of earlier work.

TABLE 4. Changes to Visible Objects Algorithm.

defined. This means that the stochastic algorithm from Chapter 7 is redefined as the following:

$$p(y \rightarrow lemma|d) = \alpha_x \times \gamma + (1 - \beta_y) \times (1 - \alpha_x) \tag{8.1}$$

A difficulty in implementation is the parallel processing of the proposed algorithm: parallel processing makes the timing of the different processes a large factor in how it performs. Without knowing how long the various mechanisms of the algorithm should take in order to produce the most human-like output, I strive to find an implementation that does not rely on timing, but can be represented more straight-forwardly, e.g., fully serially.

Following the pseudocode in Chapter 7 Section 7.5, parallel processing applies to the functions of (1) Finding the object category (line 07); (2) Processing COLOR, followed incrementally by other entailed absolute properties (lines 17–21) and (3) Processing SIZE, followed incrementally by other relative properties (LOCATION and ORIENTATION, lines 23–26). To approximate this in an algorithm that runs serially, we can run all these

processes in their written order. Since the first absolute property processed (COLOR, lines 17–19 and line 21) is simultaneous with the first relative property processed (SIZE, lines 23–26), in implementation we can approximate this by saying that the length of the identifying description (length(r)) is 0 for both of these properties. This is a simplification, assuming the processing of COLOR will take just as long as SIZE: That is, if the length penalty for both is 0, then it cannot have been the case that one was added to the identifying description as a modifier before the other. I do not evaluate processing different attributes simultaneously, with different times: For now, I simply evaluate the independent selection of COLOR and its interconnected properties from the selection of SIZE.

As discussed in Chapter 7 Section 7.5, the incremental_obj function iterates over the objects in the scene, but the order is not specified. For my evaluations, the order I use is simply the order in which the objects are listed in the various corpora. For the GRE3D3 corpus, this order spans out in proximity from the target referent. For the TUNA and Typicality corpus, the order is random. We also need to provide the order in which complex properties should be analyzed (CP). I define this in the same way I define the Incremental Algorithm's preference order, defining the order of attributes based on frequency in the training data.

With these changes, the evaluated algorithm follows the pseudocode written in Figure 3. I additionally add a main function, called given scene – a dictionary listing all objects with unique IDs and properties (as in Figure 2) – and the ID of the desired referent.

The simplified algorithm evaluated here therefore requires the following be provided:

(1) A prior distribution on the inclusion of different attributes. Represented in the algorithm as α_{att}.
(2) The properties of all objects in the scene. Represented in the algorithm as scene.
(3) A complex property list specifying the order in which to create lemmas for complex properties. Represented in the algorithm as CP.

```
01 func main(scene, referent_id):
02   obj = scene[referent_id]
03   r = refer(obj, scene)

04 func refer(obj, scene):
05   r = <>
06   cat = KB.find_category(obj['TYPE'])
07   r = analyze_SP(obj, scene, r)
08   r = analyze_CP(obj, scene, r)
09   r = incremental_obj(obj, scene, r)
10   r += <cat.type>
11   return r

12 func analyze_SP(obj, scene, r):
13   att = 'COLOR'
14   val = obj[att]
15   r += check_interconn(obj, scene, att, val, r)
16   r += add_lemma(att, val, length(r))
17   for att ∈ <'SIZE', 'LOCATION', 'ORIENTATION'>:
18     if att == 'SIZE':
19       rx = obj['WIDTH'], ry = obj['HEIGHT']
20       dx = average width, other scene objects
21       dy = average height, other scene objects
22       val = SizeMod(rx, ry, dx, dy)
23       r += add_lemma(att, val, 0)
24     else:
25       val = obj[att]
26       r += add_lemma(att, val, length(r))
27   return r

28 func check_interconn(obj, s, att, val, r):
29   i = <>
30   for i_att of KB.interconnected(att):
31     ival == obj[att]
33     if KB.implies(att, val, i_att, ival):
34       i += add_lemma(i_att, ival, length(r))
35   return i

36 func analyze_CP(obj, scene, r):
37   for att ∈ CP:
39     val = obj[att]
43     r += check_interconn(obj, scene, att, val, r)
44     r += add_lemma(att, val, length(r))
45   return r

46 func incremental_obj(obj, scene, r):
47   for d in scene:
48     dobj = scene[d]
49     if dobj == obj:
50       continue
51     if dobj['TYPE'] != obj['TYPE']:
52       continue
53     for att ∈ CP ∪ SP:
54       if att == 'SIZE':
55         rx = obj['WIDTH'], ry = obj['HEIGHT']
56         dx = dobj['WIDTH'], dy = dobj['HEIGHT']
57         val = SizeMod(rx, ry, dx, dy)
58       else:
59         dval = dobj[att]
60         val = obj[att]
61         if dval != val:
62           l = add_lemma(att, val, length(r), dval)
63           if l not in r:
64             r += l
65   return r

66 func add_lemma(att, val, len, dval=None):
67   l = <>
68   if dval:
69     l = val
70   else:
71     if throw_dice(α_att, cat.β_val, len):
72       l = val
73   return l

74 func throw_dice(α_x, β_y, len):
75   if len == 0:
76     γ = 1.0
77   else:
78     γ = 1/(len * g)
79   weight_function = α_x * γ + (1 - β_y) * (1 - α_x)
80   n = random number between 0 and 1
81   if n < weight_function:
82     return True
83   return False
```

FIGURE 3. The algorithm: Implementation details. refer is called from main, given scene – a dictionary listing all objects with unique IDs and properties – and the ID of the desired referent.

(4) A scan list specifying the order to iterate through other objects in the scene. This implementation simply uses the order in which objects are listed for scene.

(5) A knowledge base (KB) available at all times with typical properties for objects, and associated likelihoods. This information is accessed in the algorithm using KB.find_category.

(6) A knowledge base (KB) available at all times with interconnected properties. This information is accessed using KB.interconnected and KB.implies.

(1) is similar to the cost functions for the Graph-Based Algorithm, but allows the selection of attributes to be non-deterministic; attributes are selected using prior likelihoods. (2) is the same as the context set (or scene) provided for the previous algorithms, however I leave SIZE to be determined from the pixel measurements or metric measurements rather than predefining it, bringing the input closer to the visual scene. (3) is similar to the Incremental Algorithm's preference order, but applies to the order in which complex properties are realized linguistically. For the GRE3D3 and TUNA corpora, complex properties are not available and so a complex property list is not used.

(4), (5), and (6) are novel to this algorithm. (4) defines an order in which to compare the target object against other objects in the scene, similar to the order in which a person may scan through the objects in the scene. (5) and (6) serve to implement the idea that objects in the real world are situated, with prior knowledge of the object affecting how they can be described.

8.5. Algorithm Comparison: Is It Fair?

An issue in these evaluations is whether it is fair to compare the algorithms. The Graph-Based Algorithm and the Incremental Algorithm are deterministic; the proposed Visible Objects Algorithm is not. Graph and IA were not written for any particular input, but were built to be more general, while the Visible Objects Algorithm was written specifically for visual input, and all algorithms are here evaluated in domains of visible

objects. Another issue is that the features corresponding to SIZE are not the same across algorithms. The Visible Objects Algorithm uses height and width features, whereas Graph and IA require values for an attribute like SIZE to be processed before REG begins (e.g., with values such as *large* or *small*). The properties corresponding to LOCATION are also not the same between algorithms, with the Visible Objects Algorithm requiring features for LOCATION explicitly, while Graph and IA have no constraints on this type of input and can operate on, e.g., the X-DIMENSION independently of the Y-DIMENSION (an approach suggested by the annotation in the TUNA corpus).

The issue of comparing apples to oranges in the comparison of the deterministic, general Graph and IA algorithms to the stochastic, vision-specific Visible Objects Algorithm is a problem of the state of the art. At the time of this writing, there are no other implemented, non-deterministic REG algorithms. It is therefore informative to see how the proposed algorithm compares to deterministic algorithms; if it performs as well as or better than these algorithms, then the proposed algorithm offers a viable alternative approach to generating descriptions of visible objects, particularly if the goal is generating human-like reference.

There are few publicly available REG algorithms that are constructed for a specific input domain, but evaluations of Graph, IA, and variations on these algorithms tend to use as input a visual domain, frequently the TUNA domain used in this evaluation (Viethen et al., 2008; van Deemter et al., 2012; Koolen et al., 2012). Therefore, although these algorithms were not written for any particular domain, in current research they tend to be used in visual domains: in comparison with this algorithm, I aim to address such an input head-on.

The input for SIZE differs between the proposed algorithm and the IA/Graph because the proposed algorithm was constructed for a visual domain. Focusing on SIZE (see Chapter 4), it became clear that a visual input such as an image will *not* tell you what is large or what is small – it will only tell you what the height and width of objects are. Because

I developed mechanisms for SIZE specifically, the proposed algorithm can operate on object height and width, bearing a more direct connection to the visual domain (such as is provided from the output of an object recognizer, for example). However, this places it at a possible disadvantage when evaluated against Graph and IA, as it must not only decide whether or not to include a SIZE attribute, but also what its semantic value should be.

I maintain this possible disadvantage with the goal of bringing REG algorithms closer to a visual input; although my research is limited to using "gold-standard" values for most attributes, SIZE is one attribute where we can process the visual scene more directly. By evaluating on the selection of attributes and not on their values, any discrepancies caused by this difference in values should be minimized.

A similar problem to SIZE arises for LOCATION. Because the proposed algorithm is designed to handle visible objects, it requests LOCATION features explicitly (e.g., a multi-dimensional vector marking features for x, y, and z coordinates) and provides no mechanism for processing an X-DIMENSION separately from a Y-DIMENSION, as suggested by the annotations in the TUNA domain. Due to this possible discrepancy between algorithms, I do not evaluate the attributes of LOCATION, X-DIMENSION or Y-DIMENSION in the TUNA domain.

My hope in comparing against the (quite different) state of the art is to illustrate the benefit in defining the details of the input domain as clearly as possible when constructing an algorithm, focusing on modality specifics. By working on a *visual* modality, comparing across several different visual corpora, common properties of that modality that are important to handle become clear, and each can be focused on explicitly.

8.6. Evaluation 1: GRE3D3

8.6.1. The Corpus. In the first evaluation, I use the GRE3D3 corpus (Viethen & Dale, 2008). It is necessary to add semantic annotations to this corpus in order to use

```
01 func main(scene, referent_id):
02   obj = scene[referent_id]
03   r = refer(obj, scene)

04 func refer(obj, scene):
05   r = <>
06   cat = KB.find_category(obj['TYPE'])
07   r = analyze_SP(obj, scene, r)
08   r = analyze_CP(obj, scene, r)
09   r = incremental_obj(obj, scene, r)
10   r += <obj['TYPE']>
11   return r

12 func analyze_SP(obj, scene, r):
13   att = 'COLOR'
14   val = obj[att]
15   r += check_interconn(obj, scene, att, val, r)
16   r += add_lemma(att, val, length(r))
17   for att ∈ <'SIZE', 'LOCATION', 'ORIENTATION'>:
18     if att == 'SIZE':
19       rx = obj['WIDTH'], ry = obj['HEIGHT']
20       dx = average width, other scene objects
21       dy = average height, other scene objects
22       val = SizeMod(rx, ry, dx, dy)
23       r += add_lemma(att, val, 0)
24     else:
25       val = obj[att]
26       r += add_lemma(att, val, length(r))
27   return r

28 func check_interconn(obj, s, att, val, r):
29   i = <>
30   for i_att of KB.interconnected(att):
31     ival == obj[att]
33     if KB.implies(att, val, i_att, ival):
34       i += add_lemma(i_att, ival, length(r))
35   return i

36 func analyze_CP(obj, scene, r):
37   for att ∈ CP:
39     val = obj[att]
43     r += check_interconn(obj, scene, att, val, r)
44     r += add_lemma(att, val, length(r))
45   return r

46 func incremental_obj(obj, scene, r):
47   for d in scene:
48     dobj = scene[d]
49     if dobj == obj:
50       continue
51     if dobj['TYPE'] != obj['TYPE']:
52       continue
53     for att ∈ CP ∪ SP:
54       if att == 'SIZE':
55         rx = obj['WIDTH'], ry = obj['HEIGHT']
56         dx = dobj['WIDTH'], dy = dobj['HEIGHT']
57         val = SizeMod(rx, ry, dx, dy)
58       else:
59         dval = dobj[att]
60         val = obj[att]
61         if dval != val:
62           l = add_lemma(att, val, length(r), dval)
63           if l not in r:
64             r += l
65   return r

66 func add_lemma(att, val, len, dval=None):
67   l = <>
68   if dval:
69     l = val
70   else:
71     if throw_dice(α_att, cat.β_val, len):
72       l = val
73   return l
```

FIGURE 4. Algorithm: How it works in GRE3D3 domain. cat.β_{val} is the default value of 1 for all attribute-values, meaning typicality has no effect.

it in the algorithms, and so in collaboration with the first author of GRE3D3 paper, we make this revised, annotated corpus available online.[4] This corpus contains expressions for 20 simple scenes. The scenes are split into two trial sets (Trial Set 1 and Trial Set 2), with Trial Set 1 having 30 expressions for each scene and Trial Set 2 having 33 expressions for each scene. Each participant produced expressions for all 10 scenes in one of the two

[4]http://m-mitchell.com/corpora/GRE3D3/xml/

sets. Each scene contains three objects, and in two colors: blue and green or red and yellow. Each object is either a sphere or a cube, and the object can be either large or small. Scenes from one trial set are shown in Figure 5. An example of the input to the algorithms is provided in Figure 2.

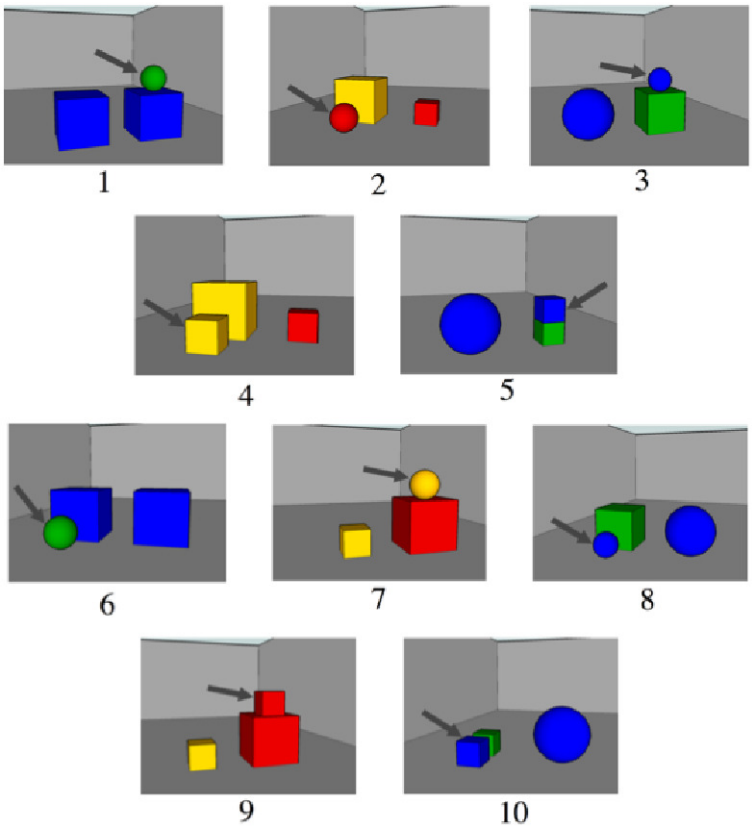


FIGURE 5. Images from GRE3D3 Trial Set 1.

Scenes in this corpus are constructed to systematically vary values for five attributes: COLOR, SIZE, LOCATION, RELATIVE LOCATION and TYPE. The LOCATION attribute picks out the location of the object in the scene (*right-hand* or *left-hand*), and the RELATIVE LOCATION attribute picks out the location of the object relative to the other objects (*on-top-of*, *right-of*). For example, an target object (tg) may be annotated as being on-top-of a landmark object (lm) and right-of the third object (obj3). Table 5 lists the full set of attributes, with possible values.

Attribute	Possible Values	Example	Explanation
COLOR	red, yellow, green, blue	color:red	The COLOR of the object is *red*.
SIZE	small, large	size:small	The SIZE of the object is *small*.
TYPE	ball, cube	type:ball	The object is a *ball*.
LOCATION	right-hand, left-hand	location:right-hand	The object is on the *right-hand* of the image.
REL_LOCATION	on-top-of, below, left-of, right-of	rel_location:lm,on-top-of	The object is *on top of* the landmark object (lm).

TABLE 5. GRE3D3 annotation labels and examples.

This corpus is particularly useful for evaluation of REG algorithms because it is well-controlled and semantically transparent. Because the corpus uses simple computer-generated geometric figures rather than real world objects, it minimizes issues of object expectation and typicality inherent in using everyday objects. Further, the objects are in a "scene" setting, arranged in physically plausible configurations within a room, which may make the broader context relatively natural. The GRE3D3 corpus is therefore useful to test the inclusion of TYPE and simple properties (COLOR, SIZE, LOCATION), without concern for complex properties such as SHAPE and MATERIAL.

Because these are not images of everyday objects, typicality does not play a role, and in the proposed algorithm all $\mathsf{cat}.\beta_{val}$ values are set to 1.0. This means that the stochastic function reduces to:

$$p(y \rightarrow lemma|d) = \alpha_x \times \gamma \tag{8.2}$$

This corpus therefore aids in applying some of the basic ideas of the Visible Objects Algorithm: That the selection of COLOR and the selection of SIZE within an identifying description may be processed separately (the selection of one not influencing the selection of the other); that these properties are foremost in the generation of reference to visible objects; that SIZE can first be determined from an overall gist of the scene, represented as height and width averages; and that natural reference to objects can be created *without ensuring that all distractors are ruled out*. The list of hypotheses I aim to address using this corpus is given below:

- COLOR and SIZE are selected independently of one another
- inclusion of an attribute in the identifying description is based on:
  (1) the description's length
  (2) the prior likelihood of including the attribute
- stochastic inclusion of each attribute aids in generating the distribution of attribute sets observed in human data

8.6.2. Preparing the Algorithms. I randomly select two scenes from Trial Set 1 (scenes 7 and 9) and their mirrored counterparts in Trial Set 2 (scenes 17 and 19) for development. The remaining eight scenes in each set are used in the evaluation.

As a first attempt at evaluating the algorithms in this domain, I do not evaluate the algorithms on their selection of RELATIVE LOCATION (e.g., *on-top-of*) and descriptions of related objects (e.g., "on top of the big red ball"). Creating language for such spatial properties and their related objects requires significantly extending the capabilities of the Visible Objects Algorithm as well as the Incremental Algorithm (Kelleher & Kruijff, 2006). RELATIVE LOCATION is therefore made available for all algorithms to select; but I exclude this property in the evaluation.

The sections of the Visible Objects Algorithm that are run in this evaluation are as shown in Figure 4. Portions of the algorithm that have no effect in this domain are colored in grey. Using the GRE3D3 development data, I find an optimal operating point with the weight for the length of the description, g, set to 5 (line 78, throw_dice function).

8.6.3. 1: Evaluation by Alignment (MaxAlign). In the first evaluation, I use eight-fold-cross-validation. In each fold, I use the seven training scenes to estimate:

- α values for attributes in the Visible Objects Algorithm.
- A preference order for attributes in the Incremental Algorithm, based on frequency (most frequent attribute first, TYPE placed either first or last and values for both versions reported).

- Relative frequencies of attribute-value pairs, which are then clustered to determine costs for the Graph-Based Algorithm, as discussed in Section 8.4.2.

I use the the algorithms to generate 30 attribute sets for Trial Set 1 and 33 attribute sets for Trial Set 2, corresponding to the number of expressions in each set. Because the proposed algorithm is non-deterministic, I run the algorithm five times in each fold and calculate the average MaxAlign.

GRE3D3		
ALGORITHM	TRIAL SET 1	TRIAL SET 2
Proposed Algorithm	88.23%	90.06%
IA - Type Last	87.71%	85.13%
IA - Type First	85.42%	84.19%
Graph	87.71%	88.73%

TABLE 6. Average Maximum Alignment (Accuracy) on GRE3D3 corpus.

Results are shown in Table 6. The proposed Visible Objects Algorithm achieves higher accuracy than either version of the Incremental Algorithm or the Graph-Based Algorithm. This suggests that the algorithm is competitive with the state of the art at producing human-like expressions for a referent. However, the algorithm may achieve a higher score because it is stochastic; there are a greater number of possible alignments to find the maximum alignment score. In the next evaluation I address this issue, comparing how well the Visible Objects Algorithm's *most likely* predicted attribute set compares with the IA and Graph's predicted attribute set.

8.6.4. 2: Evaluation of Majority (Maj). As before, I use the eight scenes in eight-fold cross-validation, estimating parameters on the seven training scenes in each fold. For each test scene, I run the proposed algorithm 1,000 times. The attribute sets predicted by each algorithm are ordered by how frequently they are predicted, and the most frequent attribute set is compared against the most frequent human-produced attribute set that contains the attributes under consideration. The majority score is the

percentage of folds where the most frequent attribute sets match. Results are shown in Table 7.

GRE3D3		
ALGORITHM	TRIAL SET 1	TRIAL SET 2
Proposed Algorithm	75.00%	50.00%
IA - Type Last	62.50%	25.00%
IA - Type First	37.50%	37.50%
Graph	62.50%	50.00%

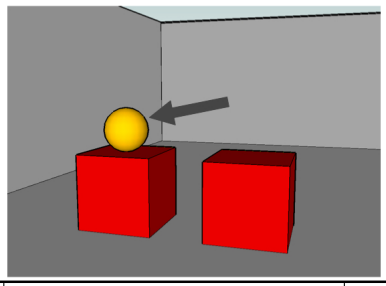
TABLE 7. Percentage of scenes where most frequently predicted expression matches most frequently observed expression.

The Graph-Based and the Visible Objects Algorithm both predict the majority attribute set in this evaluation at least 50% of the time, and reflect complementary strengths: While the Graph-Based Algorithm predicts only one attribute set, it tends to be the majority type. The Visible Objects Algorithm proposes many attribute sets, and most frequently predicts the majority attribute set.

8.6.5. 3: Frequency Prediction (FreqPred). In this part of the evaluation, we look only at the proposed algorithm and examine how the predicted attribute sets reflect the frequency of observed attribute sets. Because there is quite a bit of variation for each fold (some observed attribute sets are never predicted), I provide examples of the attribute sets for the scene with the highest probability and the attribute sets for one of the scenes with the lowest probability. Attribute set frequencies are shown in Tables 8 and 9.

Examining the differences between the predicted corpus and the observed corpus, we see some areas for improvement in the proposed algorithm. One clear area for future work is further expanding the kinds of location language it can produce, in particular for this corpus, generating attribute sets with RELATIVE LOCATION as well as LOCATION.

We also see an effect of the annotations. For example, the third row in Table 8 shows that LOCATION (corresponding to *on the left*) is not annotated as a property of the target

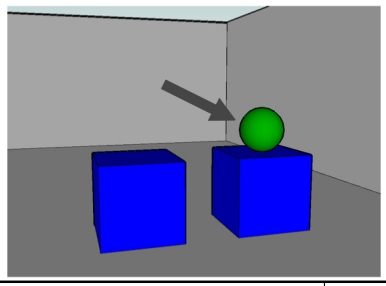


Predicted	Freq.	Observed	Freq.	Example Observed Human Expressions
tg:colour:yellow tg:type:ball	796 79.60%	tg:colour:yellow tg:type:ball	8 24.24%	*yellow ball, yellow sphere*
tg:type:ball	196 19.6%	tg:type:ball	7 21.21%	*ball, sphere*
tg:location:left tg:type:ball	4 0.40%			
tg:colour:yellow tg:location:left tg:type:ball	4 0.40%			
		tg:colour:yellow tg:rel_location: lm,on-top-of tg:type:ball	12 36.36%	*yellow ball on top of the red cube on the left*
		tg:rel_location: lm,on-top-of tg:type:ball	6 18.18%	*ball on top of the box*

TABLE 8. GRE3D3 Scene 11, best FreqPred match.
$p(x|d,n) = 0.011525$. Matching attribute sets shown in red.

referent. There is syntactic ambiguity as to where the prepositional phrase attaches; the property LOCATION:left may very well apply to both the target (tg) and the landmark (lm), but in these types of cases in the GRE3D3 corpus, LOCATION is marked as a property of the landmark alone. A prediction that includes LOCATION for a target referent will therefore not match, even though the participant may have intended this property to apply to the target referent. Further work may look into addressing these annotation issues.

It is clear from this that the distributions predicted by the algorithm are not close to the observed distributions. The algorithm predicts attribute sets that are not seen and does not predict attribute sets that are seen. I expect that any random selection of people will not produce all the attribute sets that the algorithm predicts; however, I would hope that



Predicted	Freq.	Observed	Freq.	Example Observed Human Expressions
tg:colour:green tg:type:ball	827 82.70%	tg:colour:green tg:type:ball	9 30.00%	*green ball*
tg:type:ball	156 15.60%	tg:type:ball	4 13.34%	*ball*
tg:location:right tg:type:ball	13 1.30%			
tg:colour:green tg:location:right tg:type:ball	4 0.40%			
		tg:rel_location: lm,on-top-of tg:type:ball	7 23.34%	*ball on top of the cube on the right*
		tg:colour:green tg:rel_location: lm,on-top-of tg:type:ball	6 20.00%	*green ball that is on top of the blue cube on the right*
		tg:colour:green tg:size:small tg:type:ball	2 6.67%	*small green ball*
		tg:colour:green tg:rel_location: lm,on-top-of tg:size:small tg:type:ball	1 3.34%	*small green ball on top of the blue cube on the right*
		tg:rel_location: lm,on-top-of tg:size:small tg:type:ball	1 3.34%	*little ball on top of the cube on the right*

TABLE 9. GRE3D3 Scene 1, one of the worst FreqPred matches. $p(x|d,n) = 0.00$. Matching attribute sets shown in red.

all attribute sets produced by people are at least predicted by the algorithm. But that is not the case. As can be seen in Table 9, fifth row, the set that includes COLOR, SIZE and TYPE – all individually attributes that the algorithm predicts – is not predicted.

To address this issue, I must further examine why the proposed algorithm does not include some of the attributes that people do include. A common mistake made by the

algorithm is that it does not predict SIZE when the target object is the only one of its type in the scene. SIZE is created in analyze_SP by a comparison with the overall "gist" of the scene, represented as height and width averages of other items *of the same type*. However, it makes some sense that this overall "gist" should not pay such close attention to type. This may also be different for different people, and in some cases the set of comparator objects used to determine the target object's type may include objects that are similar in some way other than basic type – for example, the fact that these are all basic geometric objects (same superordinate category) may make them similar enough to compare for size. Future work may evaluate versions of this algorithm that varies the comparison set used to determine the target object's size in analyze_SP.

8.7. Evaluation 2: TUNA

8.7.1. The Corpus. I next evaluate on the TUNA corpus furniture domain, which contains 7 sets of furniture items each with expressions elicited from 60 subjects. The sets are used in two conditions, with 30 subjects encouraged to use location (+LOC condition), and 30 subjects discouraged to use location (-LOC condition). 3 of the original 60 subjects needed to be removed from the corpus due to technical issues in their data, leaving 28 subjects in the +LOC condition and 29 in the -LOC condition.

This corpus is similar to the GRE3D3 corpus, with participant stimuli containing simple objects in basic colors, including variation of COLOR and SIZE values. This corpus also varies objects' ORIENTATION, and I add this attribute to the evaluation. RELATIVE LOCATION and properties of the relatum are also less of an issue in this corpus, as expressions containing these properties make up only small section of the data (4.1% of the +LOC condition and 2.0% of the -LOC condition).

On the other hand, there are more complications for handling location in this corpus: In the GRE3D3 corpus, locations were kept constant for each of the scenes. In the TUNA corpus, there is no location constancy; subjects presented with the same 'scene'

```
01 func main(scene, referent_id):
02   obj = scene[referent_id]
03   r = refer(obj, scene)

04 func refer(obj, scene):
05   r = <>
06   cat = KB.find_category(obj['TYPE'])
07   r = analyze_SP(obj, scene, r)
08   r = analyze_CP(obj, scene, r)
09   r = incremental_obj(obj, scene, r)
10   r += <obj['TYPE']>
11   return r

12 func analyze_SP(obj, scene, r):
13   att = 'COLOR'
14   val = obj[att]
15   r += check_interconn(obj, scene, att, val, r)
16   r += add_lemma(att, val, length(r))
17   for att ∈ <'SIZE', 'LOCATION', 'ORIENTATION'>:
18     if att == 'SIZE':
19       rx = obj['WIDTH'], ry = obj['HEIGHT']
20       dx = average width, other scene objects
21       dy = average height, other scene objects
22       val = SizeMod(rx, ry, dx, dy)
23       r += add_lemma(att, val, 0)
24     else:
25       val = obj[att]
26       r += add_lemma(att, val, length(r))
27   return r

28 func check_interconn(obj, s, att, val, r):
29   i = <>
30   for i_att of KB.interconnected(att):
31     ival == obj[att]
33     if KB.implies(att, val, i_att, ival):
34       i += add_lemma(i_att, ival, length(r))
35   return i

36 func analyze_CP(obj, scene, r):
37   for att ∈ CP:
39     val = obj[att]
43     r += check_interconn(obj, scene, att, val, r)
44     r += add_lemma(att, val, length(r))
45   return r

46 func incremental_obj(obj, scene, r):
47   for d in scene:
48     dobj = scene[d]
49     if dobj == obj:
50       continue
51     if dobj['TYPE'] != obj['TYPE']:
52       continue
53     for att ∈ CP ∪ SP:
54       if att == 'SIZE':
55         rx = obj['WIDTH'], ry = obj['HEIGHT']
56         dx = dobj['WIDTH'], dy = dobj['HEIGHT']
57         val = SizeMod(rx, ry, dx, dy)
58       else:
59         dval = dobj[att]
60         val = obj[att]
61         if dval != val:
62           l = add_lemma(att, val, length(r), dval)
63           if l not in r:
64             r += l
65   return r

66 func add_lemma(att, val, len, dval=None):
67   l = <>
68   if dval:
69     l = val
70   else:
71     if throw_dice(α_att, cat.β_val, len):
72       l = val
73   return l
```

FIGURE 6. Algorithm: How it works in TUNA domain. Lines that have no effect are in grey. In contrast to GRE3D3, ORIENTATION is used, and there are different typicality values for colors of the objects ($\mathsf{cat}.\beta_{val}$).

would see different spatial configurations (I will use the term TUNA 'scene' to refer to a specific collection of objects, without a specific spatial configuration), which makes it impossible to ascertain what human variation is for a *specific* arrangement of objects. Subjects were also split into different conditions, and in the -LOC they were made aware of the location property (discouraged from using it) (van Deemter et al., 2012), which

may prime participant responses. Location is also represented as two separate attributes, X-DIMENSION and Y-DIMENSION, and participant responses may be annotated such that just X-DIMENSION or just Y-DIMENSION appears in the attribute set for an observed expression – this complicates the training of Graph and comparison with the Visible Objects Algorithm, which instead processes a LOCATION attribute specifically (e.g., x and y coordinates are both treated as LOCATION features). These last two issues are discussed in further detail below.

The primary reason why I use this corpus is that the objects in TUNA are more complex than the geometric objects of the GRE3D3 domain (see Figure 1). The TUNA images are computer-generated images of furniture, and so the algorithms may perform differently, or reference may behave differently; for example, because furniture is a real world object, issues of typicality *may* come into play. I therefore move to the TUNA domain to test some of same aspects of the algorithms tested in GRE3D3 – COLOR and SIZE of computer-generated objects – but remove the LOCATION attribute, minimize the effect of RELATIVE LOCATION and descriptions of the relatum, and add typicality for the proposed algorithm, represented as likelihood values in the variable cat.β_{val} (line 71). This gives us a nice sense of the variation in the algorithms across somewhat similar corpora.

The list of hypotheses I aim to address using this corpus is given below. From the GRE3D3 hypotheses listed in Section 8.6, the TUNA corpus adds evaluation for (3), the typicality of the attribute's value for the object.

- COLOR and SIZE are selected independently of one another
- inclusion of an attribute in the identifying description is based on:
  (1) the description's length
  (2) the prior likelihood of including the attribute
  (3) the typicality of the attribute's value for the object
- stochastic inclusion of each attribute aids in generating the distribution of expressions observed in human data

Incremental and Graph-Based Algorithm TUNA Scene Input:

Object1	colour:grey	size:large	type:desk	x-dimension:3	y-dimension:1 orientation:front
Object2	colour:blue	size:large	type:desk	x-dimension:2	y-dimension:1 orientation:front
Object3	colour:red	size:large	type:desk	x-dimension:3	y-dimension:2 orientation:back
Object4	colour:green	size:small	type:desk	x-dimension:4	y-dimension:1 orientation:left
Object5	colour:blue	size:large	type:fan	x-dimension:1	y-dimension:1 orientation:front
Object6	colour:red	size:large	type:fan	x-dimension:5	y-dimension:1 orientation:back
Object7	colour:green	size:small	type:fan	x-dimension:2	y-dimension:2 orientation:left

Simplified Visible Objects Algorithm TUNA Scene Input:

Object1	colour:grey	size:(454,454)	type:desk	location:(3,1)	orientation:front
Object2	colour:blue	size:(454,454)	type:desk	location:(2,1)	orientation:front
Object3	colour:red	size:(454,454)	type:desk	location:(3,2)	orientation:back
Object4	colour:green	size:(254,254)	type:desk	location:(4,1)	orientation:left
Object5	colour:blue	size:(454,454)	type:fan	location:(1,1)	orientation:front
Object6	colour:red	size:(454,454)	type:fan	location:(5,1)	orientation:back
Object7	colour:green	size:(254,254)	type:fan	location:(2,2)	orientation:left

FIGURE 7. Example input to the algorithms: TUNA Scene 2.

8.7.2. Preparing the Algorithms. The TUNA objects may be perceived by a speaker to have a typical COLOR, as well as typical values for the attributes that require comparison processes: typical SIZE within a room, LOCATION within a room, and ORIENTATION within a room. On the other hand, because the scenes do not offer a room context but are instead cut-out images against a white background, the properties that require comparison properties may not to be clearly affected by typicality expectations (or may all be seen as atypical).

Typicality may come into play more clearly for COLOR. It may not – furniture items have been identified specifically as *low color diagnostic* objects (Tanaka & Presnell, 1999); color does not strongly influence their recognition, and people tend not to list typical colors when generating property lists for furniture. Further, the visual salience of object color due to its contrast against the white background will be quite high, and this is a conflating factor. To more clearly examine the effect of COLOR typicality in the Visible

Objects Algorithm, I therefore compare two versions of the algorithm. One version uses likelihood estimates for typical color values, and one does not.

The annotation of location in the TUNA corpus also poses a problem for algorithm training, particularly for Graph, as discussed above. In TUNA, location is represented as two attributes, X-DIMENSION and Y-DIMENSION, and participant responses may be marked such that just X-DIMENSION or just Y-DIMENSION is annotated in an observed expression. Applying this in a straightforward way to the training stage of Graph, this means that each x-dimension is weighted separately from each y-dimension, and we immediately run into data sparsity issues for determining costs for these properties.

This also requires that the x-dimension of a scene be selected separately from the y-dimension during reference, and places the proposed algorithm – which explicitly requests that both coordinates be analyzed simultaneously as LOCATION – at a possible advantage.[5] Due to these issues, I remove LOCATION and the related attributes of X-DIMENSION and Y-DIMENSION from the evaluation. Although all algorithms have access to these attributes and may use them in their decision processes, the algorithms are not evaluated on their selection of these attributes.

For development on this corpus, I randomly select 2 scenes (scenes 1 and 2). I find that including TYPE at the start or end of the Incremental Algorithm's preference order has no effect on its accuracy, and that setting different values in the proposed algorithm for g above 5 in the throw_dice function (line 75) appears to have little effect.

To set typicality values for the Visible Objects Algorithm in this domain, we need a database that provides information about how likely it is for, e.g., a chair to be red. In Chapter 5, I used McRae's norms to determine the frequency with which a property is mentioned for an object, and ideally these frequencies may be used as a measure of

[5]Note that the level of abstraction for LOCATION represented as (x,y) axes is similar to the level of abstraction for SIZE represented as (height,width) lengths. My view is that LOCATION, like SIZE, should be treated as a multidimensional vector defining various distances between the object's top/bottom/sides and the rest of the scene, and the problem is therefore to return the best LOCATION type given the vector space.

sofa	colour:red :0.1	colour:orange:0.0333	colour:yellow:0.0333
	colour:green:0.1000	colour:blue :0.0667	colour:purple:0.0
	colour:pink :0.0333	colour:black :0.1000	colour:brown:0.0333
	colour:grey :0.0667	colour:white :0.4333	
fan	colour:red :0.0333	colour:orange:0.0	colour:yellow:0.0
	colour:green:0.0	colour:blue :0.0667	colour:purple:0.0
	colour:pink :0.0	colour:black :0.2667	colour:brown:0.0667
	colour:grey :0.4333	colour:white :0.1333	
desk	colour:red :0.0	colour:orange:0.0	colour:yellow:0.0
	colour:green:0.0	colour:blue :0.0	colour:purple:0.0
	colour:pink :0.0	colour:black :0.1	colour:brown:0.7
	colour:grey :0.0667	colour:white :0.1333	
chair	colour:red :0.3	colour:orange:0.0333	colour:yellow:0.0
	colour:green:0.0	colour:blue :0.0667	colour:purple:0.0333
	colour:pink :0.0	colour:black :0.1667	colour:brown:0.3
	colour:grey :0.0	colour:white :0.1	

TABLE 10. Typicality values for COLOR attributes of objects in TUNA domain. Numbers correspond to the relative frequency out of the top 30 images in a Google image search for the object.

the typicality of the property; however, likely owing to the fact that these are low color diagnostic objects (Tanaka & Presnell, 1999), people did not list typical colors for any of the TUNA furniture objects. Therefore, I establish typical colors for the objects by performing a Google image search for each object, and calculating the relative color frequencies in the first 30 image results. These values are shown in Table 10.

The Visible Objects Algorithm tested in this domain can be represented in the simplified form shown in Figure 6. Portions of the algorithm that have no effect are colored in grey.

8.7.3. 1: Evaluation by Alignment (MaxAlign). I again use five-fold cross-validation on the five test scenes, setting α values for the proposed algorithm, the PO for the Incremental Algorithm, and the costs for Graph from the training data in each fold. For each test scene, I again run the proposed algorithm five times, taking the average maximum alignment score over all five runs.

TUNA		
ALGORITHM	+LOC	-LOC
Proposed Algorithm - Typicality	87.54%	84.79%
Proposed Algorithm - No Typicality	88.11%	85.35%
IA	80.00%	79.14%
Graph	68.57%	66.38%

TABLE 11. Average Maximum Alignment (Accuracy) on TUNA corpus.

Results are shown in Table 11. Again we see that the proposed Visible Objects Algorithm outperforms the IA and Graph. Interestingly, the version of the algorithm that does not use typicality likelihoods performs better than the version that does. This may be because the method for determining typicality likelihood is not a good method; this may also be because typicality does not play a significant role in reference to visible objects, which is also in line with the findings for MATERIAL in Chapter 5. It is also possible that the TUNA domain is not natural enough for issues of typicality to come into play. Such an option is addressed in the next evaluation, where I use the Typicality corpus of real world objects. We may also be able to approximate typicality in a better way; here I am trying just one method, estimating typicality using maximum likelihood estimation on Google search images.

Graph performs remarkably poorly, and this may be due to the data sparsity issue that arises when requiring the algorithm to train on attribute-values. Following previous work (Theune et al., 2011; Koolen et al., 2012, see Section 8.4.2), weights are based on the frequency of previously seen attribute-value pairs, as opposed to attributes alone. In development, I find that some attribute-value pairs receive the higher weight and are placed further down in the preference order because they are not seen, or are seen infrequently; testing on a new (unseen) scene, an attribute-value rare in training becomes significant in the test, and Graph's preference not to include it significantly hurts its accuracy.[6]

[6]Note it is also possible to use a graph-based approach where attributes alone form the weights; along with the preference order put in place in Viethen et al. (Viethen et al., 2008), this essentially becomes a graph-based implementation of the Incremental Algorithm.

8.7.4. 2: Evaluation of Majority (Maj). As in the GRE3D3 corpus, I use the TUNA scenes in five-fold cross-validation, estimating parameters on the four training scenes in each fold, and for each test scene, I run the proposed algorithm 1,000 times. I report the percentage of folds where the majority attribute set between the observed and predicted data match. Results are shown in Table 12.

TUNA		
ALGORITHM	+LOC	-LOC
Proposed Algorithm - Typicality	40.00%	40.00%
Proposed Algorithm - No Typicality	40.00%	40.00%
IA	0.00%	100.00%
Graph	20.00%	20.00%

TABLE 12. Percentage of scenes where most frequently predicted expression matches most frequently observed expression.

The Visible Objects Algorithm is relatively stable across conditions, predicting the majority attribute set in 40% of the test scenes. It does not outperform the IA in the -LOC condition, but the IA has a large range across the two conditions (0% and 100%).

8.7.5. 3: Frequency Prediction (FreqPred). In this domain, we again see that the distribution modeled by the Visible Objects Algorithm does not match the human corpus very well. The algorithm does not predict all of the seen attribute sets in any scene; all probabilities are therefore 0.0. I report values from a randomly selected scene as an example of the difference in distribution modeled by the algorithm and the frequencies of human-produced attribute sets in Tables 13 and 14.

Examining the mistakes made by the algorithm in this corpus, we see that the algorithm does not often predict any attribute without also predicting COLOR. This makes the algorithm incapable of predicting expressions that do not include COLOR (but include other attributes); for example, the algorithm does not predict the attribute set from the 6th row of Table 14, tg:orientation:front, tg:type:desk. This is one area where *visual salience* may play a role, which I have left unimplemented in this evaluation and leave for future

Predicted	Freq.	Observed	Freq.	Ex. Human Expression
tg:colour:green tg:size:small tg:type:desk	406 40.60%	tg:colour:green tg:size:small tg:type:desk	3 10.35%	*green, smaller desk*
tg:colour:green tg:type:desk	231 23.10%	tg:colour:green tg:type:desk	3 10.35%	*green desk*
tg:colour:green tg:orientation:front tg:type:desk	82 8.20%	tg:colour:green tg:orientation:front tg:type:desk	2 6.90%	*a green desk facing forwards*
tg:colour:green tg:orientation:front tg:size:small tg:type:desk	81 8.10%	tg:colour:green tg:orientation:front tg:size:small tg:type:desk	10 34.48%	*small green desk facing forward*
tg:colour:green tg:location:(2,1) tg:type:desk	91 9.10%			
tg:colour:green tg:location:(2,1) tg:size:small tg:type:desk	79 7.90%			
tg:colour:green tg:location:(2,1) tg:orientation:front tg:type:desk	22 2.20%			
tg:colour:green tg:location:(2,1) tg:orientation:front tg:size:small tg:type:desk	8 0.80%	(*More observed expressions in continued table below*)		

Continued in Table 14 . . .

TABLE 13. TUNA Scene 6, FreqPred match.
$p(x|d,n) = 0.0$. Matching attribute sets shown in red.

work. Experimenting with different stochastic functions may also bring the algorithm's output closer to a reasonable distribution.

8.8. Evaluation 3: Typicality Corpus

8.8.1. The Corpus. In these evaluations, I examine how well the algorithms fare in a corpus of real world objects, using the objects tested in Chapter 5 (bowls, boxes, envelopes, keys, mugs, rulers, and screws). This brings in further complex properties – TEXTURE, MATERIAL, SHAPE – as well as a real world setting and an overarching goal for the participant; subjects were not instructed explicitly to refer, but instead instructed to give directions to another person on how to re-create each arrangement of objects. This

... continued from Table 13.

	tg:colour:green tg:type:desk tg:x-dimension:4 tg:y-dimension:3	1 3.45%	*green desk left of the desk*
	tg:colour:green tg:orientation:front tg:type:desk tg:x-dimension:3	1 3.45%	*middle frontal green table*
	tg:colour:green tg:type:desk tg:y-dimension:1	1 3.45%	*top green desk*
	tg:colour:green tg:size:small tg:type:desk tg:y-dimension:1	1 3.45%	*smallest green desk on the top row*
	tg:type:other tg:x-dimension:4 tg:y-dimension:3	1 3.45%	*third picture on third row*
	tg:orientation:front tg:type:desk	1 3.45%	*a desk with the cupboards facing me*
	tg:type:desk tg:x-dimension:4 tg:y-dimension:3	1 3.45%	*3rd desk last row*
	tg:colour:green tg:size:small tg:type:desk tg:x-dimension:5 tg:y-dimension:2	1 3.45%	*little green table center right*
	tg:colour:green tg:type:desk tg:y-dimension:3	1 3.45%	*the green desk in the bottom row is in red box*
	tg:colour:green tg:orientation:front tg:type:desk tg:y-dimension:3	1 3.45%	*the green desk facing me in the bottom row*
	tg:x-dimension:5 tg:y-dimension:1	1 3.45%	*top right*

TABLE 14. TUNA Scene 6, FreqPred match.
$p(x|d,n) = 0.0$. Matching attribute sets shown in red. (Continued from Table 13.)

lessens experimental effects caused by participants knowing the purpose of the study. I hope that expressions collected in this domain are more clearly affected by typicality expectations, since the objects are real objects in real scenes (see Figure 8). Examples of the input for all algorithms are given in Figure 9. The list of hypotheses I aim to address using this corpus is given below:

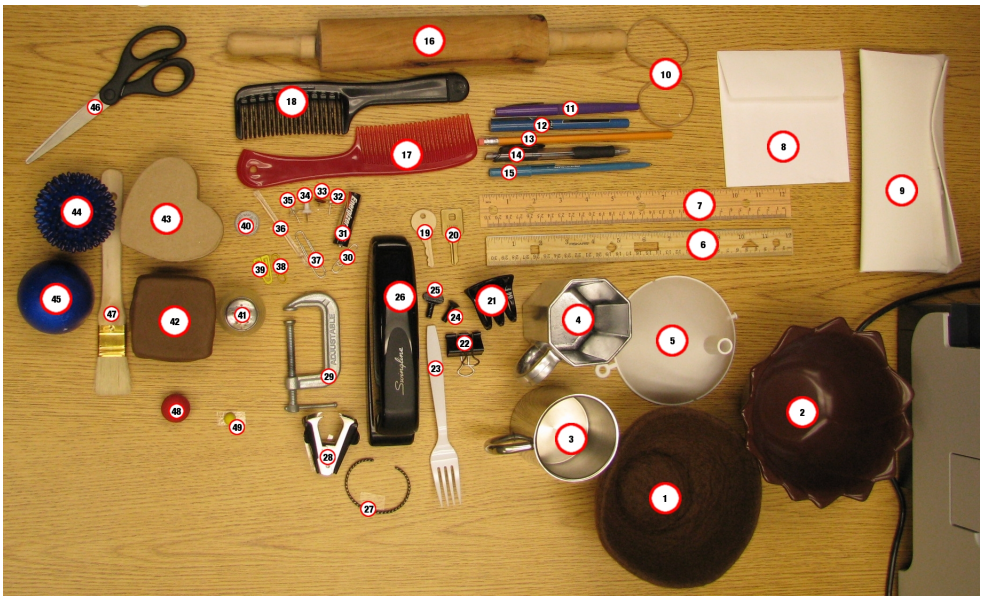


FIGURE 8. Items from typicality study.

Incremental and Graph-Based Algorithm Input:

Object1	colour:brown	size:medium	shape:flower	opacity:3
	sheen:2	material:ceramic	form:smooth	type:bowl
	location:bottom	orientation:rightside-up	texture:spiky	
Object2	colour:brown	size:medium	shape:round	opacity:3
	sheen:3	material:cloth	form:hairy	type:bowl
	location:bottom	orientation:rightside-up	texture:coarse	

Simplified Visible Objects Algorithm Input:

Object1	colour:brown	size:(10,20)	shape:flower	opacity:3
	sheen:2	material:ceramic	form:smooth	type:bowl
	location:bottom	orientation:rightside-up	texture:spiky	
Object2	colour:brown	size:(10,20)	shape:round	opacity:3
	sheen:3	material:cloth	form:hairy	type:bowl
	location:bottom	orientation:rightside-up	texture:coarse	

FIGURE 9. Example input to the algorithms: Typicality.

- COLOR and SIZE are selected independently of one another
- a given attribute value can bring in interconnected attributes that should be considered before progressing – specifically tested for the attribute of COLOR and the interconnected attribute MATERIAL.
- inclusion of an attribute in the identifying description is based on:

(1) the description's length

(2) the prior likelihood of including the attribute

(3) the typicality of the attribute's value for the object

- stochastic inclusion of each attribute aids in generating the distribution of expressions observed in human data

8.8.2. Preparing the Algorithms. For an input scene, I define properties of the target referent and the object of the same type that appears next to it. Values for OPACITY and SHEEN are written on a scale of 1 to 3, with 3 being completely opaque-/completely shiny. Further work will need to refine these representations and develop guidelines for mapping real world objects to attribute-value representations. The current annotation is a first-pass for now to understand how the algorithms perform in a real world domain.

8.8.3. 1: Evaluation by Alignment (MaxAlign). As in the TUNA domain, the Incremental Algorithm performs identically with TYPE placed at the beginning or at the end of the PO, and so I report one set of values. All lines of the Visible Objects Algorithm written in Figure 3 play a role in this domain. In contrast to the previous domains, complex properties (SHAPE, MATERIAL, etc.) are used in generating the attribute set, and interconnected properties (MATERIAL for COLOR) are analyzed by the Visible Objects Algorithm.

I do not use a development set to tune the weight on description length (g) in the algorithm, but use the value established for GRE3D3 and TUNA (5). Again, evaluation is performed using cross-validation, with the proposed algorithm averaged over five runs for each test scene.

Results are shown in Table 15. We again see that the algorithm outperforms the IA and Graph, and, interestingly, again see that including typicality likelihoods does not greatly increase algorithm accuracy. As can be seen in the case of MATERIAL, typicality as implemented lowers the algorithm's accuracy. This suggests that there may be a better

Typicality		
ALGORITHM	ATYPICAL SHAPE	ATYPICAL MATERIAL
Proposed Algorithm - Typicality	87.93%	84.32%
Proposed Algorithm - No Typicality	87.11%	85.10%
IA	83.67%	75.34%
Graph	75.17%	73.30%

TABLE 15. Average Maximum Alignment (Accuracy) on Typicality Corpus.

method for determining typicality, and/or a better method for implementing the affect of typicality in a stochastic algorithm. For now, disregarding property typicality results in an algorithm that produces a better match with human data for visible objects than the IA and Graph.

8.8.4. 2: Evaluation of Majority (Maj). As before, I use the seven test items in each condition (Atypical SHAPE or Atypical MATERIAL) in seven-fold cross-validation, estimating parameters on the six training scenes in each fold, and for each test scene, I run the proposed algorithm 1,000 times. I report the percentage of folds where the majority attribute set between the observed and predicted data match. Results are shown in Table 16.

Typicality		
ALGORITHM	ATYPICAL SHAPE	ATYPICAL MATERIAL
Proposed Algorithm - Typicality	28.57%	14.29%
Proposed Algorithm - No Typicality	42.86%	14.29%
IA	28.57%	0.00%
Graph	0.00%	0.00%

TABLE 16. Percentage of scenes where most frequently predicted expression matches most frequently observed expression.

The algorithm outperforms the other algorithms, and we again see that the typicality function does not help the algorithm better match human data – as seen in the Atypical SHAPE condition, the algorithm performs best without typicality.

8.8.5. 3: Frequency Prediction (FreqPred). In this domain, the algorithm does not predict all of the seen attribute sets in the Atypical MATERIAL condition (all probabilities are 0.0), although it does predict some complete attribute sets for folds (scenes) from the Atypical SHAPE condition. I report values from the best-scoring fold in the Atypical SHAPE condition and the corresponding scene in the Atypical MATERIAL condition to illustrate the difference in distribution between the algorithm's model and the human-produced attribute sets. These are shown in Tables 17 and 18.

Predicted	**Freq.**	**Observed**	**Freq.**	**Example Observed Human Expressions**
tg:shape:octagonal tg:type:mug	308 30.80%	tg:shape:octagonal tg:type:mug	1 8.34%	*the mug with... that is not round at the top, it's a hexagon*
tg:material:ceramic tg:shape:octagonal tg:type:mug	306 30.60%	tg:material:ceramic tg:shape:octagonal tg:type:mug	10 83.34%	*metal cup... the angled hexagonal or octagonal*
tg:colour:silver tg:type:mug	112 11.20%			
tg:colour:silver tg:material:ceramic tg:type:mug	100 10.00%			
tg:colour:silver tg:material:ceramic tg:shape:octagonal tg:type:mug	73 7.30%			
tg:colour:silver tg:shape:octagonal tg:type:mug	69 6.90%	tg:colour:silver tg:shape:octagonal tg:type:mug	1 8.34%	*the hexagonal silver cup*
tg:type:mug	16 0.16%			
tg:material:ceramic tg:type:mug	16 0.16%			

TABLE 17. Mug, Atypical SHAPE condition, FreqPred match. $p(x|d,n)$ = 2.0192e-05. Matching attribute sets shown in red.

Qualitatively examining the results, we see that in this domain, the algorithm predicts COLOR much more often than it is produced by people. It also tends to predict MATERIAL more often than people produce it, and SHAPE less often than people produce it. In the human-produced data, the two attributes are often both found in the attribute set, leading to overspecified phrases; however, for the two objects of the same type, the inclusion of one attribute (SHAPE or MATERIAL) annuls the inclusion of the other for

Predicted	Freq.	Observed	Freq.	Example Observed Human Expressions
tg:material:metal tg:type:mug	640 64.00%	tg:material:metal tg:type:mug	2 16.67%	*a tin cup*
tg:material:metal tg:shape:round tg:type:mug	271 27.10%	tg:material:metal tg:shape:round tg:type:mug	8 66.67%	*the round tin cup*
tg:colour:silver tg:material:metal tg:type:mug	67 6.70%			
tg:colour:silver tg:material:metal tg:shape:round tg:type:mug	22 2.20%			
		tg:location:bottom tg:shape:round tg:type:mug	1 8.34%	*the circular um drinking cup here*
		tg:colour:silver tg:shape:round tg:type:mug	1 8.34%	*the silver round cup*

TABLE 18. Mug, Atypical MATERIAL condition, FreqPred match. $p(x|d,n) = 0.0$. Matching attribute sets shown in red.

unique identification. Perhaps complex properties such as MATERIAL and SHAPE are processed in parallel, not incrementally, which would be one way to account for the observed overspecification.

As before, we see that the distribution modeled from the algorithm is not likely to produce the observed human corpus, and unfortunately, fails to predict some attribute sets produced by people. Experimenting with different stochastic functions may bring the algorithm's output closer to a reasonable distribution.

8.9. Discussion

I have evaluated several aspects of the Visible Objects Algorithm:

- COLOR and SIZE are selected independently of one another
- a given attribute value can bring in interconnected attributes that should be considered before progressing – specifically tested for the attribute of COLOR and the interconnected attribute MATERIAL.

- inclusion of an attribute in the identifying description is based on:
  (1) the description's length
  (2) the prior likelihood of including the attribute
  (3) the typicality of the attribute's value for the object
- stochastic inclusion of each attribute aids in generating the distribution of expressions observed in human data

We have found evidence to suggest that the proposed algorithm performs as well as or better than the state of the art for generating human-like descriptions of visible objects, using the given implementations. It achieves a good alignment to the observed human data, reaching accuracy of 85.00% or higher across corpora, and predicts the most frequently observed attribute set from the human data more consistently than either the implementation of the Incremental Algorithm or the implementation of the Graph-Based Algorithm. It reaches majority agreement of 50% and higher in the GRE3D3 corpus, around 40% in the TUNA corpus and the ATYPICAL SHAPE condition of the Typicality corpus, and around 14% in the ATYPICAL MATERIAL condition of the Typicality corpus (compared to the IA and Graph's 0% on this corpus).

The evaluated versions of the IA and Graph, which assume a literary model of reference (attempting to uniquely identify the referent by ruling out all competitor objects; see Chapter 2), lag the proposed algorithm across trials on GRE3D3 and TUNA. The exception is the IA, evaluated on the TUNA domain, using Majority agreement: In the +LOC condition, the IA predicts 0% of the observed majority expressions, while in the -LOC condition, the IA predicts 100% of the observed majority expressions. Here, the proposed algorithm predicts 40% for -LOC and for +LOC. Although it does not do as well as the IA in the -LOC condition, it is more stable across both.

The relatively strong performance of the proposed algorithm is especially interesting because these corpora were created in different modalities. In the GRE3D3 corpus and the TUNA corpus, speakers typed their answers, and there was no hearer present. In

the Typicality corpus, reference was made verbally, and a hearer was present. It is interesting that the approach taken by the proposed algorithm, which assumes a verbal/conversational modality and generates descriptive reference, performs better across all three domains. This suggests that people may be using a verbal model when they identify visual referents. This appears to be true even when there is no hearer present, and even when participants are typing.

However, it is clear that there is more work to be done. We have not come close to predicting the frequencies of expressions produced by people. This may be forgiven if people are so varied that any random set of people will have an unpredictable distribution of attribute sets; however, the algorithm does not even successfully predict all types of attribute sets in the furniture domain of the TUNA corpus or the Atypical MATERIAL condition of the Typicality corpus. One aspect of the algorithm that should clearly be varied, and may help improve the distribution predicted by the algorithm, is the stochastic throw_dice function. I have tried one function that takes into account typicality, description length, and prior attribute likelihood; there are clearly more ways than one to combine these factors in a stochastic function, and this should be further explored.

A related issue is that I have not implemented parallelism. In Chapter 7 I suggested two parallel pathways, interacting as the selection of properties in one path affects the length penalty in the other. Here, I have here taken a serial approach, with the simplification that COLOR and SIZE are selected independently. With richer corpora, and timing of expressions, we may be able to further understand whether parallel processing is a reasonable approach for REG.

I have also not connected this algorithm directly to a visual system output, using "gold-standard" attribute values for all properties except for SIZE. Analysis of the errors for SIZE suggests that further work should look into how to define the comparison set from which the size's semantic form is derived.

Other areas for future work include the implementation of typicality and interconnected properties. I have only looked at typicality values for COLOR, and have not seen improvement in the algorithm's accuracy. There may be a better method for determining typicality, and/or a better method for implementing the affect of typicality in a stochastic algorithm. I have limited the examination of interconnected properties to COLOR and MATERIAL, and clearly evaluation of interconnected properties should be further expanded before any conclusions can be made about the utility of this kind of functionality.

From the previous evaluations, we have seen that analyzing COLOR and SIZE independently while stochastically adding attributes based on description length and prior likelihood (rather than discriminatory power) may lead to more naturalistic output for reference to visible objects. Generating initial reference that is *descriptive*, using properties based on prior likelihoods and focused primarily on generating what is salient for our visual processing system, leads to a better match with human data. This suggests that people may be characterized as generating *identifying descriptions* rather than *distinguishing descriptions* when introducing a visual referent into discourse. There is still a lot of work to do to bring the distribution predicted by the algorithm closer to the distribution of observed expressions; but in contrast to earlier work, we can actually begin to capture speaker variation.

Chapter 9

Conclusions and Future Work

9.1. Overview

The research presented in this thesis has looked at how speakers refer to objects in visual domains, with the goal of conveying an intended referent to a hearer who may view the same scene. I have examined how initial reference may be descriptive, including properties because they are visually or linguistically interesting, and how speakers are varied, producing many kinds of output, but with preferences for particular attributes like color and size. I have presented several models for how to connect a visual input to a human-like referential output.

By focusing specifically on the domain of real world, visible objects, I hope to come slightly closer to understanding what humans do when they refer in visual domains, and how we can begin mimicking the kinds of visual language that humans produce. Work in this thesis may be used to further the connection between vision and language, aiding in the ability of a system to automatically generate summaries, captions, or descriptions from images. This will hopefully be beneficial to improve the state of the art in a variety of tasks, such as image indexing, vision-based web searches, communication robots, and assistive NLG technology.

The main contribution that I hope to make is to question the philosophy that *unique identification* and *deterministic algorithms* are the best we can do to generate human-like output. Rather than producing one referring expression (and only one) and stopping once a target item has been uniquely identified (or else fail), I have attempted to get closer to the variation that humans have in referring to the same object by better understanding

the variety of choices that people make when referring, and creating an algorithm that models this by producing *several* referring expressions non-deterministically. This method does not aim for unique identification, but instead the likelihood of including properties diminishes as less salient properties are considered and the description becomes longer and more complex. This is a natural way to account for the phenomena of underspecification and overspecification common to human references.

The basic approach taken in this thesis was to conduct a series of psycholinguistic studies regarding specific aspects of visible reference, in each building an annotated corpus of human reference to real world objects. Findings from these experiments were used to inform the design of a full REG algorithm that generates human-like reference to visible objects, and I demonstrate that it performs as well as or better than existing approaches for several corpora and using several metrics.

9.2. Summary

In **Chapter 1**, I provide an introduction to natural language generation, referring expression generation, human vision, and computer vision. In **Chapter 2**, I review previous research relevant for this thesis, including the philosophy of reference (what referring *is* and what it *does*), the psychology of reference (how referring *works*) and computational approaches to reference (how referring can be modeled). I also summarize models of visual processing and recent work on object detection in computer vision and what we may learn from this research.

In **Chapter 3**, I discuss an exploratory study on initial reference without specific hypothesis testing. I sought to understand how individuals, given an assortment of visually diverse objects, would initially refer to each in a monologue setting.

Several interesting findings come out of this study. One is the relative predominance of COLOR modifiers, also found in earlier work. Another is how people refer to object sizes, which clearly shows that people compare the target object dimensions against the

dimensions of other objects. We additionally see evidence of *part-whole modularity* and *analogies*, neither of which have received a great deal of attention in work on REG.

From these findings, I suggested several structures that may be useful in generating natural reference: (1) a spatial representation defining object height, width, depth, and relations between object parts, (2) a propositional representation that provides information about COLOR, MATERIAL, TEXTURE, etc., and (3) a knowledge base with representations for typical object properties. Using structures that define the propositional and spatial content of objects fits well with work in psycholinguistics, cognitive science and neurophysiology discussed in Chapter 2, and may provide the basis to generate a variety of natural-sounding references from a system that recognizes objects.

The spatial representation proposed in Chapter 3 is simplified to the x- and y-axes of objects in **Chapter 4**, where I address specific hypotheses regarding subjects' references to object SIZE. In a large-scale study, I elicited references to four different object types each in 24 different size configurations, and annotated the references to create a large corpus of SIZE-based referring expressions. Using the size of x- and y-axes of the objects as the basis for several SIZE features, I propose both a hand-written and a machine-learning approach to generate six broad SIZE types based on this data. These include forms for words like "tall", "thin", "big", "small", etc., picking out different relationships between the two axes.

Both the hand-written algorithm and the machine learning approach work well on the corpus created in this chapter, and even better on a novel corpus, the Craft Corpus introduced in Chapter 3. Using the latter corpus requires tweaking the approach to generating SIZE a bit, taking the height and width *average* of other items of the same type, which is inspired by how people initially view and process the size of objects in a scene. This method works well, and the size algorithm is later incorporated into a full REG algorithm in Chapter 7.

In **Chapter 5**, we look at the properties of SHAPE and MATERIAL, examining how they interact when *atypical* and *typical* for an object. In a study conducted in-person, real world objects were presented to subjects in a director-matcher paradigm. We see some evidence that atypical SHAPE tends to be mentioned in reference to visible objects, but the findings are less clear for MATERIAL.

One issue that arises in this work is the issue of the *interconnectedness* of different properties. Most strikingly, the materials of *wood* and *metal* were more common to include when referring to an object than their corresponding colors; some materials *imply* color, and this seems to be a factor in the kinds of references that people produce.

In **Chapter 6**, I briefly examine how common COLOR was across the corpora I built in Chapters 3, 4, and 5. In the highly diverse Craft Corpus and the relatively uniform fillers sub-corpus of the Size Corpus, we see that COLOR is exceedingly preferred. In the even more uniform Size Corpus and the Typicality Corpus, we see less evidence that COLOR is a preferred attribute, instead seeing a predominance of the attributes being studied in building each corpus.

The commonality between the Size Corpus and the Typicality Corpus that may explain this difference is that both corpora had objects of the *same type* with the *same color*, while objects in the Craft Corpus and the fillers sub-corpus of the Size Corpus had objects of the *same type* with *different colors*. This suggests that in addition to the pre-attentive role that COLOR may have in visual reference, comparison processes of a target object against another of the same type may also play role.

In **Chapter 7**, I use previous research discussed in Chapters 1 and 2 to detail the structures that an ideal computer vision front-end should provide for an REG algorithm that generates descriptions of objects.

Given this input, I use ideas in the literature and the findings from Chapters 3, 4, 5, and 6 to introduce a new approach to referring expression generation. The algorithm introduced in this chapter is built to be *non-deterministic* in order to capture naturalistic human

variation, using prior likelihoods for the inclusion of each property in the description and different mechanisms for different visual properties.

In this algorithm, COLOR, because it is one of the most frequently named visual attributes, one of the most basic visual properties, and plays a role in guiding attention, has a privileged status, and interacts with interconnected MATERIAL properties. SIZE, one of the most common visual properties after COLOR, is analyzed based on the dimensions of the objects in the scene to create a variety of size modifier types. Both COLOR and SIZE operate independently. The goal of the algorithm it *not* to rule out distractors in the visual scene; rather, it uses what we have learned about the properties that people use to describe to create an identifying description. Modifiers are decreasingly likely the longer the description is, reflecting previous findings that noun phrases rarely have more than three adjectives, and suggestive of the cognitive load that constructing and uttering longer descriptions requires.

The algorithm is evaluated in **Chapter 8**. In this chapter, I use several well-known corpora in REG, the GRE3D3 corpus and the TUNA corpus, and evaluate against top-performing implementations of both the Graph-Based Algorithm and the Incremental Algorithm. Using an alignment method, a majority match method, and a distributional comparison method, I show that the algorithm I have introduced is competitive with the state of the art. The mechanisms used in the algorithm will therefore hopefully further the breadth of the field by suggesting an approach to REG that is robust across visual domains, and creates several different kinds of human-like expressions, with different distributions, for visible objects.

9.3. Implications and Future Work

One clear area for further research is connecting my approach to generating object descriptions to the process of describing the spatial relations between objects. In particular, incorporating work on generating topological or projective relations (Kelleher & Kruijff,

2006) would allow us to capture almost all of the fundamental aspects of reference to visible objects; coupling descriptions of each object to descriptions of the spatial relations between them would also help us to further describe entire scenes.

Chapter 4 demonstrated that incorporating speaker identity into classification significantly improved accuracy. Further experiments on size not reported in this thesis demonstrated that it may be able to group speakers based on the dimensional features that best predict size modifier preference. This suggests that generating human-like language can be improved by building models for particular speaker clusters. In a system that generates natural language, these models can be constructed as speaker 'profiles' that follow different language behavior depending on the goals of the system.

Adding further modalities to this research (e.g., touch and pointing) may even better capture the kinds of reference that people produce in real world scenes, where they may be able to physically manipulate the objects. This may be particularly true for describing properties that are both visual and tactile, such as material (Chapter 5) and texture.

In Chapter 8, I used a probability density function as a way to examine how well the distribution over output expressions produced by the visible objects algorithm predicted the frequency of human-produced expressions. This method used no smoothing, which meant that when an observed expression was not predicted by the algorithm, there was no way to match the two sets (the probability was 0). In future work, I hope to fine-tune this evaluation metric, adding some smoothing to allow better comparisons between what the algorithm predicts and what we observe in human data.

Speaking further to this point, the stochastic function I introduce in Chapter 7 is a first attempt, and should be refined in future work. I have tried one function that takes into account typicality, description length, and prior attribute likelihood; there are clearly more ways than one to combine these factors in a stochastic function, and this should be further explored.

Other areas for future work in this vein include the implementation of typicality and interconnected properties in the algorithm in Chapters 7 and 8. I looked at defining typicality values for COLOR, and did not see improvement in the algorithm's accuracy. There may be a better method for determining typicality, and/or a better method for implementing the affect of typicality in a stochastic algorithm. I have limited the examination of interconnected properties to COLOR and MATERIAL, and clearly evaluation of interconnected properties should be further expanded before any conclusions can be made about the utility of this kind of functionality.

Similarly, I have addressed a simple approximation of parallelism in the algorithm, assuming that the selection of color does not affect the selection of size (and vice versa). It would be interesting to explore parallel generation in greater depth, for example, with different speeds for different parallel property analyses affecting the output description.

I also hope to connect the approaches discussed here directly to computer vision output in future work; I have discussed generation from an 'ideal', gold-standard visual input, but have not here explored generation from a computer vision output. As part of this goal, I have constructed an end-to-end vision-to-language system not discussed in this thesis (Mitchell, van Deemter, & Reiter, 2011), but have not yet provided functionality for the fine-grained referential preferences discussed in this thesis.

I hope that ideas from this thesis will be useful in further work on referring expression generation. In particular:

(1) The gap between reference and description is not that wide. Reference can incorporate description, and developing algorithms that aim to match *what people describe* in a visual scene can lead to the generation of human-like reference.

(2) What we know about how the visual system works – and how it is likely to affect what we talk about – is a useful guide in developing a visual reference algorithm.

(3) Defining values for visual attributes as multi-featured vectors (rather than as a single point) aids in generating rich, natural variation for different visual properties. For example, the space that an object takes up in a visual input can be represented as a vector of height and width features, used to generate a wide range of natural-sounding SIZE descriptors for an object (*big*, *small*, *thin*, *fat*, *thick*...).

(4) Probabilistically generating lemmas for each visual property of an object based on (a) the prior likelihood of that property being mentioned, and (b) the number of lemmas created before the current property is processed, may lead to human-like variation in the kinds of expressions produced.

(5) COLOR and SIZE should have a privileged status in reference in visual domains, and processes that construct modifiers for each may operate independently of one another. Such independence may be one of the explanations for the amount of *overspecification* one finds in referring expressions.

(6) A knowledge base of what is typical about objects may be used to guide analogies and what is remarkable in a target object.

(7) When people describe entire scenes, many of the principles that we have uncovered still apply.

I have demonstrated many of the principles underlying initial reference to objects in a visual domain. With this research, we can now begin to automatically refer to visible objects in a way that sounds human-like and natural.

References

Amazon. (2011). *Amazon mechanical turk: Artificial artificial intelligence.* https://www.mturk.com/mturk/.

Anderson, B. L. (2011). Visual perception of materials and surfaces. *Current Biology*, *21*(24), 978–983.

Appelt, D. E. (1981). *Planning natural language utterances to satisfy multiple goals.* Unpublished doctoral dissertation, Stanford University.

Appelt, D. E. (1985). Planning English referring expressions. *Artificial Intelligence*, *26*, 1-33.

Appelt, D. E., & Kronfeld, A. (1987). A computational model of referring. *Proceedings of the 10th International Conference on Artificial Intelligence*, 640–647.

Areces, C., Koller, A., & Striegnitz, K. (2008). Referring expressions as formulas of description logic. *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, 42–29.

Ariely, D. (2001). Seeing sets: Representation by statistical properties. *Psychological Science*.

Aristotle. (335 BCE). The poetics, 21.

Arts, A., Maes, A., Noordman, L., & Jansen, C. (2011). Overspecification in written instruction. *Linguistics*, *49*, 555–574.

Bard, E. G., Anderson, A. H., Sotillo, C., Aylett, M., Doherty-Sneddon, G., & Newlands, A. (2000). Controlling the intelligibility of referring expressions in dialogue. *Journal of Memory and Language*, *42*, 1–22.

Bard, E. G., Hill, R., Arai, M., & Foster, M. E. (2009). Accessibility and attention in situated dialogue: Roles and regulations. *Proceedings of the Workshop on the Production of Referring Expressions (PRE-CogSci 2009)*.

Bard, E. G., Hill, R., & Foster, M. E. (2008). What tunes accessibility of referring expressions in task-related dialogue? *Proceedings of the Thirtieth Annual Meeting of the Cognitive Science Society (CogSci 2008)*.

Bartlett, E. J. (1976). Sizing things up: The acquisition of the meaning of dimensional adjectives. *Journal of Child Language*, *3*(02), 205–219.

Belke, E., & Meyer, A. S. (2002). Tracking the time course of multidimensional stimulus discrimination: Analysis of viewing patterns and processing time during same-different decisions. *European Journal of Cognitive Psychology*, *14*(2), 237–266.

Belz, A., & Gatt, A. (2008). Intrinsic vs. extrinsic evaluation measures for referring expression generation. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008)*, 197–200.

Belz, A., Kow, E., Viethen, J., & Gatt, A. (2008). The GREC challenge: Overview and evaluation results. *Proceedings of the 5th International Conference on Natural Language Generation (INLG 2008)*, 183–191.

Berg, A. C., Berg, T. L., III, H. D., Dodge, J., Goyal, A., Han, X., et al. (2011). An exploration of how to learn from visually descriptive text. *JHU-CLSP Summer Workshop Whitepaper*.

Beun, R.-J., & Cremers, A. H. M. (1998). Object reference in a shared domain of conversation. *Pragmatics and Cognition*, *6*, 121–52.

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, *94*, 115–147.

Bierwisch, M., & Lang, E. (Eds.). (1989). *Dimensional adjectives : grammatical structure and conceptual interpretation*. New York: Springer-Verlag.

Bird, S., Loper, E., & Klein, E. (2009). *Natural language processing with python*. Sebastopol, CA: O'Reilly Media Inc.

Black, R., Waller, A., Reiter, E., Tintarev, N., & Reddington, J. (2011). "How was school today...?" A prototype system that uses a mobile phone to support personal narrative for children with complex communication needs. *Demo Session at the 2nd Workshop on Speech and Language Processing for Assistive Technologies (SLPAT 2011)*.

Blaser, E., Pylyshyn, Z. W., & Holcombe, A. O. (2000). Tracking an object through feature space. *Nature*, *408*, 196–199.

Bock, K., Eberhard, K. M., Cutting, J. C., Meyer, A. S., & Schriefers, H. (2001). Some attractions of verb agreement. *Cognitive Psychology*, *43*(2), 83-128.

Bock, K., & Levelt, W. J. M. (1994). Language production: Grammatical encoding. In M. A. Gernsbacher (Ed.), *Handbook of psycholinguistics* (pp. 945–984). San Diego: Academic Press.

Brennan, S. E., & Clark, H. H. (1996). Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*, 1482–93.

Brown, R. (1958). How shall a thing be called? *Psychological Review*, *65*, 14–21.

Brown-Schmidt, S., & Tanenhaus, M. K. (2006). Watching the eyes when talking about size: An investigation of message formulation and utterance planning. *Journal of Memory and Language*, *54*, 592–609.

Caramazza, A., & Shelton, J. R. (1998). Domain-specific knowledge systems in the brain: The animate-inanimate distinction. *Journal of Cognitive Neuroscience*, *10*(1), 1-34.

Caroll, D. W. (1999). *Psychology of language*. Pacific Grove, CA: Brooks/Cole Publishing Company. (Third edition)

Chapanis, A., Parrish, R. N., Ochsman, R. B., & Weeks, G. D. (1977). Studies in interactive communication: II. The effects of four communication modes on the linguistic performance of teams during cooperative problem solving. *Human Factors*, *19*, 101–125.

Clark, E. V. (1973). What's in a word? On the child's acquisition of semantics in his first language. In T. E. Moore (Ed.), *Cognitive development and the acquisition of language* (pp. 65–110). New York: Academic Press.

Clark, H. H., & Bangerter, A. (2004). Changing ideas about reference. In I. A. Noveck & D. Sperber (Eds.), *Experimental pragmatics* (pp. 25–49). Basingstoke, England: Palgrave Macmillan.

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

Clark, H. H., Schreuder, R., & Buttrick, S. (1983). Common ground and the understanding of demonstrative reference. *Journal of Verbal Learning and Verbal Behavior*, *22*, 245–258.

Clark, H. H., & Wilkes-Gibbs, D. (1986). Referring as a collaborative process. *Cognition*, *22*, 1–39.

Cohen, P. R. (1984). The pragmatics of referring and the modality of communication. *Computational Linguistics*, *10*(2), 97–146.

Cristy, J., Thyssen, A., & Weinhaus, F. (2011). *Imagemagick*. GNU General Public License. (http://www.imagemagick.org/www/conjure.html)

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detections. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*.

Dale, R. (1989). Cooking up referring expressions. *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL 1989)*.

Dale, R., & Haddock, N. (1991). Content determination in the generation of referring expressions. *Computational Intelligence*, *7*, 252-265.

Dale, R., & Reiter, E. (1995). Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, *19*, 233–263.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*.

Desai, C., & Ramanan, D. (2012). Detecting actions, poses, and objects with relational phraselets. *Proceedings of the 12th European Conference on Computer Vision (ECCV 2012)*.

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

Donnellan, K. (1966). Reference and definite description. *Philosophical Review*, *75*, 281–304.

Eilers, R. E., Oller, D. K., & Ellington, J. (1974). The acquisition of word-meaning for dimensional adjectives: the long and short of it. *Journal of Child Language*, *1*(02), 195–204.

Engelhardt, P. E., Bailey, K., & Ferreira, F. (2006). Do speakers and listeners observe the Gricean maxim of quantity? *Journal of Memory and Language*, *54*, 554–573.

Epshtein, B., & Ullman, S. (2007). Semantic hierarchies for recognizing objects and parts. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*.

Fabbrizio, G. D., Stent, A. J., & Bangalore, S. (2008). Trainable speaker-based referring expression generation. *Proceedings of the 12th Conference on Computational Natural Language Learning*, 151–158.

Fang, F., Boyaci, H., Kersten, D., & Murray, S. O. (2008). Attention-dependent representation of a size illusion in human V1. *Current biology*, *18*(21), 1707–1712.

Farhadi, A., Endres, I., Hoiem, D., & Forsyth, D. (2009). Describing objects by their attributes. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*.

Felzenszwalb, P. F., Girshick, R. B., & Mcallester, D. (2010). Cascade object detection with deformable part models. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*.

Ferreira, F., & Swets, B. (2002). How incremental is language production? evidence from the production of utterances requiring the computation of arithmetic sums. *Journal of Memory and Learning*, *46*, 57–84.

Ford, W., & Olson, D. (1975). The elaboration of the noun phrase in children's description of objects. *The Journal of Experimental Child Psychology*, *19*, 371–382.

Frassinelli, D. (2010). *The situated nature of concepts: Evidence from a property generation task*. Unpublished master's thesis, Università Di Pisa, Pisa, Italy.

Friedland, G., Jantz, K., & Rojas, R. (2005). SIOX: simple interactive object extraction in still images. *Proceedings of the 7th IEEE International Symposium on Multimedia (ISM 2005)*, 253–259.

Funakoshi, K., Watanabe, S., Kuriyama, N., & Tokunaga, T. (2004). Generating referring expressions using perceptual groups. *Proceedings of the 3rd International Conference on Natural Language Generation (INLG 2004)*.

Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1987). The vocabulary problem in human-system communication. *Communications of the ACM*, *30*(11), 964–971.

Gardent, C. (2002). Generating minimal definite descriptions. *Proceedings of the Fortieth Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, 96–103.

Gatt, A. (2006). Generating collective spatial references. *Proceedings of the 28th Annual Meeting of the Cognitive Science Society (CogSci 2006)*.

Gatt, A., & Belz, A. (2008). Attribute selection for referring expression generation: New algorithms and evaluation methods. *Proceedings of 5th International Natural Language Generation Conference (INLG 2008)*, 50-58.

Gatt, A., Belz, A., & Kow, E. (2009). The TUNA REG challenge 2009: Overview and evaluation results. *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, 174–182.

Gentner, D. (1982). Why nouns are learned before verbs: Linguistic relativity versus natural partitioning. *Center for the Study of Reading, Technical Report No. 257*.

Glasgow, J. I., & Papadias, D. (1992). Computational imagery. *AAAI Technical Report SS-92-02*.

Gorniak, P., & Roy, D. (2004). Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, *21*, 429–470.

Goudbeek, M., & Krahmer, E. (2012). Alignment in interactive reference production: Content planning, modifier ordering, and referential overspecification. *Topics in Cognitive Science*, 1–21.

Grabner, H., Gall, J., & Gool, L. V. (2011). What makes a chair a chair? *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*.

Grice, P. H. (1975). Logic and conversation. *Syntax and Semantics*, *3*, 41-58.

Griffin, Z. M., & Bock, K. (2000). What the eyes say about speaking. *Psychological Science*, *11*, 274–279.

Grosz, B. J., & Sidner, C. L. (1990). Plans for discourse. In P. R. Cohen, J. Morgan, & M. E. Pollack (Eds.), *Intentions in communication* (p. 417-444). Cambridge, MA: MIT Press.

Gundel, J. K., Hedberg, N., & Zacharski, R. (1993). Cognitive status and the form of referring expressions in discourse. *Language*, *69*(2), 274–307.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explorations*, *11*(1).

Haxby, J. V., Grady, C. L., Horwitz, B., Ungerleider, L. G., Mishkin, M., Carson, R. E., et al. (1991). Dissociation of object and spatial visual processing pathways in human extrastriate cortex. *Proceedings of the National Academy of Science of the USA*, *88*, 1621–1625.

Heeman, P. A., & Hirst, G. (1995). Collaborating on referring expressions. *Computational Linguistics*, *21*.

Hermann, T., & Deutsch, W. (1976). *Psychologie der objektbenennung.* Bern: Huber Verlag.

Hervás, R., & Finlayson, M. (2010). The prevalence of descriptive referring expressions in news and narrative. *Proceedings of the ACL 2010 Conference Short Papers*, 49–54.

Herzog, G., & Wazinski, P. (1994). Visual translator: Linking perceptions and natural language descriptions. *Artificial Intelligence Review*, *8*(2/3), 175–187.

Hindle, D. (1983). Deterministic parsing of syntactic non-fluencies. *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics (ACL 1983)*, 123–128.

Horton, W. S., & Keysar, B. (1996). When do speakers take into account common ground? *Cognition*, *59*(1), 91–117.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat?s visual cortex. *The Journal of Physiology*, *160*(1), 106–154.

Humphreys, G. W., Price, C. J., & Riddoch, M. J. (1999). From objects to names: A cognitive neuroscience approach. *Psychological Research*, *62*, 118–130.

Itti, L., & Arbib, M. A. (2005). Attention and the minimal subscene. In M. A. Arbib (Ed.), *Action to language via the mirror neuron system.* Cambridge: Cambridge University Press.

Itti, L., & Koch, C. (2001). Computational modelling of visual attention. *Nature Reviews Neuroscience*, *2*, 194–203.

Jordan, P. W. (2000). Influences on attribute selection in redescriptions: A corpus study. *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society (CogSci 2000)*, 250–255.

Jordan, P. W., & Walker, M. A. (2005). Learning content selection rules for generating descriptions in dialogue. *Journal of Artificial Intelligence Research*, *24*, 157–194.

Kelleher, J., & Costello, F. (2009). Applying computational models of spatial prepositions to visually situated dialog. *Computational Linguistics*, *35*(2), 271–306.

Kelleher, J., Costello, F., & van Genabith, J. (2005). Dynamically structuring, updating and interrelating representations of visual and linguistic discourse context. *Artificial Intelligence*, *167*, 62–102.

Kelleher, J., & Kruijff, G.-J. M. (2006). Incremental generation of spatial referring expressions in situation dialog. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL (COLING-ACL 2006)*, 1041–1048.

Keysar, B., & Henly, A. S. (2002). Speakers' overestimation of their effectiveness. *Psychological Science*, *13*(3), 207–212.

Koolen, R., Gatt, A., Goodbeek, M., & Krahmer, E. (2009). Need I say more? on factors causing referential overspecification. *Proceedings of the Workshop on the Production of Referring Expressions (PRE-CogSci 2009)*.

Koolen, R., Goudbeek, M., & Krahmer, E. (2011). Effects of scene variation on referential overspecification. *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society (CogSci 2011)*.

Koolen, R., Krahmer, E., & Theune, M. (2012). Learning preferences for referring expression generation: Effects of domain, language and algorithm. *Proceedings of the 7th International Workshop on Natural Language Generation (INLG 2012)*.

Kosslyn, S. M. (1980). *Image and mind*. Cambridge, MA: Harvard University Press.

Kosslyn, S. M. (1994). *Image and brain: The resolution of the imagery debate*. Cambridge, MA: MIT Press.

Krahmer, E., & Theune, M. (2002). Efficient context-sensitive generation of referring expressions. *Information Sharing: Reference and Presupposition in Language Generation and Interpretation*, *143*, 223–263.

Krahmer, E., & van der Sluis, I. (2003). A new model for generating multimodal referring expressions. *Proceedings of the 9th European Workshop on Natural Language Generation (ENLG 2003)*.

Krahmer, E., van Erk, S., & Verleg, A. (2003). Graph-based generation of referring expressions. *Computational Linguistics*, *29*(1), 53–72.

Krauss, R. M., & Glucksberg, S. (1969). The development of communication: Competence as a function of age. *Child Development*, *40*, 255–266.

Krauss, R. M., & Glucksberg, S. (1977). Social and nonsocial speech. *Scientific American*, *236*, 100–105.

Krauss, R. M., & Weinheimer, S. (1967). Effect of referent similarity and communication mode on verbal encoding. *Journal of Verbal Learning and Verbal Behavior*, *6*, 359–363.

Kronfeld, A. (1986). Donnellan's distinction and a computational model of reference. *Proceedings of the 24th Annual Meeting of Association for Computational Linguistics (ACL 1986)*, 186–191.

Kronfeld, A. (1987). Goals of referring acts. *Proceedings of the 1987 Workshop on Theoretical Issues in Natural Language Processing*, 164–170.

Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., et al. (2011). Baby talk: Understanding and generating image descriptions. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*.

Landau, B. (2001). Perceptual units and their mapping with language. In T. Shipley & P. Kellman (Eds.), *From fragments to objects: Segmentation and grouping in vision.* The Netherlands: Elsevier.

Landau, B., & Jackendoff, R. (1993). "what" and "where" in spatial language and spatial cognition. *Behavioral and Brain Sciences*, *16*, 217–265.

Landau, B., Smith, L., & Jones, S. (1998). Object perception and object naming in early development. *Trends in Cognitive Sciences*, *2*(1), 19–24.

Levelt, W. J. M. (1989). *Speaking: From intention to articulation.* Cambridge, MA: MIT Press.

Levelt, W. J. M., & Meyer, A. S. (2000). Word for word: Multiple lexical access in speech production. *European Journal of Cognitive Psychology*, *12*, 433–452.

Levelt, W. J. M., Roelofs, A. P. A., & Meyer, A. S. (1999). A theory of lexical access in speech production. *Behavioral and Brain Sciences*, *22*, 1–37.

Liu, C., Sharan, L., Adelson, E. H., & Rosenholtz, R. (2010). Exploring features in a bayesian framework for material recognition. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*.

Logothetis, N. K., Pauls, J., & Poggio, T. (1995). Shape representation in the inferior temporal cortex of monkeys. *Current Biology*, *5*, 552–563.

Logothetis, N. K., & Sheinberg, D. L. (1996). Visual object recognition. *Annual Review Neuroscience*, *19*, 577–621.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision (ICCV 1999)*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*, 91–110.

Mackworth, N. H., & Morandi, A. J. (1967). The gaze selects informative details within pictures. *Perception & Psychophysics*, *2*, 547–552.

MacNamara, J. (1972). Cognitive basis of language learning in infants. *Psychological Review*, *79*(1), 43–53.

Mangold, R., & Pobel, R. (1988). Informativeness and instrumentality in referential communication. *Journal of Language and Social Psychology*, *7*(3/4), 181–191.

Markman, E. (1989). *Categorization and naming in children: Problems of induction.* Cambridge, MA: MIT Press.

Mazzoni, D. (2010). *Audacity.* SourceForge.net. (SourceForge.net)

McRae, K. (2011). *McRae's norms.* Available from `http://amdrae.ssc.uwo.ca/McRaeLab/norms.php. Accessed 5.Oct.2011`

McRae, K., Cree, G. S., Seidenberg, M. S., & McNorgan, C. (2005). Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods, Instruments, & Computers*, *37*(4), 547–559.

Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, *85*(3), 207–238.

Mishkin, M., Underleider, L. G., & Macko, K. A. (1983). Object vision and spatial vision: Two cortical pathways. *Trends in Neurosciences*, *6*, 414–417.

Mitchell, M. (2008). *Towards the generation of natural reference.* Unpublished master's thesis, University of Washington.

Mitchell, M., Dunlop, A., & Roark, B. (2011). Semi-supervised modeling for prenominal modifier ordering. *Proceedings of the 49th Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011).*

Mitchell, M., van Deemter, K., & Reiter, E. (2011). On the use of size modifiers when referring to visible objects. *Proceedings of the 33rd Annual Conference of the Cognitive Science Society.*

Morzycki, M. (2009). Degree modification of gradable nouns: size adjectives and adnominal degree morphemes. *Natural Language Semantics*, *17*(2), 175–203.

Murata, A., Gallese, V., Luppino, G., Kaseda, M., & Sakata, H. (2000). Selectivity for the shape, size, and orientation of objects for grasping in neurons of monkey parietal area AIP. *Journal of neurophysiology*, *83*(5), 2580–2601.

Murray, S. O., Boyaci, H., & Kersten, D. (2006). The representation of perceived angular size in human primary visual cortex. *Nature Neuroscience*, *9*(3), 429–434.

Naor-Raz, G., Tarr, M. J., & Kersten, D. (2003). Is color an intrinsic property of object representation? *Perception*, *32*, 667–680.

Nelson, K. (1973). Structure and strategy in learning to talk. *Monographs of the Society for Research in Child Development*, *38*(1/2), 1–135.

Oliva, A. (2005). Gist of the scene. In L. Itti, G. Rees, & J. K. Tsotsos (Eds.), *The encyclopedia of neurobiology of attention* (pp. 251–256). Amsterdam: Elsevier, Inc.

Oliva, A., & Torralba, A. (2007). The role of context in object recognition. *Trends in Cognitive Sciences*, *11*(12), 520–527.

Olson, D. R. (1970). Language and thought: Aspects of a cognitive theory of semantics. *Psychological Review*, *77*, 257–273.

Palmeri, T. J., & Gauthier, I. (2004). Visual object understanding. *Nature Reviews Neuroscience*, *5*, 291–303.

Pechmann, T. (1989). Incremental speech production and referential overspecification. *Linguistics*, *27*, 89–110.

Poesio, M. (2000). Annotating a corpus to develop and evaluate discourse entity realization algorithms: Issues and preliminary results. *Proceedings of the 2nd Language Resources and Evaluation Conference (LREC 2000)*, 211–218.

Posner, M. I., & Cohen, Y. A. (1984). Components of visual orienting. In H. Bouma & D. G. Bouwhuis (Eds.), *Attention and performance X: Control of language processes.* Hillsdale, NJ: Lawrence Erlbaum.

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., & Belongie, S. (2007). Objects in context. *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2007)*.

Reiter, E., & Belz, A. (2009). An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, *35*(4), 529–558.

Reiter, E., & Dale, R. (1992). A fast algorithm for the generation of referring expressions. *Proceedings of the 14th International Conference on Computational Linguistics (COLING 1992)*, *1*, 232–238.

Reiter, E., & Dale, R. (2000). *Building natural language generation systems*. Cambridge: Cambridge University Press.

Reiter, E., & Sripada, S. (2002). Human variation and lexical choice. *Computational Linguistics*, *28*, 545–553.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, *2*(11).

Riesenhuber, M., & Poggio, T. (2000). Models of object recognition. *Nature Neuroscience Supplement*, *3*, 1199–1204.

Roelofs, A. (1998). Rightward incrementality in encoding simple phrasal forms in speech production: Verb–particle combinations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *24*(4), 904–921.

Rogez, G., Rihan, J., Ramalingam, S., Orrite, C., & Torr, P. H. (2008). Randomized trees for human pose detection. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*.

Rosch, E. (1975). Cognitive representation of semantic categories. *Journal of Experimental Psychology*, *104*, 192–233.

Rosch, E., C. Mervis, C., W. Gray, W., Johnson, D., & Braem, P. B. (1976). Basic objects in natural categories. *Cognitive Psychology*, *8*, 382–439.

Rosch, E., & Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, *7*, 573–605.

Roy, D., & Reiter, E. (2005). Connecting language to the world. *Artificial Intelligence*, *167*, 1–12.

Roy, D. K. (2002). Learning visually-grounded words and syntax for a scene description task. *Computer Speech and Language*, *16*, 353–385.

Rubin, A. D. (1980). A theoretical taxonomy of the differences between oral and written language. In R. J. Spiro, B. C. Bruce, & W. F. Brewer (Eds.), *Theoretical issues in reading comprehension.* Hillsdale, New Jersey: Lawrence Erlbaum Assocs.

Sacks, H., & Schegloff, E. A. (1979). Two preferences in the organization of reference to persons in conversation and their interaction. In G. Psathas (Ed.), *Everyday language: Studies in ethnomethodology* (pp. 15–21). New York: Irvington Publishers.

Savarese, S., & Fei-Fei, L. (2007). 3D generic object categorization, localization and pose estimation. *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2007).*

Schober, M. F., & Clark, H. H. (1989). Understanding by addressees and overhearers. *Cognitive Psychology*, *21*, 211–232.

Schriefers, H. (1992). Lexical access in the production of noun phrases. *Cognition*, *45*, 33–54.

Schriefers, H. (1993). Syntactic processes in the production of noun phrases. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *19*(4), 841–850.

Schwarzkopf, D. S., Song, C., & Rees, G. (2010). The surface area of human V1 predicts the subjective experience of object size. *Nature Neuroscience*, *14*(1), 28–30.

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language.* Cambridge: Cambridge University Press.

Sedivy, J. C. (2003). Pragmatic versus form-based accounts of referential contrast: Evidence for effects of informativity expectations. *Journal of Psycholinguistic Research*, *32*, 3–23.

Serre, T., Wolf, L., & Poggio, T. (2005). Object recognition with features inspired by visual cortex. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 994–1000.

Siddharthan, A., & McKeown, K. (2005). Improving multilingual summarization: Using redundancy in the input to correct mt errors. *Proceedings of the Human Language*

Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP).

Siddharthan, A., Nenkova, A., & McKeown, K. (2011). Information status distinctions and referring expressions: An empirical study of references to people in news summaries. *Computational Linguistics*, *37*(4), 811–842.

Sonnenschein, S. (1985). The development of referential communication skills: Some situations in which speakers give redundant messages. *Journal of Psycholinguistic Research*, *14*, 489–508.

Sripada, S., Reiter, E., & Davy, I. (2003). Sumtime-mousam: Configurable marine weather forecast generator. *Expert Update*, *6*, 4–10.

Stone, M. (2000). On identifying sets. *Proceedings of the 1st International Conference on Natural Language Generation (INLG 2000)*, 116–123.

Strawson, P. F. (1950). On referring. *Mind*, *59*(235), 320–344.

Su, H., Sun, M., Fei-Fei, L., & Savarese, S. (2009). Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. *Proceedings of the International Conference in Computer Vision (ICCV 2009)*.

Tanaka, J. W., & Presnell, L. M. (1999). Color diagnosticity in object recognition. *Perception and Psychophysics*, *61*(6), 1140–1153.

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

Tarr, M. J., & Gauthier, I. (2000). FFA: a flexible fusiform area for subordinate-level visual processing automatized by expertise. *Nature Neuroscience*, *3*(8), 764–769.

Therriault, D. J., Yaxley, R. H., & Zwaan, R. A. (2009). The role of color diagnosticity in object recognition and representation. *Cogn. Process.*, *10*(4), 335–42.

Theune, M., Koolen, R., Krahmer, E., & Wubben, S. (2011). Does size matter – how much data is required to train a REG algorithm? *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*.

Treisman, A. M., & Gelade, G. (1980). A feature integration theory of attention. *Cognitive Psychology*, *12*, 97–13.

Treue, S., & Trujillo, J. C. M. (1999). Feature-based attention influences motion processing gain in macaque visual cortex. *Nature*, *399*, 575–579.

Tucker, G. (1998). *The lexicogrammar of adjectives: A systemic functional approach to lexis*. London: Cassell.

Underwood, G., Templeman, E., Lamming, L., & Foulsham, T. (2008). Is attention necessary for object identification? Evidence from eye movements during the inspection of real-world scenes. *Consciousness and Cognition*, *17*, 159–170.

van Deemter, K. (2000). Generating vague descriptions. *Proceedings of the 1st Natural Language Generation Conference*, 12–16.

van Deemter, K. (2002). Generating referring expressions: Boolean extensions of the incremental algorithm. *Computational Linguistics*, *28*(1), 37–52.

van Deemter, K. (2006). Generating referring expressions that involve gradable properties. *Computational Linguistics*, *32*(2), 195–222.

van Deemter, K., Gatt, A., van der Sluis, I., & Power, R. (2012). Generation of referring expressions: Assessing the incremental algorithm. *Cognitive Science*, *36*(5), 799–836.

van Deemter, K., van der Sluis, I., & Gatt, A. (2006). Building a semantically transparent corpus for the generation of referring expressions. *Proceedings of the 4th International Conference on Natural Language Generation (INLG 2006)*.

van der Sluis, I., & Krahmer, E. (2005). Towards the generation of overspecified multimodal referring expressions. *Proceedings of the Symposium on Dialogue Modelling and Generation of the 15th Annual Meeting of the Society for Text and Discourse*.

van Gompel, R. P. G., Gatt, A., Krahmer, E., & Deemter, K. v. (2012). PRO: A computational model of referential overspecificiation. *Architectures and Mechanisms for Language Processing (AMLaP 2012)*.

Viethen, J., & Dale, R. (2008). The use of spatial relations in referring expression generation. *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, 59–67.

Viethen, J., & Dale, R. (2009). Referring expression generation: What can we learn from human data? *Proceedings of the Workshop on the Production of Referring Expressions (PRE-CogSci 2009)*.

Viethen, J., & Dale, R. (2010). Speaker-dependent variation in content selection for referring expression generation. *Proceedings of the 8th Australasian Language Technology Workshop (ALTW 2010)*, 81–89.

Viethen, J., Dale, R., Krahmer, E., Theune, M., & Touset, P. (2008). Controlling redundancy in referring expressions. *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*.

Viethen, J., Goudbeek, M., & Krahmer, E. (2012). The impact of colour difference and colour codability on reference production. *Proceedings of the 34th Annual Meeting of the Cognitive Science Society (CogSci 2012)*.

Walther, D., Itti, L., Riesenhuber, M., Poggio, T., & Koch, C. (2002). Attentional selection for object recognition - a gentle way. *Proceedings of the 2nd Workshop on Biologically Motivated Computer Vision (BMCV '02)*, 472–479.

Whitehurst, G. J. (1976). The development of communication: Changes with age and modeling. *Child Development*, *47*, 473–482.

Whorf, B. L. (1956). *Language, thought, and reality.* New York: MIT Press.

Wolfe, J. M., & Myers, L. (2010). Fur in the midst of the waters: Visual search for material type is inefficient. *Journal of Vision*, *10(9)*(8), 1–9.

Wu, L., & Barsalou, L. (2009). Perceptual simulation in conceptual combination: Evidence from property generation. *Acta Psychologica*, *132*, 173–189.

Wurm, L. H., Legge, G. E., Isenberg, L. M., & Luebker, A. (1993). Color improves object recognition in normal and low vision. *Journal of Experimental Psychology: Human Perception and Performance*, *19*(4), 899–911.

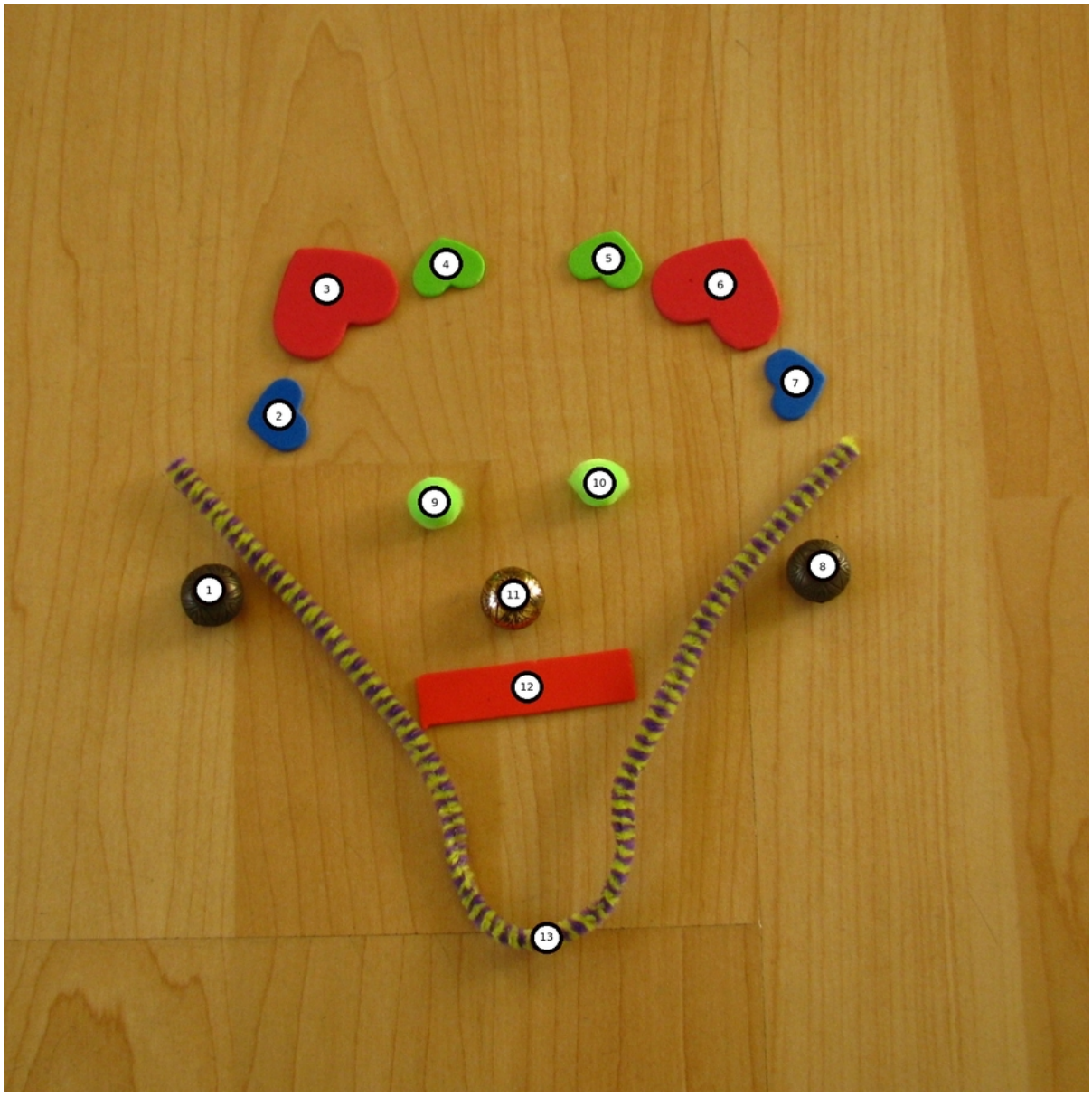
Yip, A. W., & Sinha, P. (2002). Contribution of color to face recognition. *Perception*, *31*, 995–1003.

Zheng, S., Yuille, A., & Tu, Z. (2010). Detecting object boundaries using low-, middle-, and high-level information. *Journal of Computer Vision and Image Understanding*, *114*(19), 1055–1067.

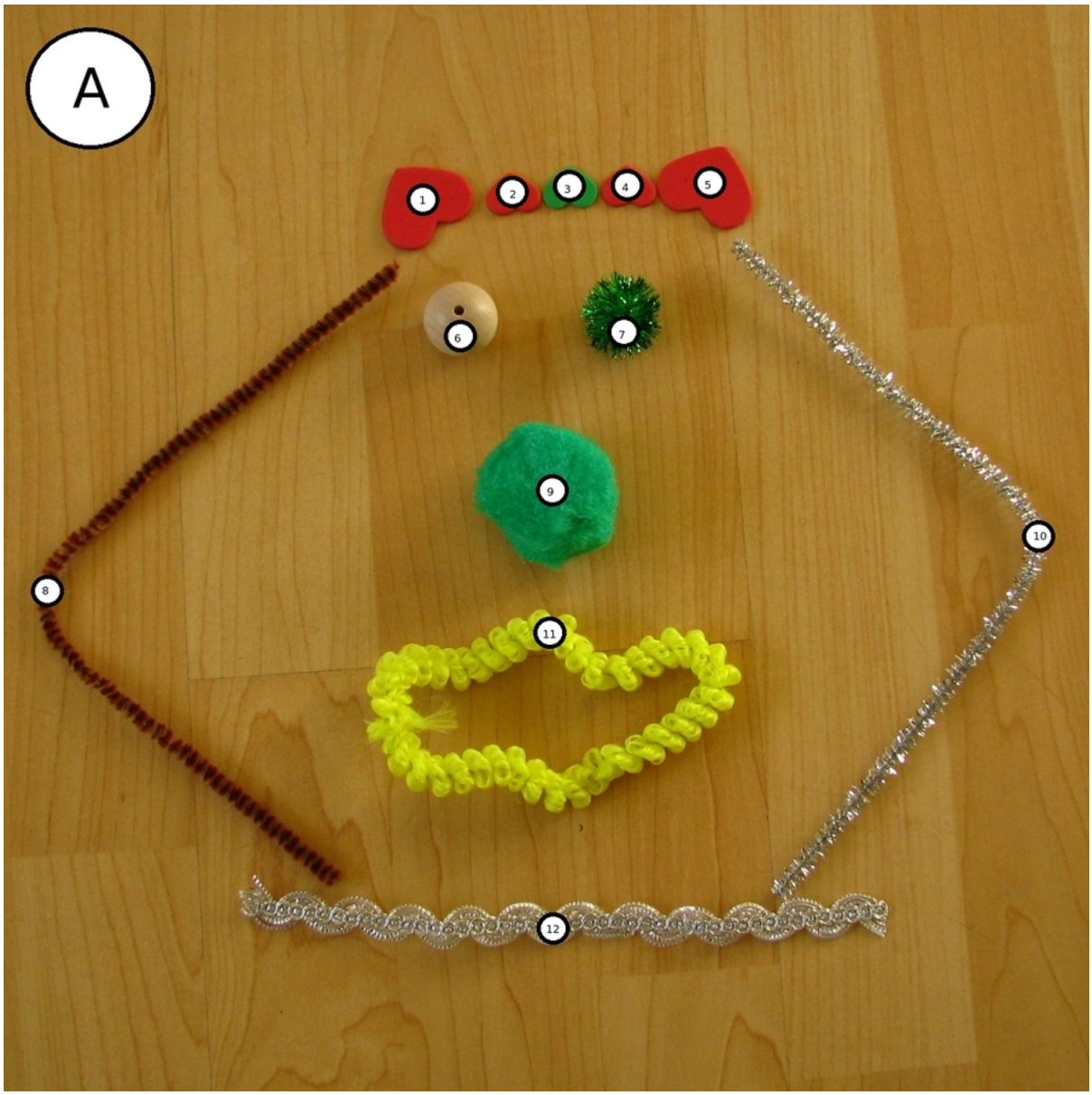
Zipf, G. K. (1935). *The psychobiology of language.* Boston, MA: Houghton-Mifflin.

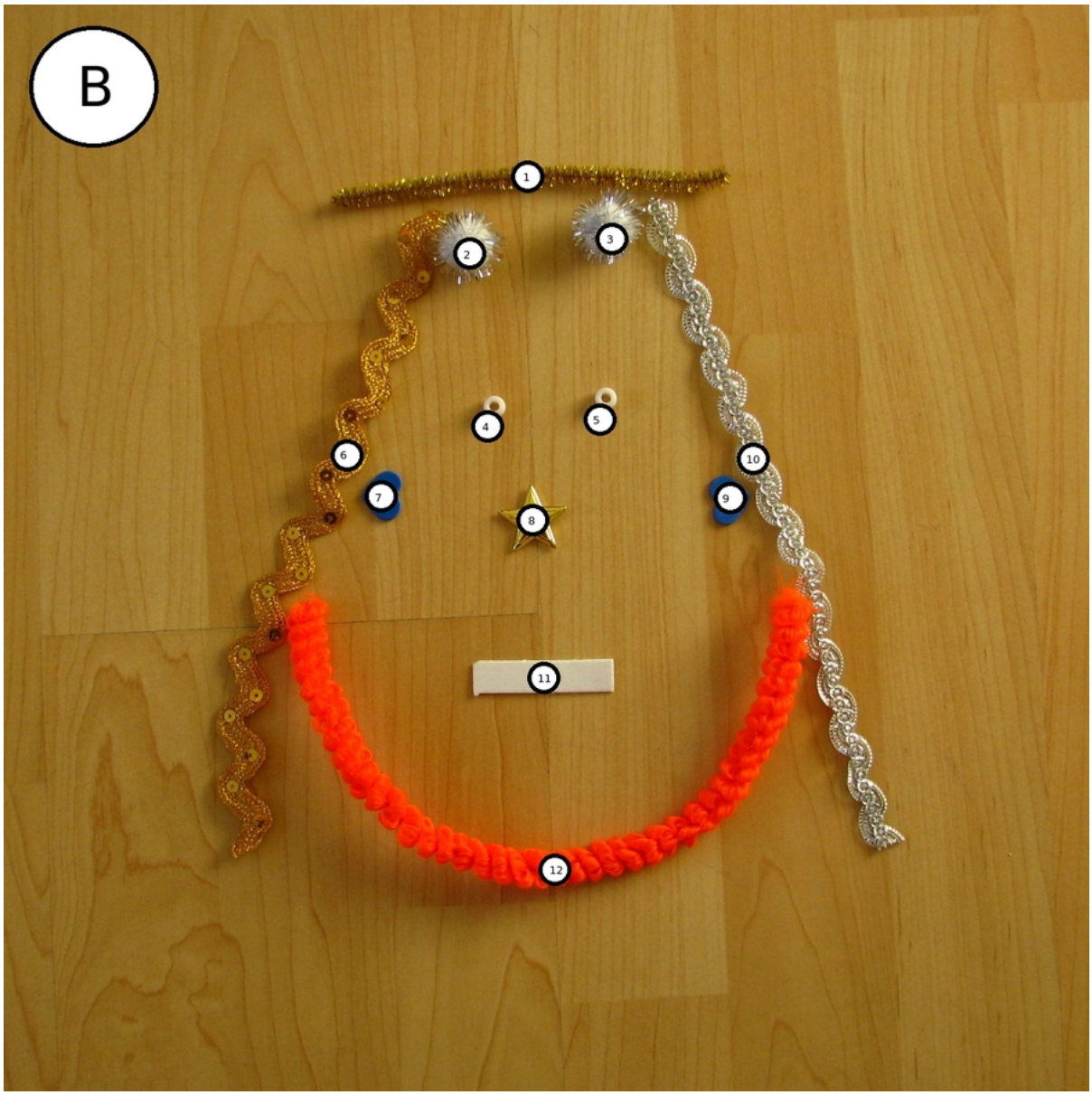
Appendix A

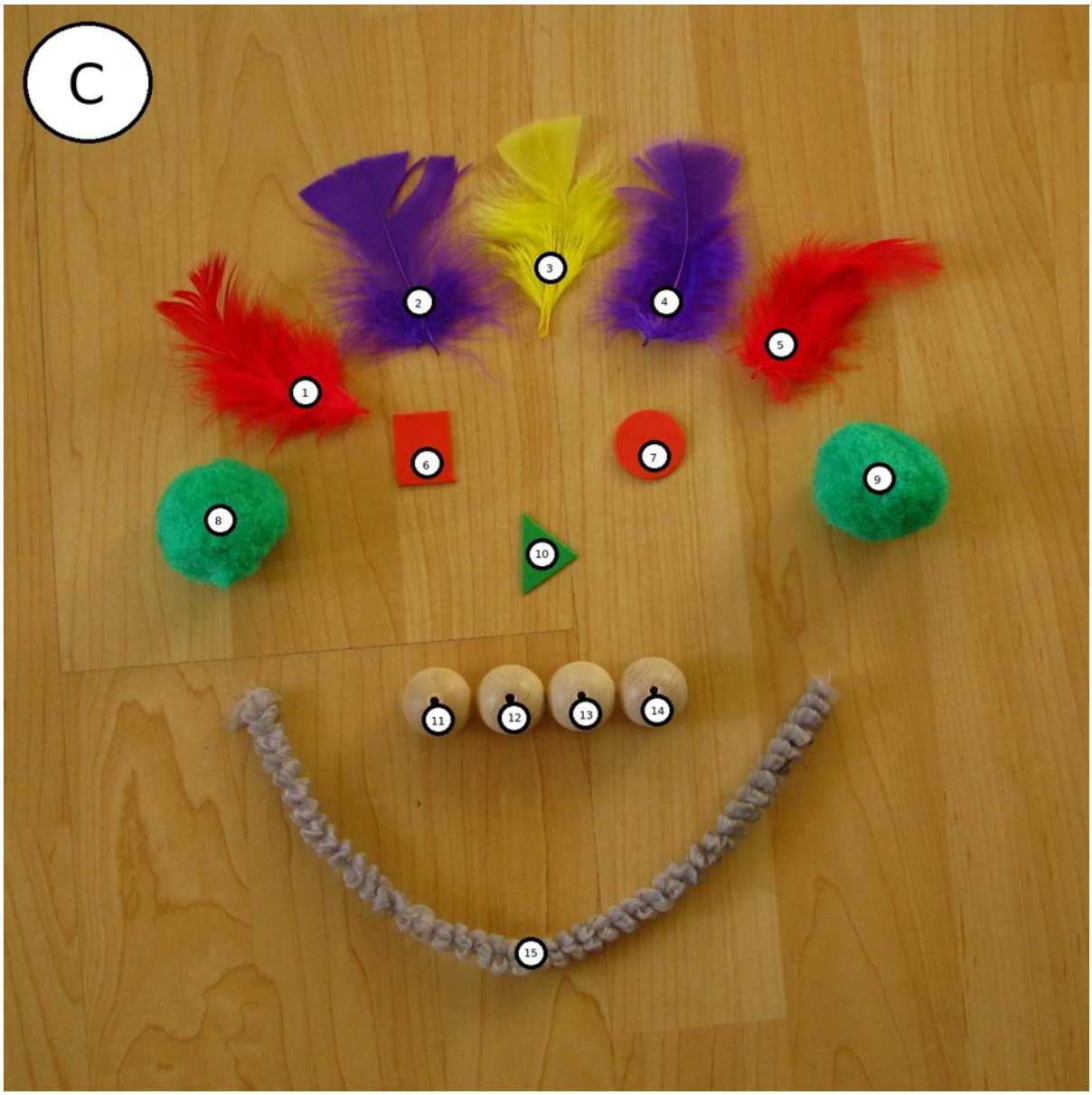
Craft Study - Annotated Faces

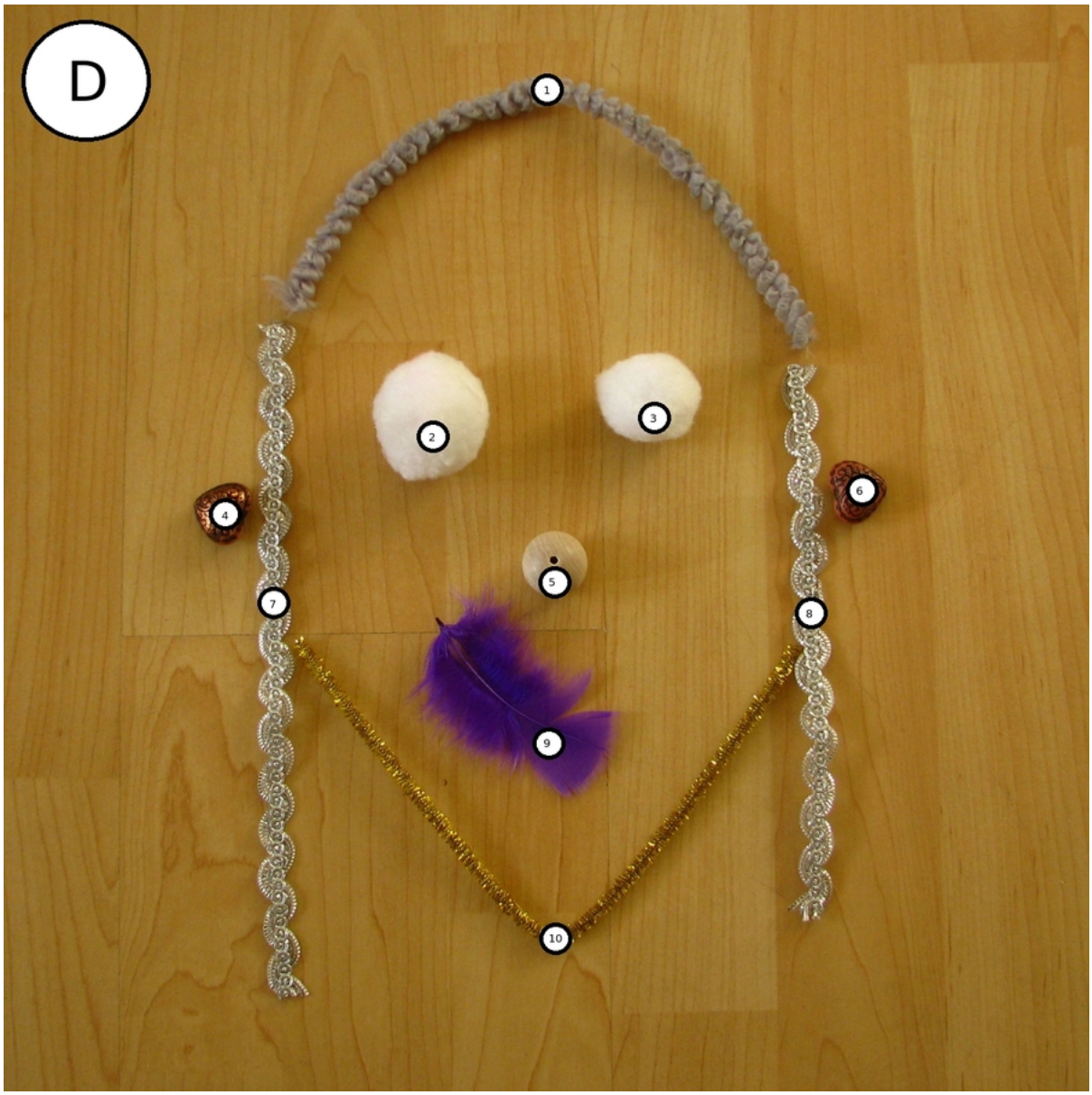


trial face (t)


A
1
2
3
4
5
6
7
8
9
10
11
12


B
1
2
3
4
5
6
7
8
9
10
11
12


C
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15


D
1
2
3
4
5
6
7
8
9
10

Appendix B

Craft Study - Instructions for Participants

Face Construction Task - Ethics Details

For this experiment, your voice will be recorded. You are under no obligation to complete this task, and may leave at any time. Your participation is voluntary, and you may withdraw from the research at any point without penalty and for any reason. Your data will be treated with full confidentiality and if the results are published, you will not be identified. All recordings will not be identifiable. After the experiment, you will be debriefed with an explanation of the study.

Thanks!

Face Construction Task - Directions

Thank you for participating in our study! You will be giving instructions for a craft face construction task. This should take about half an hour. We're looking for people to clearly explain how to put together the face in each picture, using the craft items on the table. The instructions you give should be clear enough for someone without the pictures to be able to put each face together using the same craft objects. If you have any questions, please let us know before you begin. Once you start describing, we cannot give you any feedback!

Directions

- In front of you, you will see a stack of photographs and a bunch of craft supplies.
- Look at the first photograph, and explain how to construct it utilizing the craft supplies provided.
- Please speak clearly enough for someone without the photograph to be able to reconstruct the face based on your directions.
- Don't worry about reshaping some of the supplies for the faces; the shaped pieces on the table are close enough to those in the original pictures.
- When you are done with the first photograph, you may move on to the second. You may introduce this in the recording by saying "Next face".
- Assume that a listener may not construct the faces in the same order as they appear in your stack, so try to give directions specific only to the current face.
- Repeat this for all five photographs.
- When you are done with all five photographs, you may just leave everything as is. We will have a feedback sheet for you to let us know what you thought about the task.

Thanks again!

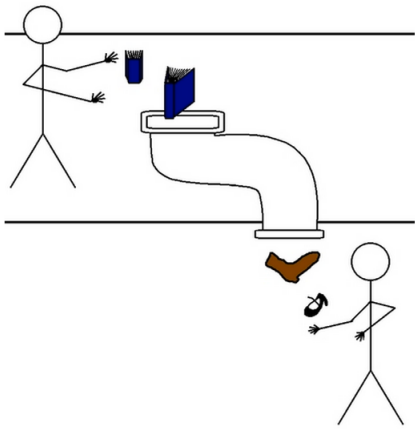
Appendix C

Size Study - Instructions for Participants

Below is an example section of an Amazon Mechanical Turk HIT shown to participants, with instructions included.

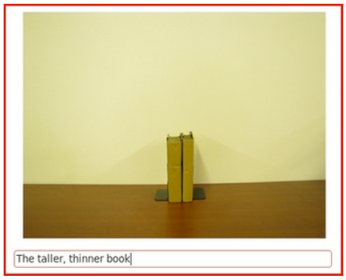
Name the object!

In this game, objects are falling through a chute to a partner downstairs.
You have been chosen as the Thrower, tossing two or three objects at a time down the chute.
For each toss, you want your partner to pick up just one of the objects.
The goal of the game is to *tell your partner which object to pick up*.



Below, you will see 40 different pictures of objects.
Each picture has 2 or 3 objects.
The object for your partner to pick up will always appear to you as the object on the right.
However, after falling through the chute, the objects might not be in the same positions!
-- **Your task** is to identify the object so that your partner can figure out which one it is.

Example:



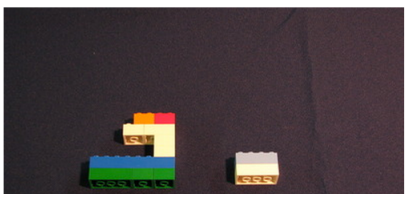
The objects are:

- Spatulas
- Brownies
- Shoes
- Sponges
- Boards
- Books
- Legos

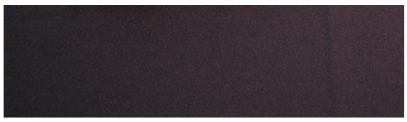
Clearly identify the object on the right so your partner can pick it up.

To be approved, please identify the rightmost object in all 40 pictures.

Click on the image to make it larger; click it again to make it smaller.



Below is an example section of an Amazon Mechanical Turk HIT shown to participants, with instructions included. (Continued.)



1/40

2/40

Remember: Clearly identify the object on the right for each picture.
The objects are spatulas, brownies, shoes, sponges, boards, books, and legos.

3/40

(And continues down as subjects scroll.)

Appendix D

Typicality Study - Instructions for Participants

In this study, you will be instructing the person across from you, _____, which objects to place in front of him/her to recreate the pictures you see on the screen. I will be recording then transcribing this, then deleting the recording.

There will be 7 pictures, and you can flip through them as quickly or as slowly as you'd like.

Before the study, you're free to touch the objects. Once the study begins, you can't touch or point to the objects – you can only describe them.

When instructing _____, the exact placement of the objects doesn't matter, but try and explain how to line them up roughly on each column of dots. [Gesture to each row, numbering them 1 – 5].

After you describe which object to move, _____ will lift his/her hand to begin moving it. When his/her hand is lifted over the set of objects, you can't speak until the object is in the area of the dots. What I'm trying to do here is just avoid the situation where you just say "left, left, left..." [motion hand above objects]. Just say which object you mean, and let _____ grab it. _____, if it's unclear which object is meant, just make your best guess. You can correct an object once the incorrect object has been placed.

Feel free to examine the objects now; you won't be able to touch them once the study begins.

We'll begin with a practice picture, which should be the first picture in your set, and then you can ask me any questions you have.

Appendix E

Publications from this Thesis

Mitchell, M., van Deemter, K., and Reiter, E. (2013). Generating Expressions that Refer to Visible Objects. *Proceedings of NAACL 2013*.

Mitchell, M., Reiter, E., and van Deemter, K. (2013). Typicality and Object Reference. *Proceedings of CogSci 2013*.

Mitchell, M., Han, X., and Hayes, J. (2012). Midge: Generating Descriptions of Images. *Proceedings of INLG 2012*. System demonstration.

Mitchell, M., Dodge, J., Goyal, A., Yamaguchi, K., Stratos, K., Han, X., Mensch, A., Berg, A., and Berg, T. L., Daumé III, H. (2012). Midge: Generating Image Descriptions From Computer Vision Detections. *Proceedings of EACL 2012*.

Mitchell, M., van Deemter, K., and Reiter, E. (2011). Two Approaches for Generating Size Modifiers. *Proceedings of ENLG 2011*.

Mitchell, M. (2011). From an Image to a Description. *Vision and Language Workshop 2011*.

Mitchell, M., van Deemter, K., and Reiter, E. (2011). Applying Machine Learning to the Choice of Size Modifiers. *Proceedings of PRE-CogSci 2011*.

Mitchell, M., van Deemter, K., and Reiter, E. (2011). On the Use of Size Modifiers When Referring to Visible Objects. *Proceedings of CogSci 2011*.

Mitchell, M., van Deemter, K., and Reiter, E. (2010). Natural Reference to Objects in a Visual Domain. *Proceedings of INLG 2010*.