LEARNING IN THE RATIONAL SPEECH ACTS MODEL

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Will Monroe
June 2018

Abstract

When a person says something that has multiple possible interpretations, which interpretation stands out as the most likely intended meaning often depends on context outside the utterance itself: salient objects in the environment, utterances the speaker could have chosen but didn't, common-sense knowledge, etc. Systematically predicting these contextual effects is a major unsolved problem in computational natural language understanding.

A recently-developed framework, known in cognitive science as the *rational speech acts* (RSA) model, proposes that speaker and listener reason probabilistically about each other's goals and private knowledge to produce interpretations that differ from literal meanings. The framework has shown promising experimental results in predicting a wide variety of previously hard-to-model contextual effects. This dissertation describes a variety of methods combining RSA approaches to context modeling with machine learning methods of language understanding and production. Learning meanings of utterances from examples avoids the need to build an impractically large, brittle lexicon, and having models of both speaker and listener also provides a way to reduce the search space by sampling likely subsets of possible utterances and meanings.

Using recently-collected corpora of human utterances in simple language games, I show that a combination of RSA and machine learning yields more human-like models of utterances and interpretations than straightforward machine learning classifiers. Furthermore, the RSA insight relating the listener and speaker roles enables the use of a generation model to improve understanding, as well as suggesting a new way to evaluate natural language generation systems in terms of an understanding task.

To Granddad
and Grandpop

Acknowledgments

If there is one thing I have learned from graduate school (and I would like to think I have learned at least one thing!), it is that big things can rarely be done alone. I cannot hope to give a full account of everyone who has helped me personally and academically in this journey, but my debt to them demands that I try.

Firstly, my advisor, Chris Potts, deserves my immense gratitude for his years of patient, encouraging mentorship, and especially for his willingness to take a chance on a student from another department with only a course project to judge my potential. His enthusiasm and support helped me get through what was without a doubt the most stressful period of my life, the first year or two of the Ph.D. program, and regain confidence in myself. I am also grateful to Noah Goodman, Dan Jurafsky, and Percy Liang both for their skilled research advising and for all the annoying extra work that I have often had to request of them, as well as to Roger Romani for agreeing to be the chair of my dissertation defense committee.

The members of the Stanford Natural Language Processing group have been my role models, collaborators, friends, and support group throughout my time as a graduate student. Research would have been far less meaningful without the friendships that I gained as a member of the NLP group. I would especially like to thank Jiwei Li, Danqi Chen, Angel Chang, Sida Wang, Gabor Angeli, and Spence Green for providing serious research help and enjoyable distractions, listening to my complaints, and generally keeping me sane for the last few years.

The faculty who worked with me in the first year as part of the Computer Science Department's rotation system gave me direction and helped me cross the chasm between undergraduate coursework and publishing research; of these, the ones I have not

previously mentioned by name are Pat Hanrahan and Chris Manning. I must express both gratitude and apologies to Pat and the others in the graphics lab—gratitude for believing in me and giving me the opportunity to stay at Stanford when I applied to graduate school wanting to work on rendering and physical simulation, and apologies for abandoning them to ride the artificial intelligence wave (I hope, at least, that my attention to color in this work is some consolation).

I have also been lucky to be able to collaborate on research with many other inspiring and hard-working individuals: Manolis Savva, Alan Ritter, Michel Galley, Jianfeng Gao, Amir Goldberg, Sameer Srivastava, Govind Manian, Robert Hawkins, Tianlin Shi, Sébastien Jean, Jennifer Hu, Andrew Jong, Arianna Yuan, Yu Bai, and Nate Kushman. Several of these collaborations owe an additional thank-you to the unnamed heroes of computational natural language research: the workers of Amazon Mechanical Turk, who by now are probably thoroughly bored with describing colors.

The teachers and mentors from my undergraduate career and before who guided me to where I am today are far too many to name, but among those who helped me on the path to graduate school, whether by mentoring me in undergraduate research, writing letters of recommendation for me, or simply teaching inspiring classes: Eric Roberts, Mehran Sahami, Jerry Cain, Julie Zelenski, Chris Piech, Phil Levis, Virginia Williams, Ryan Williams, Mike Meinert, and Chris McCart.

I would like to thank my amazing girlfriend Eileen for being there for me year after year, phone call after phone call, despite the thousands of miles of distance between us, putting up with complaints about my research code even at the end of long days at her own software engineering job. 辛苦你了。

Finally, I owe the world to my family—my brother, my grandparents, my aunts and uncles and cousins, but especially my parents—for making it possible for me to get this far. Every tall building needs a good foundation, and beneath all the figures and equations and jargon that are about to follow are all the things my parents taught me when I was growing up, and all the things they continued to teach me as I pushed onward through graduate school: how not to give up, how to recover from failure, and the importance of having fun through it all.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

An intriguing property of natural language is the symmetry between production and understanding: when a person who is speaking wants to communicate a concept, the word the speaker produces is typically a word the same speaker would readily understand as referring to the same concept. In other words, we speak the language we hear (as a general rule; there are of course exceptions).

Such symmetry is a natural consequence of people's participation in conversation as speakers and listeners, and it has other cognitive benefits, such as the ability to leverage mechanisms for producing language to improve the ability to understand it, and vice versa. For example, a listener can understand speech masked by distortions if it matches the listener's prediction of the speaker's intent (Warren, 1970).

This dissertation is about building better systems for computational natural language understanding and generation by taking advantage of the speaker–listener symmetry. The proposed systems use a framework for mathematically modeling the relationship between speaker and listener, which, under the name *rational speech acts* (RSA), has achieved remarkable experimental success recently in predicting a variety of difficult-to-model cases of non-literal language understanding. The general form of this framework assumes a Bayesian listener and a speaker that acts to maximize a utility function related to the listener's understanding.

1.1 Overview

In the remaining sections of this chapter, I provide an overview of the RSA model described above, some of its drawbacks, and the proposals for overcoming them presented in this dissertation.

Chapter 2 shows one method for combining RSA with the ability to learn from examples. This method uses the gradient of probabilities computed by the RSA model to optimize the parameters of an underlying machine learning model. *This chapter is based on a paper published in the 20th Amsterdam Colloquium as Monroe and Potts (2015). It represents joint work with Christopher Potts, who implemented the RSA baselines, contributed to the writing, and advised the project.*

Chapters 3 through 5 demonstrate a related approach to combining RSA with machine learning on tasks related to describing colors. Chapter 3 introduces recurrent neural network (RNN) sequence models, a general method for converting between variable-length sequences and general-purpose vector representations, which can be related to various forms of meaning and external context. It then presents a system for describing colors using an RNN, which will be the basis of the speaker model for RSA approaches in the later chapters. *This chapter is based on a paper published in the Conference on Empirical Methods in Natural Language Processing as Monroe et al. (2016). It represents joint work with Christopher Potts and Noah D. Goodman, who contributed to the writing and advised the project.*

Chapter 4 presents a model for understanding color descriptions in context that takes advantage of this RNN speaker model in two ways: by taking samples from the space of possible utterances it must consider, and by combining (ensembling) RSA listeners with a pure RNN listener, allowing the model to learn from producing utterances in context to improve its ability to understand them. *This chapter is based on an article published in the Transactions of the Association for Computational Linguistics as Monroe et al. (2017). It represents joint work with Robert X.D. Hawkins, who performed the collection and analysis of the corpus data and wrote the corresponding sections; and Christopher Potts and Noah D. Goodman, who contributed to the writing and advised the project.*

Chapter 5 examines the effects of training a single speaker model to produce color descriptions in more than one language. Development of this speaker model includes the use of an RSA listener to evaluate the effectiveness of the speaker at producing utterances that are distinctive in context. *This chapter is based on an article published in the North American meeting of the Association for Computational Linguistics as Monroe et al. (2018). It represents joint work with Jennifer Hu and Andrew Jong, who performed the collection and analysis of the corpus data and wrote the corresponding sections; and Christopher Potts and Noah D. Goodman, who contributed to the writing and advised the project.*

Finally, Chapter 6 concludes with a brief summary and a discussion of desirable future research in this area.

1.2 The rational speech acts model of pragmatics

The main mathematical model of the relationship between speaker and listener used in this work, as applied to pragmatic (context-sensitive) language understanding, is known as *rational speech acts* (RSA) in cognitive science (Frank and Goodman, 2012; Goodman and Frank, 2016). RSA models language use as a recursive process in which speakers and listeners reason about each other to enrich the literal semantics of their language. This increases the efficiency and reliability of their communication compared to what more purely literal agents can achieve.

To give a concrete context for the formalization of this model and demonstrate its predictions, it is helpful to consider a particular simplified setting for communication, the *reference game* (Rosenberg and Cohen, 1964; Krauss and Weinheimer, 1964; Paetzel et al., 2014). Reference games embed language use in a goal-oriented communicative context (Clark, 1996; Tanenhaus and Brown-Schmidt, 2008). Since they offer the simplest experimental setup where many pragmatic and discourse-level phenomena emerge, these games have been used widely in cognitive science to study topics like common ground and conventionalization (Clark and Wilkes-Gibbs, 1986), referential domains (Brown-Schmidt and Tanenhaus, 2008), perspective-taking (Hanna et al., 2003), and overinformativeness (Koolen et al., 2011).

c_1 c_2 c_3

(a) Simple reference game.

	beard	*glasses*	*tie*
c_1	.5	.5	0
c_2	0	.5	.5
c_3	0	0	1

(b) S_0

	c_1	c_2	c_3
beard	1	0	0
glasses	.5	.5	0
tie	0	.33	.67

(c) L_1

	beard	*glasses*	*tie*
c_1	.67	.33	0
c_2	0	.6	.4
c_3	0	0	1

(d) S_2

Figure 1.1: Ambiguity avoidance in RSA.

Figure 1.1a shows an example of a context for a reference game between a speaker S and a listener L. The speaker is privately assigned referent c_1 and must send a message (here, limited to *beard*, *glasses*, and *tie*) that conveys this to the listener. A literal speaker chooses randomly between *beard* and *glasses*. However, if S imagines itself in the situation of L receiving these messages, then S will see that *glasses* creates uncertainty about the referent whereas *beard* does not, so S will favor *beard*. In short, the pragmatic speaker chooses *beard* because it's unambiguous for the listener.

RSA formalizes this reasoning in probabilistic Bayesian terms. The RSA model, as described by Goodman and Frank (2016), consists of two basic assumptions. First, speakers act to approximately maximize some utility function that depends on a listener's understanding. A standard choice for this utility function combines the log probability of the listener correctly understanding the message (identifying the target) with a notion of the cost of producing a message, represented by a function K mapping messages to real numbers. Formally, if the speaker S is modeled as a probability distribution over utterances u given the set of possible referents (context) C and the identity of the target t, and the listener is modeled as a distribution over targets given the context and an utterance, this gives the following equation for deriving a speaker from a listener (using notation from Bergen et al., 2016):

$$S_n(u \mid t, C) \propto \exp\left(\lambda \left(\log L_{n-1}(t \mid u, C) - K(u)\right)\right) \tag{1.1}$$

The cost term $K(u)$ discourages the speaker from producing impractically long and detailed messages (which in theory could make the listener choose the correct

target with arbitrary certainty). The addition of a cost term also ensures the speaker distribution is well-defined over infinite sets of messages. Most of the work in this dissertation uses finite sets of utterances and ignores the cost term, assuming "talk is cheap"; however, I include the cost term in this introduction of RSA to anticipate the theoretical issues that arise from omitting it in a general model of language use.

The second assumption of the RSA model is that the listener performs optimal Bayesian reasoning using the speaker's utterance as evidence. This can be expressed mathematically as follows:

$$L_n(t \mid u, C) \propto S_{n-1}(u \mid t, C)P(t) \tag{1.2}$$

A notable property of these speakers and listeners is that they describe a distribution over messages (1.1) and targets (1.2), in which the agent is merely more likely to choose higher-utility utterances or higher-probability targets according to a *softmax* distribution. As a self-contained model of a speaker or listener, this may seem less rational than consistently choosing the maximum; however, when using one model to define the next, leaving the distribution as a softmax has an important smoothing effect and allows the higher-level agent to consider the possibility that the other agent will behave suboptimally some of the time. The temperature parameter λ partially governs this smoothing effect in the speaker, with higher values leading to a speaker model that more consistently chooses the maximum-utility utterance. The λ parameter also indirectly affects the listener's interpretations: the more reliably the speaker chooses the optimal utterance for a referent, the more the listener will take deviations from the optimum as a signal to choose a different referent.

1.2.1 Speaker-based models

The above two equations recursively define a hierarchy of speakers and listeners. To instantiate this hierarchy, one needs a base case for the recursion. In RSA, this base case is a *literal agent*, either a speaker or a listener. A literal agent replaces the model of a lower-level agent with a definition of the meaning of messages as defined by a lexicon $\mathcal{L}$, where $\mathcal{L}(u, t, C) = 1$ if u is true of t in the context of C (and 0 otherwise).

That is, a *literal speaker* chooses a message based solely on its semantic compatibility with the target and possibly the cost of the message, and a *literal listener* interprets a message using only the semantically compatible referents and possibly the prior probability of the target.

Constructing a literal speaker this way and then deriving two higher-level agents from it using (1.1) and (1.2), we have:

$$S_0(u \mid t, C) \propto \mathcal{L}(u, t, C) \exp\left(-\lambda K(u)\right) \tag{1.3}$$

$$L_1(t \mid u, C) \propto S_0(u \mid t, C) P(t) \tag{1.4}$$

$$S_2(u \mid t, C) \propto \exp\left(\lambda \left(\log L_1(t \mid u, C) - K(u)\right)\right) \tag{1.5}$$

The pragmatic speaker S_2 reasons not about the semantics directly but rather about a listener L_1 reasoning about a literal speaker S_0. Figure 1.1 tracks the RSA computations for the reference game in Figure 1.1a. Here, the message costs K are all 0, the prior over referents is flat, and $\lambda = 1$. The chances of success for the literal speaker S_0 are low, since it chooses true messages at random. In contrast, the chances of success for S_2 are high, since it derives the unambiguous system highlighted in gray.

1.2.2 Listener-based models

Section 1.2.1 described a model of agents derived from a literal speaker. More commonly, RSA models have started with a literal listener reasoning only in terms of the lexicon $\mathcal{L}$ and target priors. In such models, the literal listener chooses a random target consistent with the utterance, according to the prior distribution:

$$L_0(t \mid u, C) \propto \mathcal{L}(u, t, C) P(t) \tag{1.6}$$

A pragmatic speaker and pragmatic listener can then be derived using (1.1) and (1.2):

$$S_1(u \mid t, C) \propto e^{\lambda \log(L_0(t|u,C)) - K(u)} \tag{1.7}$$

$$L_2(t \mid u, C) \propto S_1(u \mid t, C) P(t) \tag{1.8}$$

Here, it is the literal listener that reasons about the semantics, while the speaker reasons about this listener.

It should be noted that much prior work in RSA (for example, Bergen et al., 2016) does not use both a literal listener and a literal speaker, and uses a different subscript convention: L_n is derived from S_n rather than S_{n-1}. In this work, the subscript consistently denotes the number of applications of an RSA recursion equation—(1.1) or (1.2)—on top of some base agent. This allows speaker-based and listener-based models to coexist without notational clashes: L_1 is always a listener based on a literal speaker, L_2 is a listener based on a speaker based on a literal listener, and so forth.

1.2.3 RSA's heritage

RSA is a descendant of a series of related models from game theory. The idea that a speaker chooses a message to maximize its expected utility in circumstances requiring coordination for an effective strategy, and that such strategies require the development of a conventional language, can be traced back to the signaling systems of Lewis (1969).

Most following work that has featured the sort of back-and-forth reasoning that RSA proposes has assumed that players choose among the highest expected-utility actions, and assume others do the same. Camerer et al. (2004) postulate a distribution over the number of recursions players do, but assumes each player only chooses an action if it maximizes expected utility. Iterated best response models (Franke, 2009; Jäger, 2011) consider the limit of this maximizing behavior as the number of recursions goes to infinity. This model is often poorly predictive of people's behavior, owing partly to the perfect maximization assumption. Instead, it is better to assume some probability that the other player will perform suboptimally; the iterated cautious response model (Jäger, 2014) computes all strategies that respond optimally to any probability distribution over the other player's strategies. Both of these models sometimes have problems with unrealistically broad sets of best actions that can result after convergence.

The use of the softmax distribution instead of a model that consistently chooses

the optimal message dates to at least Rosenberg and Cohen (1964), who modeled a word-based reference game with it. They noted that it is equivalent to assuming Luce's choice axiom (Luce, 1959). The quantal response equilibrium model (McKelvey and Palfrey, 1995) considered the infinite limit of recursive reasoning with softmax choice distributions, a limit which is guaranteed to exist and produces a highly-predictive statistical alternative to the Nash equilibrium. RSA combines this with the observation of Camerer et al. (2004) that people on average compute only a small number of recursions (1.61, in their estimate).

The RSA back-and-forth interpretive process can be thought of as a probabilistic formalization of conversational implicature as described by Grice (1975); it also reflects more general ideas from Bayesian cognitive modeling (Tenenbaum et al., 2011). Recent variants of RSA have been able to capture a wide range of linguistic phenomena that have proven difficult to model by other means, particularly those having to do with pragmatic (context-sensitive) or non-literal language use. While Figure 1.1 highlights the prediction that an efficient communication system can evolve from an ambiguous one, a model of the form given in Section 1.2.2 with a nearly identical context predicts scalar implicature (e.g., that *some* is generally interpreted as *some but not all*; Goodman and Stuhlmüller, 2013). RSA-based models have also yielded insight into metaphor (Kao et al., 2014a), hyperbole (Kao et al., 2014b), and one-shot word learning (Smith et al., 2013); see the conclusion, Section 6.1.1, for more discussion of this work. Finally, such models have been observed to produce computational systems that are more effective at communicating, either with human listeners (Tellex et al., 2014; Golland et al., 2010) or each other (Vogel et al., 2013, 2014).

1.3 Challenges

Given the effectiveness of RSA at capturing non-literal language, it is desirable to expand the implementation of RSA to serve in a general-purpose language interpretation or production system. However, accomplishing this is not straightforward. In this section I outline several apparent problems that prevent the RSA model as described above from being used in a broad-coverage implementation.

1.3.1 Coverage of the semantic interpretation function

The speaker-based and listener-based versions of RSA, as previously described, make use of an interpretation function $\mathcal{L}$. This is a function that must determine, for any pair of message and referent and any context, whether the message is true of the referent in that context. For example, in the context of the reference game in 1.1a, a person might produce the description *the happier-looking spectacled guy*. An effective interpretation function must know, among other things:

(i) how to measure the extent to which a person in an image is "happy-looking";

(ii) that "happier-looking" requires comparing two referents along this dimension and choosing the one with the greater value;

(iii) how to detect when a referent is wearing glasses;

(iv) that "spectacled" is true of a referent if the referent is wearing glasses;

(v) how to detect when a referent is a human male;

(vi) that "guy" is true of a referent if the referent is a human male; and

(vii) that the full message *the happier-looking spectacled guy* is true of one referent, which is found by first picking the subset of referents satisfying both (iv) and (vi), and then choosing one member of this set using the comparison in (ii),

to arrive at the value 1 if the referent is c_2 and 0 otherwise.

In most of the RSA work cited above, all such procedures are constructed manually. This is a task that borders on impossible in all but the most restricted settings. For one thing, even such a seemingly limited task as determining the emotion represented by a person's facial expression is the subject of an extensive body of research (see, e.g., Zeng et al., 2009). For another, the attempt to give a comprehensive definition of even a single word can require handling a long tail of unexpected usages to capture seemingly ordinary language—consider that the word *guy* or *guys* can be

used in everyday speech not just to refer to women (contrary to the assumption of (vi) that a "guy" must be male) but also to pets, or even mathematical symbols.[1]

1.3.2 Coverage of alternative utterances (and referents)

The definitions of S_0 (1.3) and S_1 (1.7) above hide a fundamental limitation: each proportionality relation requires the computation of a normalization constant that involves a sum over all utterances. That is, (1.3) in full takes the form

$$S_0(u \mid t, C) = \frac{\mathcal{L}(u, t, C) \exp\left(-\lambda K(u)\right)}{\sum_{u'} \mathcal{L}(u', t, C) \exp\left(-\lambda K(u')\right)} \tag{1.9}$$

with the sum in the denominator in theory considering all possible utterances u'. If this set of all possible utterances is not finite (as one would expect of the complete set of utterances generated by a compositional, recursive grammar), then in general there is no exact way to normalize the S_0 scores. In some cases the interpretation function $\mathcal{L}$ may factorize in a particularly clean way, thereby allowing the computation of this sum exactly for S_0. However, even in this case, no such computation is possible for S_1 and S_2's constants of proportionality in (1.7) and (1.5).

The enumeration of alternative utterances has implications for the cognitive plausibility of the model. It is unrealistic to imagine that human speakers consider the space of all possible utterances (Dale and Reiter, 1995). Rather, this search space must be limited by various factors, such as syntactic structure or recency of exposure. This expectation aligns with empirical findings about the relationship between syntactic constraints and pragmatic implicature (Chierchia, 2004; Collins, 2016) and response times after priming with potential alternatives (Degen and Tanenhaus, 2015).

1.3.3 Accounting for other factors influencing language use

RSA has also been criticized on the grounds that it predicts unrealistic speaker behavior (Gatt et al., 2013). For instance, in Figure 1.1, the agents are confined to a

[1] "Stanford women's basketball absolutely *crushed* the other guys yesterday." "I think the little guy on the left needs a walk." "This guy in the denominator is a constant, so we can ignore him."

simple message space. If permitted to use natural language, they will often produce utterances expressing predicates that are redundant from an RSA perspective—for example, by describing c_1 as *the man with the long beard and sweater*, even though *man* has no power to discriminate, and *beard* and *sweater* each uniquely identify the intended referent. This tendency has several explanations, including a preference for including certain kinds of descriptors, a desire to hedge against the possibility that the listener is not pragmatic, cognitive pressures that make optimal descriptions impossible, and syntactic expectations such as the requirement that the description of an object be a noun phrase.

1.3.4 Proposed solutions

In the remainder of this dissertation, I present various proposals for overcoming these obstacles. In short, I argue for learning $\mathcal{L}$ with a machine learning model, as well as sampling from a generation model to narrow the space of alternative utterances to a finite set.

Chapter 2 substitutes a simple machine learning model for the literal speaker S_0. This addresses the concerns of both Section 1.3.1 and Section 1.3.3: it removes the need to hand-design an interpretation function in favor of learning utterance meanings from example utterance-referent pairs, and it provides a way of accounting for preferences in human utterances that aren't captured by RSA's notions of semantic compatibility, target priors, and utterance costs.

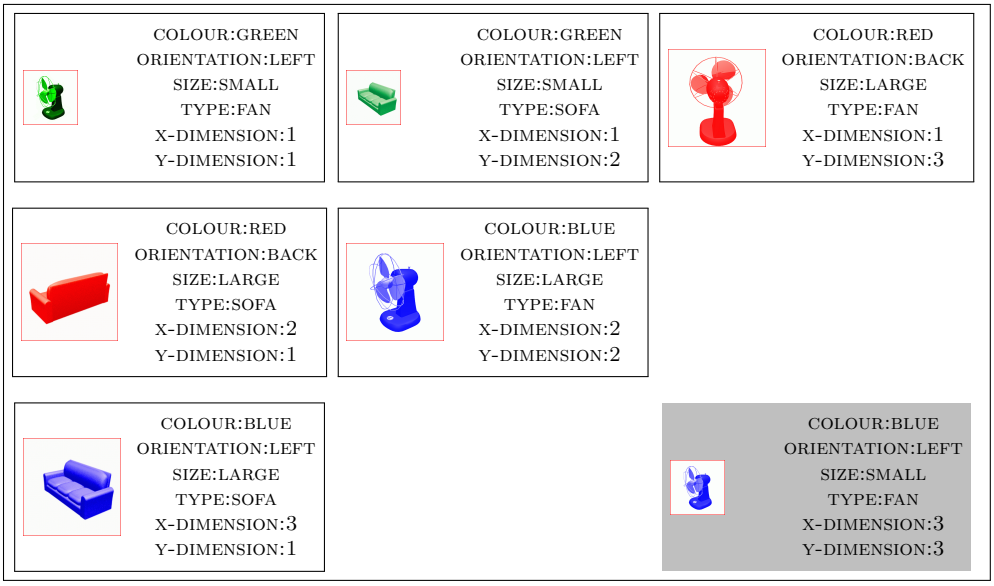
Chapter 4 goes further by replacing both L_0 and S_0 with RNN-based listener agents. In addition to its role as a base model for a pragmatic listener analogous to L_1 in (1.4), the S_0 agent is also used acquire sample utterances for tractably approximating the normalization required in defining the S_1 agent in (1.7). Speaker samples provide a solution to the problem raised in Section 1.3.2, while the use of RNN-based agents improves the expressiveness of the system for semantic interpretation (Section 1.3.1).

Chapter 2

Learning in the rational speech acts model

This chapter extends RSA by showing how to define it as a trained statistical classifier, which we call *learned RSA*. At the heart of learned RSA is the back-and-forth reasoning between speakers and listeners that characterizes RSA. However, whereas standard RSA requires a hand-built lexicon, learned RSA infers a lexicon from data. And whereas standard RSA makes predictions according to a fixed calculation, learned RSA seeks to optimize the likelihood of whatever examples it is trained on. Agents trained in this way exhibit the pragmatic behavior characteristic of RSA, but their behavior is governed by their training data and hence is only as rational as that experience supports. To the extent that the speakers who produced the data are pragmatic, learned RSA discovers that; to the extent that their behavior is governed by other factors, learned RSA picks up on that too.

We validate the model on the task of *attribute selection for referring expression generation* with a widely-used corpus of referential descriptions (the TUNA corpus; van Deemter et al., 2006; Gatt et al., 2007), showing that it improves on heuristic-driven models and pure RSA by synthesizing the best aspects of both.



Utterance: "blue fan small"
Utterance attributes: [*colour:blue*]; [*size:small*]; [*type:fan*]

Figure 2.1: Example item from the TUNA corpus. Target is in gray.

2.1 The TUNA corpus

In Section 2.4, we evaluate RSA and learned RSA in the TUNA corpus (van Deemter et al., 2006; Gatt et al., 2007), a widely used resource for developing and testing models of natural language generation. TUNA is a corpus of utterances collected from people playing a reference game (Section 1.2) with contexts of seven possible referents. Trials were performed in two domains, *furniture* and *people*, each with a *singular* condition (describe a single entity) and a *plural* condition (describe two). Figure 2.1 provides a (slightly simplified) example from the singular furniture section, with the target item identified by shading. In this case, the participant wrote the message "blue fan small". All entities and messages are annotated with their semantic attributes, as given in simplified form here. (Participants saw just the images; we include the attributes in Figure 2.1 for reference.)

The task for the speaker model in this chapter is *attribute selection*: reproducing the multiset of attributes in the message produced in each context. Thus, for Figure 2.1, we would aim to produce {[*size:small*], [*colour:blue*], [*type:fan*]}. Section 2.4

provides additional details on how to evaluate a model at this task.

Attribute selection is less demanding than full natural language generation, in that no lexical, morphological, or syntactic choices are required beyond determining which properties of the target to mention. This still requires computation exponential in the number of attributes if a model is to consider all possible utterances; however, in this dataset, the maximum number of attributes a referent can have is just small enough to keep the computation feasible. Chapter 3 introduces a class of models capable of generating complete natural language descriptions given a target's attributes, and Chapter 4 examines ways of making RSA compatible with such models without exponential amounts of computation.

2.2 Learned RSA

We now formulate RSA as a machine learning model that can incorporate the quirks and limitations that characterize natural descriptions while still presenting a unified model of pragmatic reasoning. This approach builds on the two-layer speaker-centric classifier of Golland et al. (2010), but differs from theirs in that we directly optimize the performance of the pragmatic speaker in training, whereas Golland et al. apply a recursive reasoning model on top of a pre-trained classifier. Like RSA, the model can be generalized to allow for additional intermediate agents, and it can easily be reformulated to begin with an analogue of a literal listener.

2.2.1 Feature representations

To build an agent that learns effectively from data, we must represent the items in our dataset in a way that accurately captures their important distinguishing properties and permits robust generalization to new items (Domingos, 2012; Liang and Potts, 2015). We define our feature representation function ϕ very generally as a map from state–utterance–context triples $\langle t, u, C \rangle$ to vectors of real numbers. This gives us the freedom to design the feature function to encode as much relevant information as necessary.

As noted above, in learned RSA, we do not presuppose a semantic lexicon, but rather induce one from the data as part of learning. The feature representation function determines a large, messy hypothesis space of potential lexica that is refined during optimization. For instance, as a starting point, we might define the feature space in terms of the cross-product of all possible entity attributes and all possible utterance meaning attributes. For m entity attributes and n utterance attributes, this defines each $\phi(t, u, C)$ as an mn-dimensional vector. Each dimension of this vector records the number of times that its corresponding pair of attributes co-occurs in t and u. Thus, the representation of the target entity in Figure 2.1 would include a 1 in the dimension for clearly good pairs like COLOUR:BLUE $\wedge$ [*colour:blue*] as well as for intuitively incorrect pairs like SIZE:SMALL $\wedge$ [*colour:blue*].

Because ϕ is defined very generally, we can also include information that is not clearly lexical. For instance, in our experiments, we add dimensions that count the color attributes in the utterance in various ways, ignoring the specific color values. We can also define features that intuitively involve negation, for instance, those that capture entity attributes that go unmentioned. This freedom is crucial to bringing generation-specific insights into the RSA reasoning.

2.2.2 Base speaker

Learned RSA is built on top of a *log-linear model*, standard in the machine learning literature and widely applied to classification tasks (Hastie et al., 2009; McCullagh and Nelder, 1989).

$$S_0(u \mid t, C; \theta) \propto \exp(\theta^T \phi(t, u, C)) \tag{2.1}$$

This model serves as our base speaker, analogous to the literal speaker S_0 in (1.3). The lexicon of this model is embedded in the *parameters* (or *weights*) θ, replacing the hand-built lexicon $\mathcal{L}$—accordingly, for the rest of this chapter, the models using the lexicon $\mathcal{L}$ will be denoted by $S_0^{\mathcal{L}}$ (1.3) and $S_2^{\mathcal{L}}$ (1.5).

Intuitively, θ is the direction in feature representation space that the base speaker believes is most positively correlated with the probability that the message will be correct. We train the model by searching for a θ to maximize the conditional likelihood

the model assigns to the messages in the training examples. Assuming the training is effective, this increases the weight for correct pairings between utterance attributes and entity attributes and decreases the weight for incorrect pairings.

To find the optimal θ, we seek to maximize the conditional likelihood of the training examples using first-order optimization methods (described in more detail in Section 2.2.4, below). This requires the gradient of the likelihood with respect to θ. To simplify the gradient derivation and improve numerical stability, we maximize the log of the conditional likelihood:

$$\begin{aligned} J_{S_0}(t, u, C, \theta) &= \log S_0(u \mid t, C; \theta) \\ &= \log \left[\exp(\theta^T \phi(t, u, C))\right] - \log \sum_{u'} \exp(\theta^T \phi(t, u', C)) \\ &= \theta^T \phi(t, u, C) - \log \sum_{u'} \exp(\theta^T \phi(t, u', C)) \end{aligned} \tag{2.2}$$

The gradient of this log-likelihood is

$$\begin{aligned} \frac{\partial J_{S_0}}{\partial \theta} &= \frac{\partial}{\partial \theta} \theta^T \phi(t, u, C) - \frac{\partial}{\partial \theta} \log \sum_{u'} \exp(\theta^T \phi(t, u', C)) \\ &= \phi(t, u, C) - \frac{1}{\sum_{u'} \exp(\theta^T \phi(t, u', C))} \sum_{u'} \exp(\theta^T \phi(t, u', C)) \phi(t, u', C) \\ &= \phi(t, u, C) - \sum_{u'} \frac{\exp(\theta^T \phi(t, u', C))}{\sum_{u''} \exp(\theta^T \phi(t, u'', C))} \phi(t, u', C) \\ &= \phi(t, u, C) - \sum_{u'} S_0(u' \mid t, C; \theta) \phi(t, u', C) \qquad (2.3) \\ &= \phi(t, u, C) - \mathbb{E}_{u' \sim S_0(\cdot \mid t, C; \theta)} \left[\phi(t, u', C)\right] \qquad (2.4) \end{aligned}$$

where step (2.3) is by substitution of the definition of S_0 (with expanded proportionality constant) in reverse.

2.2.3 Pragmatic speaker

We now define a pragmatic listener L_1 and a pragmatic speaker S_2. We will show experimentally (Section 2.4) that the learned pragmatic speaker S_2 agrees better

with human speakers on a referential expression generation task than either the base speaker S_0 or the pure RSA pragmatic speaker $S_2^{\mathcal{L}}$ from (1.5).

The parameters for L_1 and S_2 are still the parameters of the base speaker S_0; we wish to update them to maximize the performance of S_2, the agent that acts according to $S_2(u \mid t, C; \theta)$, where

$$S_2(u \mid t, C; \theta) \propto L_1(t \mid u, C; \theta) \tag{2.5}$$

$$L_1(t \mid u, C; \theta) \propto S_0(u \mid t, C; \theta) \tag{2.6}$$

This corresponds to a simplification of the speaker-based pragmatic RSA speaker, S_2 in (1.5), by setting $\lambda = 1$ and message costs and state priors to uniform: $S_2(u \mid t, C) \propto L_1(t \mid u, C) \propto S_0(u \mid t, C)$.

In optimizing the performance of the pragmatic speaker S_2 by adjusting the parameters to the simpler classifier S_0, the RSA back-and-forth reasoning can be thought of as a non-linear function through which errors are propagated in training, similar to the activation functions in neural network models (Rumelhart et al., 1986). However, unlike neural network activation functions, the RSA reasoning applies a different non-linear transformation depending on the pragmatic context (sets of available referents and utterances).

For convenience, we define symbols for the log-likelihood of each of these probability distributions:

$$J_{S_2}(t, u, C, \theta) = \log S_2(u \mid t, C; \theta) \tag{2.7}$$

$$J_{L_1}(t, u, C, \theta) = \log L_1(t \mid u, C; \theta) \tag{2.8}$$

The log-likelihood of each agent has the same form as the log-likelihood of the base speaker, but with the value of the distribution from the lower-level agent substituted for the score $\theta^T \phi$. By a derivation similar to the one in (2.4) above, the gradient of these log-likelihoods can thus be shown to have the same form as the gradient of the base speaker, but with the gradient of the next lower agent substituted for the feature

values:

$$\frac{\partial J_{S_2}}{\partial \theta} = \frac{\partial J_{L_1}}{\partial \theta}(t, u, C, \theta) - \mathbb{E}_{u' \sim S_2(\cdot|t,C;\theta)} \left[\frac{\partial J_{L_1}}{\partial \theta}(t, u', C, \theta) \right] \tag{2.9}$$

$$\frac{\partial J_{L_1}}{\partial \theta} = \frac{\partial J_{S_0}}{\partial \theta}(t, u, C, \theta) - \mathbb{E}_{t' \sim L_1(\cdot|u,C;\theta)} \left[\frac{\partial J_{S_0}}{\partial \theta}(t', u, C, \theta) \right] \tag{2.10}$$

The value J_{S_0} in (2.10) is as defined in (2.2).

2.2.4 Training

As mentioned above, our primary objective in training is to maximize the (log) conditional likelihood of the messages in the training examples given their respective states and contexts. We add to this an ℓ_2 regularization term,[1] which expresses a Gaussian prior distribution over the parameters θ. Imposing this prior helps prevent overfitting to the training data and thereby damaging our ability to generalize well to new examples (Chen and Rosenfeld, 1999). With this modification, we instead maximize the log of the posterior probability of the parameters and the training examples jointly. For a dataset of M training examples $\langle t_i, u_i, C_i \rangle$, this log posterior is:

$$J(\theta) = -\frac{M}{2}\ell||\theta||^2 + \sum_{i=1}^{M} \log S_1(u_i \mid t_i, C_i; \theta) \tag{2.11}$$

The stochastic gradient descent (SGD) family of first-order optimization techniques (Bottou, 2010) can be used to approximately maximize $J(\theta)$ by obtaining noisy estimates of its gradient and "hill-climbing" in the direction of the estimates. (Strictly speaking, we are employing stochastic gradient *ascent* to maximize the objective rather than minimize it; however, SGD is the much more commonly seen term for the technique.)

[1] Here ℓ_2 refers to the 2-norm of the vector of parameters θ. Another common notation is "L_2"; we use lowercase to avoid conflict with the notation for a twice-derived pragmatic listener.

The exact gradient of this objective function is

$$\frac{\partial J}{\partial \theta} = -M\ell\theta + \sum_{i=1}^{M} \frac{\partial J_{S_2}}{\partial \theta}(t_i, u_i, C_i, \theta) \tag{2.12}$$

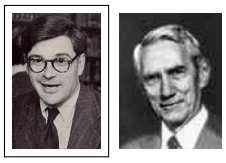
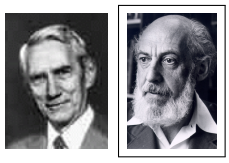
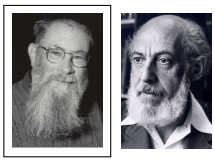
using the per-example gradient $\frac{dJ_{S_2}}{d\theta}$ given in (2.9).

In this chapter, gradients have been derived explicitly to make their role in the optimization process more transparent; however, most current implementations would use a software library that supports automatic gradient calculations (automatic or symbolic differentiation), making explicit derivations of the gradient unnecessary. Later chapters simply state the objective function J, which can be implemented with such a library.

SGD uses the per-example gradients (and a simple scaling of the ℓ_2 regularization penalty) as its noisy estimates, thus relying on each example to guide the model in roughly the correct direction towards the optimal parameter setting. Formally, for each example $\langle t, u, C \rangle$, the parameters are updated according to the formula

$$\theta \;:=\; \theta + \alpha \left(-\ell\theta + \frac{\partial J_{S_2}}{\partial \theta}(t, u, C, \theta) \right) \tag{2.13}$$

The learning rate α determines how "aggressively" the parameters are adjusted in the direction of the gradient. Small values of α lead to slower learning, but a value of α that is too large can result in the parameters overshooting the optimal value and diverging. To find a good learning rate, we use AdaGrad (Duchi et al., 2011), which sets the learning rate adaptively for each example based on an initial step size η and gradient history. The effect of AdaGrad is to reduce the learning rate over time such that the parameters can settle down to a local optimum despite the noisy gradient estimates, while continuing to allow high-magnitude updates along certain dimensions if those dimensions have exhibited less noisy behavior in previous updates.

	Training examples		Test example
Context			
	c_2 c_3	c_3 c_4	c_1 c_4
Utterance	[*person*] *with* [*glasses*]	[*person*] *with* [*beard*]	[*person*] *with* [*glasses*]
Features for true utterance	PERSON ∧ [*person*] PERSON ∧ [*glasses*] GLASSES ∧ [*person*] GLASSES ∧ [*glasses*] ¬BEARD ∧ [*person*] ¬BEARD ∧ [*glasses*]	PERSON ∧ [*person*] PERSON ∧ [*beard*] ¬GLASSES ∧ [*person*] ¬GLASSES ∧ [*beard*] BEARD ∧ [*person*] BEARD ∧ [*beard*]	PERSON ∧ [*person*] PERSON ∧ [*glasses*] GLASSES ∧ [*person*] GLASSES ∧ [*glasses*] BEARD ∧ [*person*] BEARD ∧ [*glasses*]

Gradient

	[*person*]	[*glasses*]	[*beard*]
PERSON	**1**	1	-1
GLASSES	**2**	**2**	**-2**
BEARD	0	0	0
¬GLASSES	-1	**-1**	**1**
¬BEARD	1	**1**	**-1**

	[*person*]	[*glasses*]	[*beard*]
PERSON	**1**	-1	1
GLASSES	0	0	0
BEARD	**2**	**-2**	**2**
¬GLASSES	1	**-1**	**1**
¬BEARD	-1	**1**	**-1**

(unused)

(a) Learned S_2 model training. Gradient values given are $6\frac{\partial J_{S_2}}{\partial \theta}$, evaluated at $\theta = \vec{0}$.

	c_1	c_4		c_1	c_4		c_1	c_4	
$S_2^{\mathcal{L}}$	.08	.25	S_0	.03	.00	S_2	.10	.11	$\emptyset$
	.08	.25		.22	.10		.16	.13	[*person*]
	.17	0		.03	.00		.11	.07	[*glasses*]
	.08	.25		.03	.04		.08	.17	[*beard*]
	.17	0		.22	.01		.18	.08	[*person*], [*glasses*]
	.08	.25		.22	.74		.12	.19	[*person*], [*beard*]
	.17	0		.03	.00		.10	.11	[*glasses*], [*beard*]
	.17	0		.22	.10		.16	.11	[*person*], [*glasses*], [*beard*]

(b) RSA with hand-built lexicon ($S_2^{\mathcal{L}}$), linear classifier (S_0), and learned RSA (S_2) utterance distributions. RSA alone minimizes ambiguity but can't learn overgeneration from the examples. The linear classifier learns to produce [*person*] but fails to minimize ambiguity. The weights in learned RSA retain the tendency to produce [*person*] in all cases, while the recursive reasoning yields a preference for the unambiguous descriptor [*glasses*].

Figure 2.2: Specificity implicature and overgeneration in learned RSA.

2.3 Example

In Figure 2.2, we illustrate crucial aspects of how our model is optimized, fleshing out the concepts from the previous section. The example also shows the ability of the trained S_2 model to make a specificity implicature without having observed one in its data, while preserving the ability to produce uninformative attributes if encouraged to do so by experience.

As in our main experiments, we frame the learning task in terms of attribute selection with TUNA-like data. In this toy experiment, the agent is trained on two example contexts, consisting of a target referent, a distractor referent, and a human-produced utterance. It is evaluated on a third test example. This small dataset is given in the top two rows of Figure 2.2. The utterance on the test example is shown for comparison; it is not provided to the agent.

Our feature representations of the data are in the third row. Attributes of the referents are in SMALL CAPS; semantic attributes of the utterances are in [*square brackets*]. These representations employ the cross-product features described in Section 2.2; in TUNA data, properties that the target entities do not possess (e.g., ¬GLASSES) are also included among their "attributes."

Below the feature representations, we summarize the gradient of the log likelihood ($\frac{\partial J_{S_2}}{\partial \theta}$) for each example, as an $m \times n$ table representing the weight update for each of the mn cross-product features. (We leave out the ℓ_2 regularization and AdaGrad learning rate for simplicity.) Tracing the formula for this gradient (2.9) back through the RSA layers to the base speaker (2.2), one can see that the gradient consists of the feature representation of the triple $\langle t, u, C \rangle$ containing the correct (human-produced) message, minus adjustments that penalize the other messages according to how much the model was "fooled" into expecting them.

The RSA reasoning yields gradients that express both lexical and contextual knowledge. From the first training example, the model learns the lexical information that [*person*] and [*glasses*] should be used to describe the target. However, this knowledge receives higher weight in the association with GLASSES, because that attribute is disambiguating in this context. As one would hope, the overall result is

that intuitively good pairings generally have higher weights, though the training set is too small to fully distinguish good features from bad ones. For example, after seeing both training examples and failing to observe both a beard and glasses on the same individual, the model incorrectly infers that [*beard*] can be used to indicate a lack of glasses and vice versa. Additional training examples could easily correct this.

Figure 2.2b shows the distribution over utterances given target referent as predicted by the learned pragmatic speaker S_2 after one pass through the data with a fixed learning rate $\alpha = 1$ and no regularization ($\ell = 0$). We compare this distribution with those predicted by the learned base speaker S_0 and the pure RSA speaker $S_2^{\mathcal{L}}$. We wish to determine whether each model can (i) minimize ambiguity; and (ii) learn a prior preference for producing certain descriptors even if they are redundant.

The distributions in Figure 2.2b show that the linear classifier correctly learns that human-produced utterances in the training data tend to mention the attribute [*person*] even though it is uninformative. However, for the referent that was not seen in the training data, the model cannot decide among mentioning [*beard*], [*glasses*], both, or neither, even though the messages that don't mention [*glasses*] are ambiguous in context. The pure RSA model, meanwhile, chooses messages that are unambiguous, but because it has no mechanism for learning from the examples, it does not prefer to produce [*person*] without a manually-specified prior.

Our pragmatic speaker S_2 gives us the best of both models: the parameters θ in learned RSA show the tendency exhibited in the training data to produce [*person*] in all cases, while the RSA recursive reasoning mechanism guides the model to produce unambiguous messages by including the attribute [*glasses*].

2.4 Experiments

2.4.1 Data

We report experiments on the TUNA corpus (Section 2.1 above). We focus on the *singular* portion of the corpus, which was used in the 2008 and 2009 Referring Expression Generation Challenges. We do not have access to the train/dev/test splits

from those challenges, so we report five-fold cross-validation numbers. The singular portion consists of 420 *furniture* trials involving 176 distinct referents and 360 *people* trials involving 228 distinct referents.

2.4.2 Evaluation metrics

The primary evaluation metric used in the attribute selection task with TUNA data is *multiset Dice* calculated on the attributes of the generated messages:

$$\frac{2\sum_{x\in u\cup u^*}\min\left[\#_u(x),\#_{u^*}(x)\right]}{|u|+|u^*|} \tag{2.14}$$

Here, u is a multiset of attributes predicted by the model, u^* is a multiset of attributes mentioned by a human speaker in the evaluation set, $\cup$ denotes non-multiset union, $\#_X(x)$ is the number of occurrences of x in the multiset X, and $|u|$ is the cardinality of multiset u. In the case that no attributes are mentioned more than once in either the model output or the human utterance, this is equivalent to the F_1 precision-recall measure of the overlap between the two. (The human utterances can mention attributes more than once, hence their treatment as multisets. However, multiple mentions of the same attribute is rare enough that restricting our models to mentioning each attribute at most once does not severely reduce the maximum possible Dice coefficient or accuracy.)

Accuracy is the fraction of examples for which the subset of attributes is predicted perfectly (equivalent to achieving multiset Dice 1).

2.4.3 Experimental setup

We evaluate all our agents in the same pragmatic contexts: for each trial in the *singular* corpus, we define the set of all utterances u to be the powerset of the attributes used in the referential description and the set of referents C to be the set of entities in the trial, including the target t. The message predicted by a speaker agent is the one with the highest probability given the target entity; if more than one message has the highest probability, we allow the agent to choose randomly from the highest

probability ones.

In learning, we use initial step size $\eta = 0.01$ and regularization constant $\ell = 0.01$. RSA agents are not trained, but we cross-validate to optimize λ and the function defining message costs, choosing from (i) $K(u) = 0$; (ii) $K(u) = |a(u)|$; and (iii) $K(u) = -|a(u)|$.

2.4.4 Features

We use *indicator features* as our feature representation; that is, the dimensions of the feature representation take the values 0 and 1, with 1 representing the truth of some predicate $P(t, u, C)$ and 0 representing its negation. Thus, each vector of real numbers that is the value of $\phi(t, u, C)$ can be represented compactly as a set of predicates.

The baseline feature set consists of indicator features over all conjunctions of an attribute of the referent and an attribute in the candidate message (e.g., $P(t, u, C) = \textsc{red}(t) \wedge [\textit{blue}] \in u$). We compare this to a version of the model with additional *generation features* that seek to capture the preferences identified in prior work on generation. These consist of indicators over the following features of the message:

(i) attribute type (e.g., $P(t, u, C) =$ "u contains a color");

(ii) pair-wise attribute type co-occurrences, where one can be negated (e.g., "u contains a color and a size", "u contains an object type but not a color"); and

(iii) message size in number of attributes (e.g., "u consists of 3 attributes").

For comparison, we also separately train base speakers S_0 as in (2.1) (the log-linear model) with each of these feature sets using the same optimization procedure.

2.4.5 Results

The results (Table 2.1) show that training a speaker agent with learned RSA generally improves generation over the ordinary classifier and RSA models. On the more complex *people* dataset, the pragmatic S_2 model significantly outperforms all other models. The value of the model's flexibility in allowing a variety of feature designs

Table 2.1: Learned RSA experimental results: mean accuracy and multiset Dice (five-fold cross-validation). **Bold**: best result; ***bold italic***: not significantly different from best ($p > 0.05$, Wilcoxon signed-rank test).

	Furniture		People		All	
Model	Acc.	Dice	Acc.	Dice	Acc.	Dice
RSA $S_0^{\mathcal{L}}$ (random true message)	1.0%	.475	0.6%	.125	1.7%	.314
RSA $S_2^{\mathcal{L}}$	1.9%	.522	2.5%	.254	2.2%	.386
Learned S_0, basic feats.	16.0%	.779	9.4%	.697	12.9%	.741
Learned S_0, gen. feats. only	5.0%	.788	7.8%	.681	6.3%	.738
Learned S_0, basic + gen. feats.	**28.1**%	**.812**	17.8%	.730	***23.3***%	***.774***
Learned S_2, basic feats.	23.1%	.789	11.9%	.740	17.9%	.766
Learned S_2, gen. feats. only	17.4%	.740	1.9%	.712	10.3%	.727
Learned S_2, basic + gen. feats.	***27.6***%	.788	**22.5**%	**.764**	**25.3**%	**.777**

can be seen in the comparison of the different feature sets: we observe consistent gains from adding generation features to the basic cross-product feature set. Moreover, the two types of features complement each other: neither the cross-product features nor the generation features in isolation achieve the same performance as the combination of the two.

Of the models in Table 2.1, all but the last exhibit systematic errors. Pure RSA performs poorly for reasons predicted by Gatt et al. (2013)—for example, it under-produces color terms and head nouns like *desk*, *chair*, and *person*. This problem is also observed in the trained S_2 model, but is corrected by the generation features. On the *people* dataset, the S_0 models under-produce *beard* and *hair*, which are highly informative in certain contexts. This type of communicative failure is eliminated in the S_2 speakers.

The performance of the learned RSA model on the *people* trials also compares favorably to the best dev set performance numbers from the 2008 Challenge (Gatt et al., 2008), namely, .762 multiset Dice, although this comparison must be informal since the test sets are different. (In particular, the Accuracy values given by Gatt et al. are unfortunately not comparable with the values we present, as they reflect "perfect match with *at least one* of the two reference outputs" [emphasis in original].)

Together, these results show the value of being able to train a single model that synthesizes RSA with prior work on generation.

2.5 Discussion

The experiments in this chapter demonstrate the utility of RSA as a trained classifier in generating referential expressions. The primary advantages of this version of RSA stem from the flexible ways in which it can learn from available data. This not only removes the need to specify a complex semantic lexicon by hand, but it also provides the analytic freedom to create models that are sensitive to factors guiding natural language production that are not naturally expressed in standard RSA.

However, the model described here is still limited in its generality. First, as discussed in Section 2.1, the output space of the model is a small set of pre-specified attributes of the referent. This output space does not capture the full complexity of natural language generation, neglecting morphology, syntactic constraints, and other factors in producing idiomatic descriptions.

Second, the model's representation of the referents is still specified by hand. The use of a log-linear model requires that any representation of the input that is usable for classification decisions must be directly encoded in the features. Although the use of feature templates allows a substantial reduction in work compared to the manual construction of a semantic lexicon, any manually-built feature set is bound to be incomplete in its coverage of useful information for making semantic decisions, meaning that the problem of lexical coverage discussed in Section 1.3.1 is not completely solved by this approach.

Third, the full context is not explicitly captured by the features. Instead, the RSA reasoning is the only source of contextual considerations. This prevents accurate representation of the semantics of superlatives (*the tallest person*), for example, which require computation involving objects in the context other than the target for semantic judgments.

In the next chapter, we discuss a model architecture that allows for a more flexible representation of both the input and the output. We show that this model can

generate grammatical descriptions in natural language that capture compositional aspects of meaning, and that it is able to capture complex properties of the input objects without minimal manual feature design.

Chapter 3

Generating color descriptions

The previous chapter demonstrated the combination of learned semantics with a model structure based on RSA to decide what to mention about the target object that would distinguish it from the distractors in context. One of its main limitations is its incomplete representations of the input (referent) and output (utterance). The model described in this chapter addresses these limitations by using a different representation structure, one which can in theory learn arbitrary features of the input and how it relates to the output without requiring those features to be anticipated in the model's design.

To investigate the ability of the model to capture arbitrary features of the input, we move from a domain with referents that are represented as a small number of discrete features to one with referents in a continuous space. Our chosen domain for this work is color. The production of color language is essential for referring expression generation (Krahmer and Van Deemter, 2012) and image captioning (Kulkarni et al., 2011; Mitchell et al., 2012), among other grounded language generation problems. Furthermore, color descriptions represent a microcosm of grounded language semantics. Basic color terms like *red* and *blue* provide a rich set of semantic building blocks; because the space of colors is continuous, these terms are necessarily vague, requiring models of color language to represent the boundaries of denotations in a soft or probabilistic way. From these building blocks, more complex color descriptions can be composed to express meanings not covered by basic terms, such as *greenish*

blue or *the color of the rust on my aunt's old Chevrolet* (Berlin and Kay, 1969).

This chapter presents an effective model of color description generation as a grounded language modeling problem. It combines a Fourier-basis representation of colors, inspired by feature representations in computer vision, with a powerful and general model of language production based on a *recurrent neural network*.

We compare our model with LUX (McMahan and Stone, 2015), a *Bayesian generative model* of color description generation. Generative models of language production have a much longer history than the RSA formalism used in this dissertation; for example, Raviv (1967) applies such an approach to optical character recognition. However, it is interesting to note the conceptual similarities between LUX and the design of the learned pragmatic speaker articulated in Chapter 2: LUX uses a machine learning model with a domain-specific feature representation to specify a literal listener, combines this with a prior over utterances to derive a speaker, and trains the model to maximize the probability of training examples on the speaker task. That is, it can be considered a learned RSA S_1 model (as opposed to the S_2 model of Chapter 2).

Our model improves on their approach in several respects, which we demonstrate by examining the meanings it assigns to various unusual descriptions: (1) it can generate compositional color descriptions not observed in training (Figure 3.3); (2) it learns correct denotations for underspecified modifiers, which name a variety of colors (*dark*, *dull*; Figure 3.2); and (3) it can model non-convex denotations, such as that of *greenish*, which includes both greenish yellows and blues (Figure 3.4). As a result, our model also produces significant improvements on several grounded language modeling metrics.

The model used in this chapter does not meaningfully employ the RSA insight. However, the improvements over LUX do not necessarily disparage the value of the Bayesian generative paradigm (and by corollary, the use of an RSA S_1 model). Rather, they come primarily from the increased flexibility offered by the recurrent neural network (RNN) design. This chapter examines the benefits of using an RNN for grounded language production. Chapter 4 will then combine the representational power of the

RNN with the back-and-forth, context-sensitive reasoning of RSA, showing specifically that a blend of RNN-based RSA L_1 and L_2 models outperforms an RNN-only baseline.

3.1 Recurrent neural network sequence modeling

The model presented in this chapter is built on the *long short-term memory (LSTM) recurrent neural network* architecture (Hochreiter and Schmidhuber, 1997; Graves, 2013), a widely-used system for modeling generic sequence representation and production. As a baseline for evaluation, this work also includes a model that does not capture the sequence structure of color descriptions, a *feedforward neural network* or *multi-layer perceptron*. Both of these models transform input x to output y by means of a domain-agnostic representation h in a continuous vector space. The functions mapping x to h and h to y are optimized (using general-purpose numerical optimization techniques including SGD—see Section 2.2.4) to assign a high probability to example input-output pairs.

The use of a real-valued vector representation is in contrast to symbolic approaches, which retain discrete features of the input for use in making decisions about the output; and linear machine learning models, which have a single learned, continuous operation mapping input to output without an intermediate representation. In the following discussion I use the term *neural* to refer to this collection of properties. Although the term originated from real similarities between the optimization process and mechanisms of activation and learning in human neurons, here it is not intended to make specific claims about human cognition; it is simply a label for a particular family of algorithms.

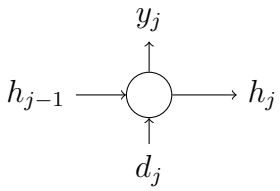
3.1.1 Embeddings

Neural models typically require the input to be a real-valued vector, like the intermediate representation. If the input is not already real-valued, this requires a method for converting between the discrete objects in question and some continuous input

representation. In representing language, at least one of input or output is usually a discrete linguistic object (most often a word or sequence of words, but sometimes characters, whole sentences, or longer documents can be the main unit of language representation). If language is the input x, it must be converted to a real-valued vector $d = D(x)$. The vector d is known as an *embedding* of the input. Many ways exist of constructing embeddings for words; the method we use in this dissertation is the one most commonly used with neural models, which is to initialize a lookup table randomly and allow the optimization algorithm to learn embeddings that are appropriate for the task (Bengio et al., 2003).

3.1.2 Recurrent cell

What distinguishes a recurrent neural network from a feed-forward neural network is the operation that reduces a sequence of inputs to a single representation. This operation is often called the *cell*. It is a function that takes in an internal *state* of the neural network as well as the embedding of the current input from the sequence, and produces two vectors: the new internal state and an output representation of the whole sequence so far. This can be represented schematically as a node with two inputs and two outputs:



In this diagram, j represents the index (time step) along the sequence, h is used for the internal state, and y is used for the output representation.

The definition of the LSTM cell is

$$
\begin{aligned}
i_j &= \sigma(W_{di}d_j + W_{yi}y_{j-1} + w_{hi} \odot h_{j-1} + b_i) \\
f_j &= \sigma(W_{df}d_j + W_{yf}y_{j-1} + w_{hf} \odot h_{j-1} + b_f) \\
h_j &= f_j \odot h_{j-1} + i_j \odot \sigma(W_{xh}d_j + W_{yh}y_{j-1} + b_h) \\
o_j &= \sigma(W_{do}d_j + W_{yo}y_{j-1} + w_{ho} \odot h_j + b_o) \\
y_j &= o_j \odot \sigma_y(h_j)
\end{aligned} \tag{3.1}
$$

This definition is from Graves (2013) and implemented in Lasagne (Dieleman et al., 2015), a neural network library based on Theano (Al-Rfou et al., 2016). Here σ denotes the *logistic function* (or *sigmoid function*), $\sigma(z) = \frac{1}{1+e^{-z}}$, and σ_y is a configurable nonlinear function that is applied to the output, commonly $\sigma_y(z) = \tanh(z)$.

3.1.3 Output layers

If the output is a sequence of words, then the output representation is used to construct a probability distribution over the output word at each time step. The construction of this probability distribution consists of two transformations to the output representation. The first is a *fully-connected layer*, which allows the same output representation to be used for multiple different types of outputs:

$$z = \sigma_y(W_{yz}y_j + b_z) \tag{3.2}$$

The dimensionality of z is the number of possible output words, denoted $|V|$.

The second is a *softmax function*, which maps the unconstrained vector z to a probability distribution over the $|V|$ words:

$$S(u_j = k|u_{1:j-1}, t) = \frac{e^{z_k}}{\sum_{k'} e^{z_{k'}}} \tag{3.3}$$

The probability of a sequence is the product of probabilities of the output tokens

up to and including the end token `</s>`:

$$S(u \mid t) = \prod_{j=1}^{|u|} S(u_j | u_{1:j-1}, t) \tag{3.4}$$

3.1.4 Training

As in the log-linear model of Section 2.2, neural models are typically trained with variants of SGD. Whereas the log-linear model had a single vector of parameters θ that are optimized in training, the RNN has a number of vectors and matrices that parameterize its objective function. The trainable parameters of the model are the word vectors d, the weights of the LSTM cell (W_{**}, w_{**}, b_*), and the weights of the fully-connected output layer (W_{yz}, b_z). To save space, these parameters are collectively denoted by θ, just like the log-linear model's parameters.

The full objective function of the model is the log conditional likelihood of a complete training set of M example pairs $\langle t_i, u_i \rangle$:

$$J(\theta) = \prod_{i=1}^{M} S(u_i \mid t_i) \tag{3.5}$$

Using ℓ_2 regularization is possible with an RNN model just as with a log-linear model; however, other forms of regularization are more common, particularly dropout (Hinton et al., 2012).

3.2 Benefits and tradeoffs of RNNs

3.2.1 Completeness

The main benefit of the use of neural sequence models is its completeness in representing the input with minimal feature design. An RNN sequence model produces an output representation at every time step, thus assigning a "meaning" to every partial sequence. It defines a function that makes use of the entire input in choosing the output, and can theoretically capture arbitrary continuous relationships between the

two, if the hidden state is sufficiently large. This reduces the amount of engineering of input preprocessing that is necessary to ensure that important information about the input is not left out.

3.2.2 Common representation space

Another advantage of the neural approach is that it provides a common representation space for linguistic and non-linguistic objects: any type of data that can be converted to a real-valued vector of constant dimensionality can be used as input to a neural model alongside embeddings of words. In this chapter and Chapter 4, this means representations of colors and sequences of colors; Chapter 5 also includes a binary flag to represent a choice between languages (English or Chinese) with its other inputs. The shared representation space makes techniques like multitask learning, in which one model can be trained to produce a representation that is useful for multiple outputs, particularly easy.

3.2.3 Interpretability

Neural models offer this simplicity and completeness at the expense of interpretability: the weights determining the internal representation don't usually have clear meanings, and when the model makes the wrong choice, it is often unclear why it has made that choice or how best to fix it. Incorrect choices by a linear model can more easily be traced to poorly chosen weights or missing components of the feature function (though this becomes more difficult the more features are added).

However, neural models can often still be interpreted at the input and output layers, which we do in Section 3.5 by plotting the model's probabilities as a function of the input, and in Section 5.4.1 by examining geometric relationships between the weights in the fully-connected layer for words that are translations of each other in English and Chinese.

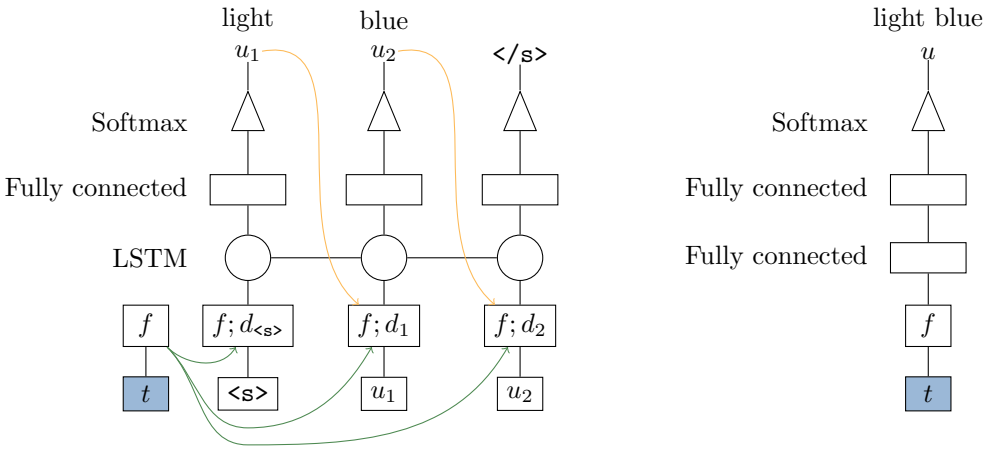


Figure 3.1: Left: sequence model architecture; right: atomic-description baseline.

3.3 Model formulation

Formally, a model of color description generation is a probability distribution $S(u \mid t)$ over sequences of tokens u conditioned on a color t, where t is represented as a 3-dimensional real vector in HSV space.[1]

3.3.1 Neural network architecture

Our main model is a recurrent neural network sequence decoder (Figure 3.1, left panel). An input color $t = (h, s, v)$ is mapped to a representation f (see Color features, below). At each time step, the model takes in a concatenation of f and an embedding for the previous output token u_i, starting with the start token $u_0 =$ <s>. This concatenated vector is passed as the input x to the LSTM layer, the output of which is used to produce a probability distribution for the following token as described in Section 3.1.3.

The model is substantively similar to well-known models for image caption generation (Karpathy and Fei-Fei, 2015; Vinyals et al., 2015), which use the output of a convolutional neural network as the representation of an input image and provide

[1]HSV: hue-saturation-value. The visualizations and tables in this chapter instead use HSL (hue-saturation-lightness), which yields somewhat more intuitive diagrams and differs from HSV by a trivial reparameterization.

this representation to the RNN as an initial state or first word (we represent the target color using a deterministic feature function described in the next section and concatenate the color representation onto each input word vector).

We also implemented a simple feed-forward neural network, to demonstrate the value gained by modeling descriptions as sequences. This architecture (*atomic*; Figure 3.1, right panel) consists of two fully-connected hidden layers, with a ReLU nonlinearity after the first and a softmax output over all full color descriptions seen in training. This model therefore treats the descriptions as atomic symbols rather than sequences.

3.3.2 Color features

We compare three representations:

- *Raw*: The original 3-dimensional color vectors, in HSV space.
- *Buckets*: A discretized representation, dividing HSV space into rectangular regions at three resolutions (90×10×10, 45×5×5, 1×1×1) and assigning a separate embedding to each region.
- *Fourier*: Transformation of HSV vectors into a Fourier basis representation. Specifically, the representation f of a color $\langle h, s, v \rangle$ is given by

$$\hat{f}_{jk\ell} = \exp\left[-2\pi i \left(jh^* + ks^* + \ell v^*\right)\right]$$
$$f = \left[\Re\hat{f} \quad \Im\hat{f}\right] \qquad j, k, \ell = 0..2$$

  where $\langle h^*, s^*, v^* \rangle = \langle h/360, s/200, v/200 \rangle$.

The Fourier representation is inspired by the use of Fourier feature descriptors in computer vision applications (Zhang and Lu, 2002). It is a nonlinear transformation that maps the 3-dimensional HSV space to a 54-dimensional vector space. This representation has the property that most regions of color space denoted by some description are extreme along a single direction in Fourier space, thus largely avoiding the need for the model to learn non-monotonic functions of the color representation.

3.3.3 Training

We again train using Adagrad (Duchi et al., 2011) with initial learning rate $\eta = 0.1$, hidden layer size and cell size 20, and dropout (Hinton et al., 2012) with a rate of 0.2 on the output of the LSTM and each fully-connected layer. We identified these hyperparameters with random search, evaluating on a held-out subset of the training data.

We use random normally-distributed initialization for embeddings ($\sigma = 0.01$) and LSTM weights ($\sigma = 0.1$), except for forget gates, which are initialized to a constant value of 5. Dense weights use normalized uniform initialization (Glorot and Bengio, 2010).

3.4 Experiments

We demonstrate the effectiveness of our model using the same data and statistical modeling metrics as McMahan and Stone (2015).

3.4.1 Data

The dataset used to train and evaluate our model consists of pairs of colors and descriptions collected in an open online survey (Munroe, 2010). Participants were shown a square of color and asked to write a free-form description of the color in a text box. McMahan and Stone filtered the responses to normalize spelling differences and exclude spam responses and descriptions that occurred very rarely. The resulting dataset contains 2,176,417 pairs divided into training (1,523,108), development (108,545), and test (544,764) sets.

3.4.2 Evaluation metrics

We quantify model effectiveness with the following evaluation metrics:

- *Perplexity*: The geometric mean of the reciprocal probability assigned by the model to the descriptions in the dataset, conditioned on the respective colors.

Model	Features	Perplexity	AIC	Accuracy
atomic	raw	28.31	1.08×10^6	28.75%
atomic	buckets	16.01	1.31×10^6	38.59%
atomic	Fourier	15.05	8.86×10^5	38.97%
RNN	raw	13.27	8.40×10^5	40.11%
RNN	buckets	13.03	1.26×10^6	39.94%
RNN	Fourier	**12.35**	$\mathbf{8.33\times10^5}$	**40.40%**
HM	buckets	14.41	4.82×10^6	39.40%
LUX	raw	13.61	4.13×10^6	39.55%
RNN	Fourier	**12.58**	$\mathbf{4.03\times10^6}$	**40.22%**

Table 3.1: Experimental results. Top: development set; bottom: test set. AIC is not comparable between the two splits. HM and LUX are from McMahan and Stone (2015). We reimplemented HM and re-ran LUX from publicly available code, confirming all results to the reported precision except perplexity of LUX, for which we obtained a figure of 13.72.

That is, for a test set consisting of M color-description pairs $\langle t_i, u_i \rangle$:

$$\text{ppl} = \left[\prod_{i=1}^{M} \frac{1}{S(u_i \mid t_i)} \right]^{1/M}$$

This expresses the same objective as log conditional likelihood, because it differs only in dividing by the number of examples and taking the exponential (both of which do not change the relative ordering of models). We follow McMahan and Stone (2015) in reporting perplexity per-description, not per-token as in the language modeling literature.

- *AIC*: The Akaike information criterion (Akaike, 1974) is given by AIC $= 2\ell + 2k$, where ℓ is log likelihood and k is the total number of real-valued parameters of the model (e.g., weights and biases, or bucket probabilities). This quantifies a tradeoff between accurate modeling and model complexity.

- *Accuracy*: The percentage of most-likely descriptions predicted by the model that exactly match the description in the dataset (recall@1).

3.4.3 Results

The top section of Table 3.1 shows development set results comparing modeling effectiveness for atomic and sequence model architectures and different features. The Fourier feature transformation generally improves on raw HSV vectors and discretized embeddings. The value of modeling descriptions as sequences can also be observed in these results; the LSTM models consistently outperform their atomic counterparts.

Additional development set experiments (not shown in Table 3.1) confirmed smaller design choices for the recurrent architecture. We evaluated a model with two LSTM layers, but we found that the model with only one layer yielded better perplexity. We also compared the LSTM with GRU and vanilla recurrent cells; we saw no significant difference between LSTM and GRU, while using a vanilla recurrent unit resulted in unstable training. Also note that the color representation f is input to the model at every time step in decoding. In our experiments, this yielded a small but significant improvement in perplexity versus using the color representation as the initial state.

Test set results appear in the bottom section. Our best model outperforms both the histogram baseline (HM) and the improved LUX model of McMahan and Stone (2015), obtaining state-of-the-art results on this task. Improvements are highly significant on all metrics ($p < 0.001$, approximate permutation test, $R = 10{,}000$ samples; Padó 2006).

3.5 Analysis

Given the general success of LSTM-based models at generation tasks, it is perhaps not surprising that they yield good raw performance when applied to color description. The color domain, however, has the advantage of admitting faithful visualization of descriptions' semantics: colors exist in a 3-dimensional space, so a two-dimensional visualization can show an acceptably complete picture of an entire distribution over the space. We exploit this to highlight three specific improvements our model realizes over previous ones.

We construct visualizations by querying the model for the probability $S(u \mid t)$

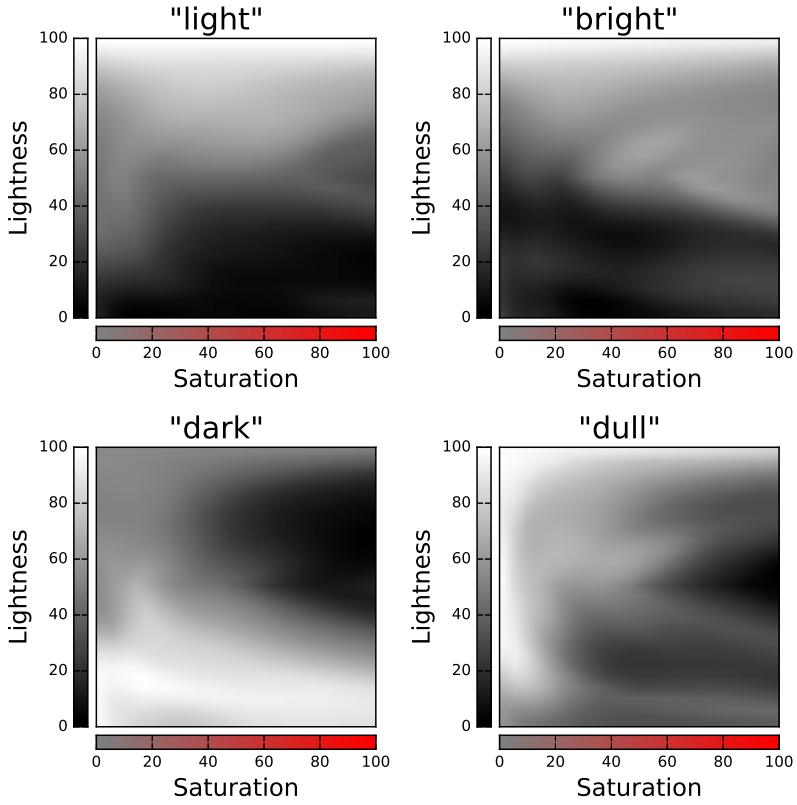


Figure 3.2: Conditional likelihood of bare modifiers according to our generation model as a function of color. White represents regions of high likelihood. We omit the hue dimension, as these modifiers do not express hue constraints.

of the same description for each color in a uniform grid, summing the probabilities over the hue dimension (left cross-section) and the saturation dimension (right cross-section), normalizing them to sum to 1, and plotting the log of the resulting values as a grayscale image. Formally, each visualization is a pair of functions $\langle L, R \rangle$, where

$$L(s, \ell) = \log \left[\frac{\int dh \; S(u \mid t = \langle h, s, \ell \rangle)}{\int dt' \; S(u \mid t')} \right]$$
$$R(h, \ell) = \log \left[\frac{\int ds \; S(u \mid t = \langle h, s, \ell \rangle)}{\int dt' \; S(u \mid t')} \right]$$

The maximum value of each function is plotted as white, the minimum value is black, and intermediate values linearly interpolated.

Color	Top-1	Sample
(83, 80, 28)	*green*	*very green*
(232, 43, 37)	*blue*	*royal indigo*
(63, 44, 60)	*olive*	*pale army green*
(39, 83, 52)	*orange*	*macaroni*

Table 3.2: A selection of color descriptions sampled from our model that were not seen in training. Color triples are in HSL. *Top-1* shows the model's highest-probability prediction.

3.5.1 Learning modifiers

As noted in Section 3.2.1, an RNN model can build a representation for arbitrary sequences and partial sequences. By visualizing the model's output distribution, we can see that the model learns accurate meanings of adjectival modifiers apart from the full descriptions that contain them. We examine this in Figure 3.2, by plotting the probabilities assigned to the bare modifiers *light*, *bright*, *dark*, and *dull*. *Light* and *dark* unsurprisingly denote high and low lightness, respectively. Less obviously, they also exclude high-saturation colors. *Bright*, on the other hand, features both high-lightness colors and saturated colors—*bright yellow* can refer to the prototypical, highly saturated yellow, whereas *light yellow* cannot. Finally, *dull* denotes unsaturated colors in a variety of lightnesses.

3.5.2 Compositionality

A consequence of the completeness of RNN sequence modeling is that such models can put words together into sequences that have not been seen before. Our model is able to produce descriptions not found in the training set, such as those shown in Table 3.2. However, for such new sequences to be useful, they should be *compositional*: the meaning of the whole should be constructed from the meaning of the parts. Figure 3.3 shows that this is indeed the case for the utterance *faded teal*, which is not seen in training, by visualizing its probability distribution along with those of *faded* and *teal* individually. The meaning of *faded teal* is *intersective*: it describes those colors that

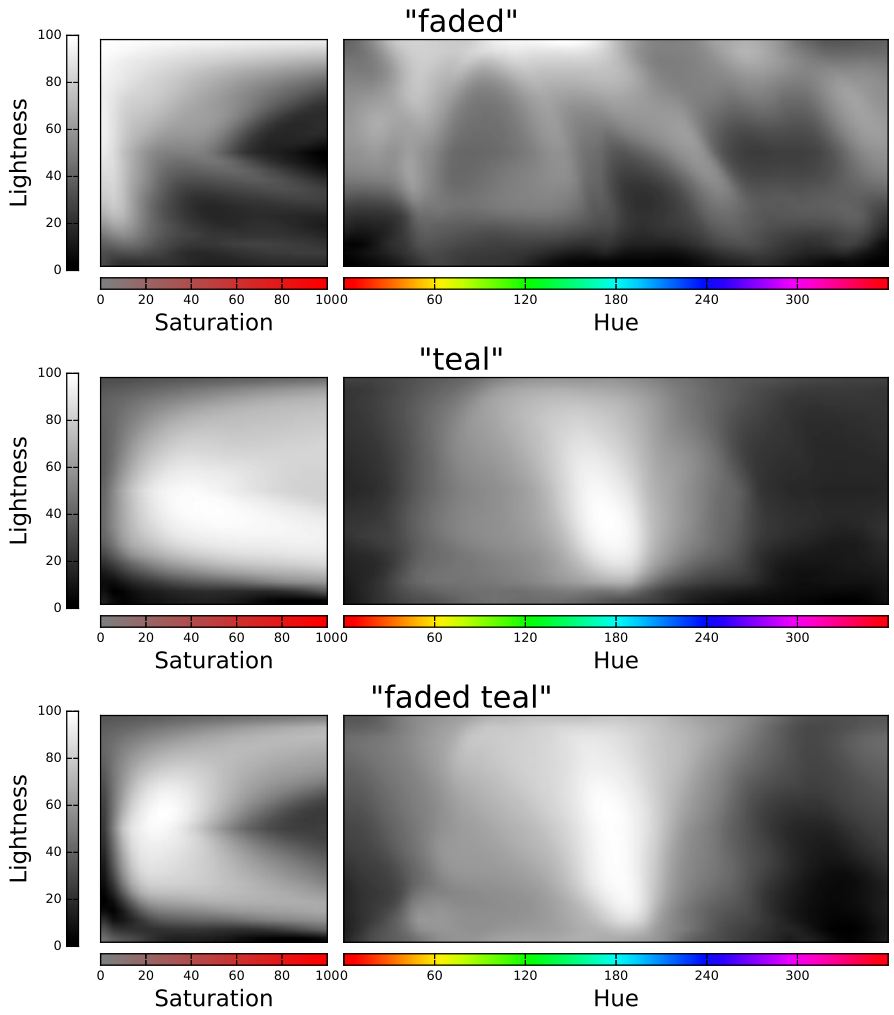


Figure 3.3: Conditional likelihood of *faded*, *teal*, and *faded teal*. The two meaning components can be seen in the two cross-sections: *faded* denotes a low saturation value, and *teal* denotes hues near the center of the spectrum.

are both *faded* and *teal*. *Faded* colors are lower in saturation, excluding the colors of the rainbow (the V on the right side of the left panel); and *teal* denotes colors with a hue near 180° (center of the right panel). Our model successfully represents these denotations, and its distribution for *faded teal* incorporates both the exclusion of highly saturated colors and the constraint on the hue.

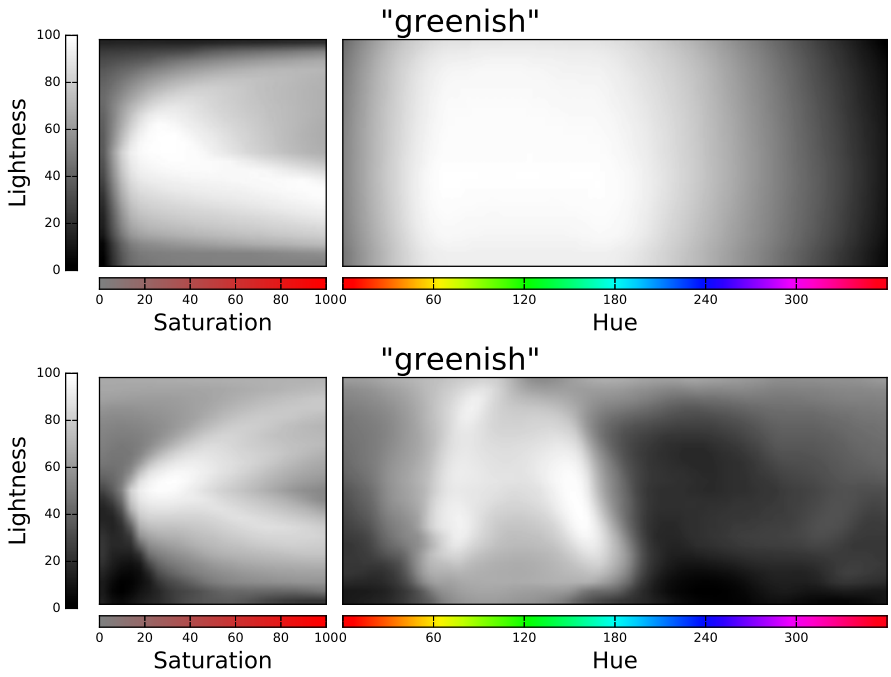


Figure 3.4: Conditional likelihood of *greenish* as a function of color. The distribution is bimodal, including greenish yellows and blues but not true greens. Top: LUX; bottom: our model.

3.5.3 Non-convex denotations

A final strength of neural models in general is their ability to model arbitrary continuous functions. This property, along with the Fourier feature transformation, allows our model to capture a rich set of denotations. In particular, our model addresses the shortcoming identified by McMahan and Stone (2015) that their model cannot capture *non-convex* denotations (sets in color space that surround one or more regions excluded from the set). The description *greenish* (Figure 3.4) has such a denotation: *greenish* is used to describe a region of color space surrounding, but not including, true greens.[2] Our model correctly produces *greenish* when the color is greenish blue and greenish yellow, but not when it is a pure green.

[2]Arguably, *greenish* does not truly have a non-convex denotation, but the exclusion of true greens is rather a pragmatic implicature arising from the availability of alternate descriptions such as *green*/*true green*. However, lacking an explicit account of alternatives, the RNN must accommodate the non-convex input region regardless of whether it results from semantic or pragmatic factors.

Color	Top-1	Sample
(36, 86, 63)	*orange*	*ugly*
(177, 85, 26)	*teal*	*robin's*
(29, 45, 71)	*tan*	*reddish green*
(196, 27, 71)	*grey*	*baby royal*

Table 3.3: Error analysis: some color descriptions sampled from our model that are incorrect or incomplete.

3.5.4 Error analysis

Table 3.3 shows some examples of errors found in samples taken from the model. The main type of error the system makes is ungrammatical descriptions, particularly fragments lacking a basic color term (e.g., *robin's*). Rarer are grammatical but meaningless compositions (*reddish green*) and false descriptions. When queried for its single most likely prediction, $\arg\max_u S(u \mid t)$, the result is nearly always an acceptable, "safe" description—manual inspection of 200 such top-1 predictions did not identify any errors.

3.6 Discussion

This chapter presented a model for generating compositional color descriptions that is capable of producing novel descriptions not seen in training and significantly outperforms prior work at conditional language modeling.[3]

One natural extension is the use of character-level sequence modeling to capture complex morphology (e.g., *-ish* in *greenish*). Kawakami et al. (2016) build character-level models for predicting colors given descriptions in addition to describing colors. Their model uses a *Lab*-space color representation and uses the color to initialize the LSTM instead of feeding it in at each time step; they also focus on visualizing point predictions of their description-to-color model, whereas we examine the full distributions implied by our color-to-description model.

[3]Code for this model is available at `https://github.com/stanfordnlp/color-describer`.

This chapter focused on generating descriptions of single colors in isolation. In the next chapter, we return to looking at context and the RSA approach by transitioning to a task involving descriptions of colors situated among other colors. This setting will demonstrate the power of the general, shared representation that is produced by neural sequence models. We will also investigate the listener role of this task in addition to the speaker role, and find that RSA gives us an effective way of leveraging improvements in modeling one role to improve the other.

Chapter 4

Colors in context

This chapter presents a scalable, learned model of pragmatic language understanding. The model is built around a version of the RSA model in which the literal semantic agents are RNNs that produce and interpret color descriptions in context. Like in Chapter 3, these models are learned from data and scale easily to large datasets containing diverse utterances. The RSA recursion is then defined in terms of these base agents: a pragmatic (S_1) speaker produces utterances based on a literal RNN listener using a sampling technique introduced by Andreas and Klein (2016), and a pragmatic (L_2) listener interprets utterances based on the pragmatic speaker's behavior.

Unlike the previous two chapters, this chapter focuses on a listener task (i.e., language understanding rather than generation). However, our most successful model integrates speaker and listener perspectives, combining predictions made by a system trained to understand color descriptions and one trained to produce them.

We evaluate this model with a new corpus of reference games in which the referents are patches of color. From the speaker perspective, this transforms the task described in Chapter 3 from an isolated setting to a fundamentally situated one. Table 4.1 illustrates the situated nature of color description understanding in a reference game with utterances from our corpus. In context 1, the comparative *darker* implicitly refers to both the target (boxed) and one of the other colors. In contexts 2 and 3, the target color is the same, but the distractors led the speaker to choose different basic color terms. In context 4, *blue* is a pragmatic choice even though two colors are shades

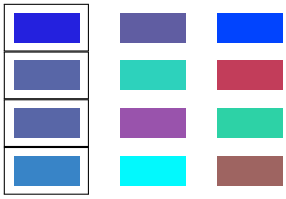
	Context	Utterance
1.		darker blue
2.		Purple
3.		blue
4.		blue

Table 4.1: Examples of color reference in context, taken from our corpus. The target color is boxed. The speaker's description is shaped not only by this target, but also by the other context colors and their relationships.

of blue, because the interlocutors assume about each other that they find the target color a more prototypical representative of blue and would prefer other descriptions (*teal*, *cyan*) for the middle color. The fact that *blue* appears in three of these four cases highlights the flexibility and context dependence of color descriptions.

Our task is fundamentally the same as that of Baumgaertner et al. (2012), but the corpus we release is larger by several orders of magnitude, consisting of 948 complete games with 53,365 utterances produced by human participants paired into dyads on the web. The linguistic behavior of the players exhibits many of the intricacies of language in general, including not just the context dependence and cognitive complexity discussed above, but also the compositionality and vagueness characteristic of the color descriptions discussed in Chapter 3. Unlike previous datasets featuring descriptions of individual colors, including the dataset used in the previous chapter (Munroe, 2010) and others (Cook et al., 2005; Kawakami et al., 2016), our new corpus situates colors in a communicative context. Adding context elicits greater variety in language use, including negations, comparatives, superlatives, metaphor, and shared associations.

Experiments on the data in our corpus show that this combined pragmatic model improves accuracy in interpreting human-produced descriptions over the basic RNN listener alone. We find that the largest improvement over the single RNN comes from blending it with an RNN trained to perform the speaker task, despite the fact that a model based only on this speaker RNN performs poorly on its own. Pragmatic reasoning on top of the listener RNN alone also yields improvements, which moreover

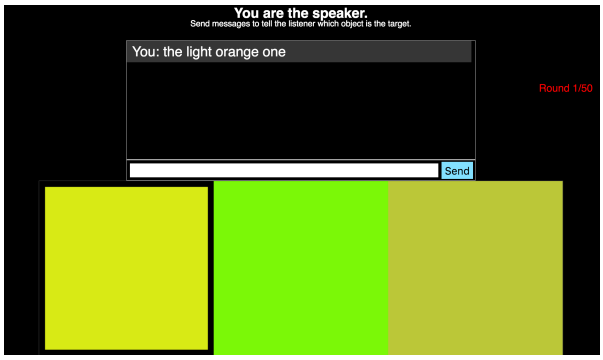


Figure 4.1: Example trial in corpus collection task, from speaker's perspective. The target color (boxed) was presented among two distractors on a neutral background.

come primarily in the hardest cases: 1) contexts with colors that are very similar, thus requiring the interpretation of descriptions that convey fine distinctions; and 2) target colors that most referring expressions fail to identify, whether due to a lack of adequate descriptive terms or a consistent bias against the color in the RNN listener.

4.1 Task and data collection

We evaluate our agents on a task of language understanding in a reference game as described in Section 1.2. To obtain a corpus of natural color reference data across varying contexts, we recruited 967 unique participants from Amazon Mechanical Turk to play 1,059 games of 50 rounds each, using the open-source framework of Hawkins (2015). Participants were sorted into dyads, randomly assigned the role of speaker or listener, and placed in a game environment containing a chat box and an array of three color patches (Figure 4.1). On each round, one of the three colors was chosen to be the target and highlighted for the speaker. They were instructed to communicate this information to the listener, who could then click on one of the colors to advance to the next trial. Both participants were free to use the chat box at any point.

To ensure a range of difficulty, we randomly interspersed an equal number of trials from three different conditions: 1) *close*, where colors were all within a distance of

θ from one another but still perceptible,[1] 2) *split*, where one distractor was within a distance of θ of the target, but the other distractor was farther than θ, and 3) *far*, where all colors were farther than θ from one another. Colors were rejection sampled uniformly from RGB (red, green, blue) space to meet these constraints.

After excluding extremely long messages,[2] incomplete games, and games whose participants self-reported confusion about the instructions or non-native English proficiency, we were left with a corpus of 53,365 speaker utterances across 46,994 rounds in 948 games. The three conditions are equally represented, with 15,519 *close* trials, 15,693 *split* trials, and 15,782 *far* trials. Participants were allowed to play more than once, but the modal number of games played per participant was one (75%). The modal number of messages sent per round was also one (90%). We release the filtered corpus we used throughout our analyses alongside the raw, pre-filter data collected from these experiments (see Footnote 11).

4.2 Human data analysis

Our corpus was developed not only to facilitate the development of models for grounded language understanding, but also to provide a richer picture of human pragmatic communication. The collection effort was thus structured like a large-scale behavioral experiment, closely following experimental designs like those of Clark and Wilkes-Gibbs (1986). This paves the way to assessing our model not solely based on the listener's classification accuracy, but also in terms of how qualitative features of the speaker's production compare to that of our human participants. Thus, the current section briefly reviews some novel findings from the human corpus that we use to inform our model assessment.

[1] We used the most recent CIEDE standard to measure color differences, which is calibrated to human vision (Sharma et al., 2005). All distances were constrained to be larger than a lower bound of $\epsilon = 5$ to ensure perceptible differences, and we used a threshold value of $\theta = 20$ to create conditions.

[2] Specifically, we set a length criterion at 4σ of the mean number of words per message (about 14 words, in our case), excluding 627 utterances. These often included meta-commentary about the game rather than color terms.

	human			S_0			S_1		
	far	split	close	far	split	close	far	split	close
# Chars	7.8	12.3	14.9	9.0	12.8	16.6	9.0	12.8	16.4
# Words	1.7	2.7	3.3	2.0	2.8	3.7	2.0	2.8	3.7
% Comparatives	1.7	14.2	12.8	3.6	8.8	13.1	4.2	9.0	13.7
% High Specificity	7.0	7.6	7.4	6.4	8.4	7.6	6.8	7.9	7.5
% Negatives	2.8	10.0	12.9	4.8	8.9	13.3	4.4	8.5	14.1
% Superlatives	2.2	6.1	16.7	4.7	9.7	17.2	4.8	10.3	16.6

Table 4.2: Corpus statistics and statistics of samples from artificial speakers (rates per utterance). S_0: RNN speaker; S_1: pragmatic speaker derived from RNN listener (see Section 4.3.3). The human and artificial speakers show many of the same correlations between language use and context type.

4.2.1 Listener behavior

Since color reference is a difficult task even for humans, we compared listener accuracy across conditions to calibrate our expectations about model performance. While participants' accuracy was close to ceiling (97%) on the *far* condition, they made significantly more errors on the *split* (90%) and *close* (83%) conditions (see Figure 4.3).

4.2.2 Speaker behavior

For ease of comparison to computational results, we focus on five metrics capturing different aspects of pragmatic behavior displayed by both human and artificial speakers in our task (Table 4.2). In all cases, we report test statistics from a mixed-effects regression including condition as a fixed effect and game ID as a random effect; except where noted, all test statistics reported correspond to p-values $< 10^{-4}$ and have been omitted for readability.

Words and characters

We expect human speakers to be more verbose in *split* and *close* contexts than *far* contexts; the shortest, simplest color terms for the target may also apply to one or both distractors, thus incentivizing the speaker to use more lengthy descriptions to fully distinguish it. Indeed, even if they *know* enough simple color terms to distinguish

all the colors lexically, they might be unsure their listeners will and so resort to modifiers anyway. To assess this hypothesis, we counted the average number of words and characters per message. Compared to the baseline *far* context, participants used significantly more words both in the *split* context ($t = 45.85$) and the *close* context ($t = 73.06$). Similar results hold for the character metric.

Comparatives and superlatives

As noted at the beginning of this chapter, comparative morphology implicitly encodes a dependence on the context; a speaker who refers to the target color as *the darker blue* is presupposing that there is another (lighter) blue in the context. Similarly, superlatives like *the bluest one* or *the lightest one* presuppose that all the colors can be compared along a specific semantic dimension. We thus expect to see this morphology more often where two or more of the colors are comparable in this way. To test this, we used the Stanford CoreNLP part-of-speech tagger (Toutanova et al., 2003) to mark the presence or absence of comparatives (JJR or RBR) and superlatives (JJS or RBS) for each message.

We found two related patterns across conditions. First, participants were significantly more likely to use both comparatives ($z = 37.39$) and superlatives ($z = 31.32$) when one or more distractors were close to the target. Second, we found evidence of an asymmetry in the use of these constructions across the *split* and *close* contexts. Comparatives were used significantly more often in the *split* context ($z = 4.4$), where only one distractor was close to the target, while superlatives were much more likely to be used in the *close* condition ($z = 32.72$).[3]

Negatives

In our referential contexts, negation is likely to play a role similar to that of comparatives: a phrase like *not the red or blue one* singles out the third color, and *blue but*

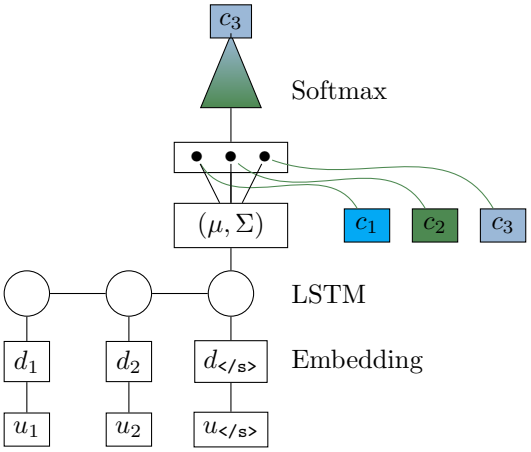
[3]We used Helmert coding to test these specific patterns: the first regression coefficient compares the 'far' condition to the mean of the other two conditions, and the second regression coefficient compares the 'split' condition to the 'close' condition.

not bright blue achieves a more nuanced kind of comparison. Thus, as with comparatives, we expect negation to be more likely where one or more distractors are close to the target. To test this, we counted occurrences of the string 'not' (by far the most frequent negation in the corpus). Compared to the baseline *far* context, we found that participants were more likely to use negative constructions when one ($z = 27.36$) or both ($z = 34.32$) distractors were close to the target.

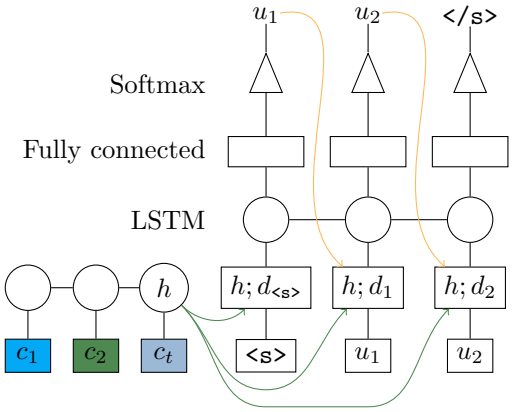
WordNet specificity

We expect speakers to prefer basic color terms wherever they suffice to achieve the communicative goal, since such terms are most likely to succeed with the widest range of listeners. Thus, a speaker might choose *blue* even for a clear periwinkle color. However, as the colors get closer together, the basic terms become too ambiguous, and thus the risk of specific terms becomes worthwhile (though lengthy descriptions might be a safer strategy, as discussed above). To evaluate this idea, we use WordNet (Fellbaum, 1998) to derive a specificity hierarchy for color terms, and we hypothesized that *split* or *close* conditions will tend to lead speakers to go lower in this hierarchy.

For each message, we transformed adjectives into derivationally-related noun forms (e.g. 'reddish' $\rightarrow$ 'red'), filtered to include only nouns with 'color' in their hypernym paths, calculated the depth of the hypernym path of each color word, and took the maximum depth occurring in a message. For instance, the message "deep magenta, purple with some pink" received a score of 9. It has three color terms: "purple" and "pink," which have the basic-level depth of 7, and "magenta," which is a highly specific color term with a depth of 9. Finally, because there weren't meaningful differences between words at depths of 8 ("rose", "teal") and 9 ("tan," "taupe"), we conducted our analyses on a binary variable thresholded to distinguish "high specificity" messages with a depth greater than 7. We found a small but reliable increase in the likelihood of "high specificity" messages from human speakers in the *split* ($z = 2.84, p = 0.005$) and *close* ($z = 2.33, p = 0.02$) contexts, compared to the baseline *far* context.



(a) The L_0 agent processes tokens u_i of a color description u sequentially. The final representation is transformed into a Gaussian distribution in color space, which is used to score the context colors $c_1 \dots c_3$.



(b) The S_0 agent processes the target color c_t in context and produces tokens u_i of a color description sequentially. Each step in production is conditioned by the context representation h and the previous word produced.

Figure 4.2: The neural base speaker and listener agents.

4.3 Models

4.3.1 Base listener

Our base listener agent L_0 (Figure 4.2a) is an LSTM encoder model that predicts a Gaussian distribution over colors in a transformed representation space. The input words are embedded in a 100-dimensional vector space. Word embeddings are initialized to random normally-distributed vectors ($\mu = 0$, $\sigma = 0.01$) and trained. The sequence of word vectors is used as input to an LSTM with 100-dimensional hidden state, and a linear transformation is applied to the output representation to produce the parameters μ and Σ of a quadratic form[4]

$$\text{score}(f) = -(f - \mu)^T \Sigma (f - \mu)$$

[4]The quadratic form is not guaranteed to be negative definite and thus define a Gaussian; however, it is for $> 95\%$ of inputs. The distribution over the finite set of context colors is well-defined regardless.

where f is a vector representation of a color. Each color is represented in its simplest form as a three-dimensional vector in RGB space. These RGB vectors are then Fourier-transformed as described in Section 3.3 to obtain the representation f.

The values of score(f) for each of the n context colors are normalized in log space to produce a probability distribution over the context colors. We denote this distribution by $L_0(t \mid u, C; \theta)$, where θ represents the vector of parameters that define the trained model.

4.3.2 Base speaker

We also employ an RNN-based speaker model $S_0(u \mid t, C; \phi)$. This speaker serves two purposes: 1) it is used to define a pragmatic listener akin to l_1 in (1.4), and 2) it provides samples of alternative utterances for each context, to avoid enumerating the intractably large space of possible utterances.

The speaker model consists of an LSTM context encoder and an LSTM description decoder (Figure 4.2b). In this model, the colors of the context $c_i \in C$ are transformed into Fourier representation space, and the sequence of color representations is passed through an LSTM with 100-dimensional hidden state. The context is reordered to place the target color last, minimizing the length of dependence between the most important input color and the output (Sutskever et al., 2014) and eliminating the need to represent the index of the target separately. The final cell state of this recurrent neural network is concatenated with a 100-dimensional embedding for the previous token output at each step of decoding. The resulting vector is input along with the previous cell state to the LSTM cell. The remainder of the model is identical to the output layers of the RNN speaker model in Section 3.3.

4.3.3 Pragmatic agents

Using the above base agents, we define a pragmatic speaker S_1 and a pragmatic listener L_2:

$$S_1(u \mid t, C; \theta) = \frac{L_0(t \mid u, C; \theta)^\alpha}{\sum_{u'} L_0(t \mid u', C; \theta)^\alpha} \tag{4.1}$$

$$L_2(t \mid u, C; \theta) = \frac{S_1(u \mid t, C; \theta)}{\sum_{t'} S_1(u \mid t', C; \theta)} \tag{4.2}$$

These definitions mirror those in (1.7) and (1.8) above, with $\mathcal{L}$ replaced by the learned weights θ.

Just as in (1.7), the denominator in (4.1) should consist of a sum over the entire set of potential utterances, which is exponentially large in the maximum utterance length and might not even be finite. As mentioned in Section 4.3.2, we limit this search by taking m samples from $S_0(u \mid i, C; \phi)$ for each target index i, adding the actual utterance from the testing example, and taking the resulting multiset as the universe of possible utterances, weighted towards frequently-sampled utterances.[5] Taking a number of samples from S_0 for each referent in the context gives the pragmatic listener a variety of informative alternative utterances to consider when interpreting the true input description. We have found that m can be small; in our experiments, it is set to 8.

To reduce the noise resulting from the stochastically chosen alternative utterance sets, we also perform this alternative-set sampling n times and average the resulting probabilities in the final L_2 output. We again choose $n = 8$ as a satisfactory compromise between effectiveness and computation time.

[5] An alternative would be to enforce uniqueness within the alternative set, keeping it a true set as in the basic RSA formulation; this could be done with rejection sampling or beam search for the highest-scoring speaker utterances. We found that doing so with rejection sampling hurt model performance somewhat, so we did not pursue the more complex beam search approach.

Blending with a speaker-based agent

A second pragmatic listener L_1 can be formed in a similar way, analogous to l_1 in (1.4):

$$L_1(t \mid u, C; \phi) = \frac{S_0(u \mid t, C; \phi)}{\sum_{t'} S_0(u \mid t', C; \phi)} \tag{4.3}$$

We expect L_1 to be less accurate than L_0 or L_2, because it is performing a listener task using only the outputs of a model trained for a speaker task. However, this difference in training objective can also give the model strengths that complement those of the two listener-based agents. One might also expect a realistic model of human language interpretation to lie somewhere between the "reflex" interpretations of the neural base listener and the "reasoned" interpretations of one of the pragmatic models. This has an intuitive justification in people's uncertainty about whether their interlocutors are speaking pragmatically: "should I read more into that statement, or take it at face value?" We therefore also evaluate models defined as a weighted average of L_0 and each of L_1 and L_2, as well as an "ensemble" model that combines all of these agents. Specifically, we consider the following blends of neural base models and pragmatic models, with $\mathbf{L}_i$ abbreviating $L_i(t \mid u, C; \theta, \phi)$ for convenience:

$$\mathbf{L}_a \propto {\mathbf{L}_0}^{\beta_a} \cdot \mathbf{L}_1^{1-\beta_a} \tag{4.4}$$

$$\mathbf{L}_b \propto {\mathbf{L}_0}^{\beta_b} \cdot \mathbf{L}_2^{1-\beta_b} \tag{4.5}$$

$$\mathbf{L}_e \propto {\mathbf{L}_a}^{\gamma} \cdot \mathbf{L}_b^{1-\gamma} \tag{4.6}$$

The hyperparameters in the exponents allow tuning the blend of each pair of models—e.g., overriding the neural model with the pragmatic reasoning in L_b. The value of the weights β_a, β_b, and γ can be any real number; however, we find that good values of these weights lie in the range $[-1, 1]$. As an example, setting $\beta_b = 0$ makes the blended model L_b equivalent to the pragmatic model L_2; $\beta_b = 1$ ignores the pragmatic reasoning and uses the base model L_0's outputs; and $\beta_b = -1$ "subtracts" the base model from the pragmatic model (in log probability space) to yield a "hyperpragmatic" model.

4.3.4 Training

We split our corpus into approximately equal train/dev/test sets (15,665 train trials, 15,670 dev, 15,659 test), ensuring that trials from the same dyad are present in only one split. We preprocess the data by 1) lowercasing; 2) tokenizing by splitting off punctuation as well as the endings *-er*, *-est*, and *-ish*;[6] and 3) replacing tokens that appear once or not at all in the training split[7] with `<unk>`. We also remove listener utterances and concatenate speaker utterances on the same context. We leave handling of interactive dialogue to future work (Section 4.7).

We use ADADELTA (Zeiler, 2012) and Adam (Kingma and Ba, 2014), adaptive variants of stochastic gradient descent (SGD), to train listener and speaker models. The choice of optimization algorithm and learning rate for each model were tuned with grid search on a held-out tuning set consisting of 3,500 contexts.[8] We also use a fine-grained grid search on this tuning set to determine the values of the pragmatic reasoning parameters α, β, and γ. In our final ensemble L_e, we use $\alpha = 0.544$, base weights $\beta_a = 0.492$ and $\beta_b = -0.15$, and a final blending weight $\gamma = 0.491$. It is noteworthy that the optimal value of β_b from grid search is *negative*. The effect of this is to amplify the difference between L_0 and L_2: the listener-based pragmatic model, evidently, is not quite pragmatic enough.

4.4 Model results

4.4.1 Speaker behavior

To compare human behavior with the behavior of our embedded speaker models, we performed the same behavorial analysis done in Section 4.2.2. Results from this analysis are included alongside the human results in Table 4.2. Our pragmatic speaker model S_1 did not differ qualitatively from our base speaker S_0 on any of the metrics,

[6] We only apply this heuristic ending segmentation for the listener; the speaker is trained to produce words with these endings unsegmented, to avoid segmentation inconsistencies when passing speaker samples as alternative utterances to the listener.

[7] 1.13% of training tokens, 1.99% of dev/test.

[8] For L_0: ADADELTA, learning rate $\eta = 0.2$; for S_0: Adam, learning rate $\alpha = 0.004$.

so we only summarize results for humans and the pragmatic model.

Words and characters

We found human speakers to be more verbose when colors were closer together, in both number of words and number of characters. As Table 4.2 shows, our S_1 agent shows the same increase in utterance length in the *split* ($t = 18.07$) and *close* ($t = 35.77$) contexts compared to the *far* contexts.

Comparatives and superlatives

Humans used more comparatives and superlatives when colors were closer together; however, comparatives were preferred in the *split* contexts, superlatives in the *close* contexts. Our pragmatic speaker shows the first of these two patterns, producing more comparatives ($z = 14.45$) and superlatives ($z = 16$) in the *split* or *close* conditions than in the baseline *far* condition. It does not, however, capture the peak in comparative use in the *split* condition. This suggests that our model is simulating the human strategy at some level, but that more subtle patterns require further attention.

Negations

Humans used more negations when the colors were closer together. Our pragmatic speaker's use of negation shows the same relationship to the context ($z = 8.55$ and $z = 16.61$, respectively).

WordNet specificity

Humans used more "high specificity" words (by WordNet hypernymy depth) when the colors were closer together. Our pragmatic speaker showed a similar effect ($z = 2.65, p = 0.008$ and $z = 2.1, p = 0.036$, respectively).

model	accuracy (%)	perplexity
L_0	83.30	1.73
$L_1 = L(S_0)$	80.51	1.59
$L_2 = L(S(L_0))$	83.95	1.51
$L_a = L_0 \cdot L_1$	84.72	1.47
$L_b = L_0 \cdot L_2$	83.98	1.50
$L_e = L_a \cdot L_b$	**84.84**	**1.45**
human	*90.40*	
L_0	85.08	1.62
L_e	**86.98**	**1.39**
human	*91.08*	

Table 4.3: Accuracy and perplexity of the base and pragmatic listeners and various blends (weighted averages, denoted $A \cdot B$). Top: dev set; bottom: test set.

4.4.2 Listener accuracy

Table 4.3 shows the accuracy and perplexity of the base listener L_0, the pragmatic listeners L_1 and L_2, and the blended models L_a, L_b, and L_e at resolving the human-written color references. Accuracy differences are significant[9] for all pairs except L_2/L_b and L_a/L_e. As we expected, the speaker-based L_1 alone performs the worst of all the models. However, blending it with L_0 doesn't drag down L_0's performance but rather produces a considerable improvement compared to both of the original models, consistent with our expectation that the listener-based and speaker-based models have complementary strengths.

We observe that L_2 significantly outperforms its own base model L_0, showing that pragmatic reasoning on its own contributes positively. Blending the pragmatic models with the base listener also improves over both individually, although not significantly in the case of L_b over L_2. Finally, the most effective listener combines both pragmatic models with the base listener. Plotting the number of examples changed by condition on the dev set (Figure 4.3) reveals that the primary gain from including the pragmatic models is in the *close* and *split* conditions, when the model has to distinguish highly

[9] $p < 0.012$, approximate permutation test (Padó, 2006) with Bonferroni correction, 10,000 samples.

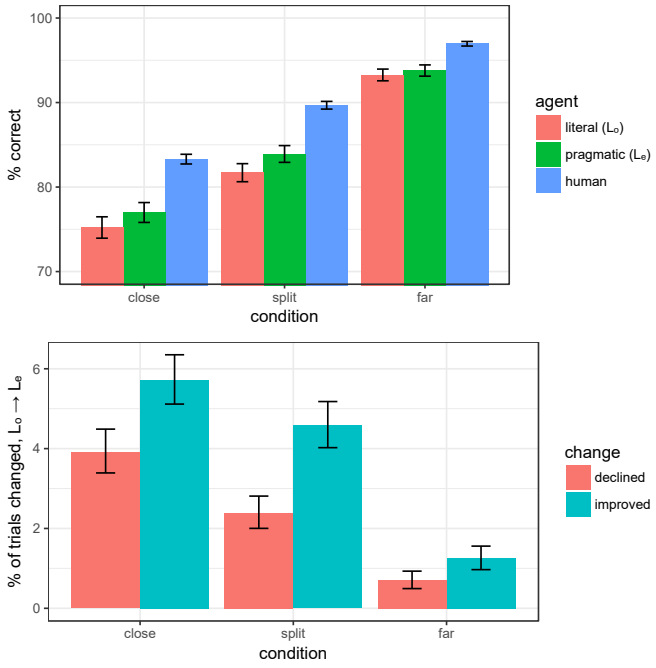


Figure 4.3: Human and model reference game performance (top) and fraction of examples improved and declined from L_0 to L_e (bottom) on the dev set, by condition.

similar colors and often cannot rely only on basic color terms. On the test set, the final ensemble improves significantly[10] over the base model on both metrics.

4.5 Model analysis

Examining the full probability tables for various dev set examples offers insight into the value of each model in isolation and how they complement each other when blended together. In particular, we see that the listener-based (L_2) and speaker-based (L_1) pragmatic listeners each overcome a different kind of "blind spot" in the neural base listener's understanding ability.

First, we inspect examples in which L_2 is superior to L_0. In most of these examples, the alternative utterances sampled from S_0 for one of the referents i fail to identify their intended referent to L_0. The pragmatic listener interprets this to mean that

[10] $p < 0.001$, approximate permutation test, 10,000 samples.

L_0			
blue	9	**91**	<1
true blue	11	**89**	<1
light blue	<1	**>99**	<1
brightest	<1	**>99**	<1
bright blue	<1	**>99**	<1
red	<1	1	**99**
purple	<1	2	**98**
S_1			
blue	41	19	<1
true blue	**47**	19	<1
light blue	5	**20**	<1
brightest	<1	**20**	<1
bright blue	2	**20**	<1
red	1	2	**50**
purple	5	1	**50**
L_2			
blue	**68**	32	<1
S_0	*5.71*	***7.63***	*0.01*
L_1	43	**57**	<1
L_a	**50**	**50**	<1
L_b	**68**	32	<1
L_e	**59**	41	<1

L_0			
drab green not the bluer one	<1	<1	**>99**
gray	**96**	4	<1
blue dull green	24	**76**	<1
blue	<1	**>99**	<1
bluish	<1	**>99**	<1
green	4	1	**95**
yellow	<1	<1	**>99**
S_1			
drab green not the bluer one	1	<1	**34**
gray	**58**	5	<1
blue dull green	27	28	<1
blue	2	**32**	<1
bluish	1	**32**	<1
green	10	3	33
yellow	<1	<1	**34**
L_2			
drab green not the bluer one	5	<1	**95**
S_0 ($\times 10^{-9}$)	***5.85***	*0.38*	*<0.01*
L_1	**94**	6	<1
L_a	**92**	6	2
L_b	8	1	**91**
L_e	**63**	6	32

Figure 4.4: Conditional probabilities (%) of all agents for two dev set examples. The target color is boxed, and the human utterances (*blue*, *drab green not the bluer one*) are **bolded**. Boxed cells for alternative utterances indicate the intended target; largest probabilities are in **bold**. S_0 probabilities (*italics*) are normalized across all utterances. Sample sizes are reduced to save space; here, $m = 2$ and $n = 1$ (see Section 4.3.3).

referent i is inherently difficult to refer to, and it compensates by increasing referent i's probability.

This is beneficial when i is the true target. The left column of Figure 4.4 shows one such example: a context consisting of a somewhat prototypical blue, a bright cyan, and a purple-tinged brown, with the utterance *blue*. The base listener interprets this as referring to the cyan with 91% probability, perhaps due to the extreme saturation of the cyan maximally activating certain parts of the neural network. However, when the pragmatic model takes samples from S_0 to probe the space of alternative utterances, it becomes apparent that indicating the more ordinary blue to the listener is difficult: for the utterances chosen by S_0 intending this referent (*true blue*, *light blue*), the listener also chooses the cyan with >89% confidence.

Pragmatic reasoning overcomes this difficulty. Only two utterances in the alternative set (the actual utterance *blue* and the sampled alternative *true blue*) result in any appreciable probability mass on the true target, so the pragmatic listener's model of the speaker predicts that the speaker would usually choose one of these two utterances for the prototypical blue. However, if the target were the cyan, the speaker would have many good options. Therefore, the fact that the speaker chose *blue* is interpreted as evidence for the true target. This mirrors the back-and-forth reasoning behind the definition of conversational implicature (Grice, 1975).

This reasoning can be harmful when i is one of the distractors: the pragmatic listener is then in danger of overweighting the distractor and incorrectly choosing it. This is a likely reason for the small performance difference between L_0 and L_2. Still, the fact that L_2 is more accurate overall, in addition to the negative value of β_b discovered in grid search, suggests that the pragmatic reasoning provides value on its own.

However, the final performance improves greatly when we incorporate both listener-based and speaker-based agents. To explain this improvement, we examine examples in which both listener-based agents L_0 and L_2 give the wrong answer but are overridden by the speaker-based L_1 to produce the correct referent. The discrepancy between the two kinds of models in many of these examples can be explained by the fact that the speaker takes the context as input, while the listener does not. The

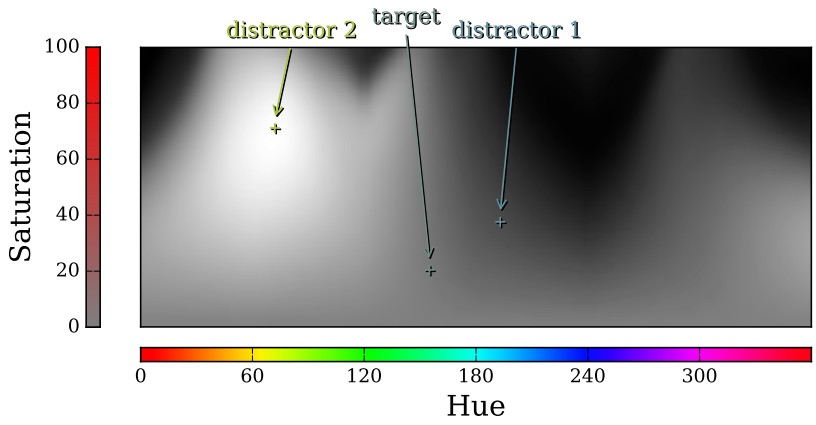


Figure 4.5: L_0's log marginal probability density, marginalizing over V (value) in HSV space, of color conditioned on the utterance *drab green not the bluer one*. White regions have higher probability. Labeled colors are the three colors from the right column of Figure 4.4.

listener is thus asked to predict a region of color space from the utterance *a priori*, while the speaker can take into account relationships between the context colors in scoring utterances.

The right column of Figure 4.4 shows an example of this. The context contains a grayish green (the target), a grayish blue-green ("distractor 1"), and a yellowish green ("distractor 2"). The utterance from the human speaker is *drab green not the bluer one*, presumably intending *drab* to exclude the brighter yellowish green. However, the L_0 listener must choose a region of color space to predict based on the utterance alone, without seeing the other context colors.

Figure 4.5 shows a visualization of the listener's prediction using the method described in Section 3.5. The input description mentions two properties of the target color: the color should be *drab* (low-saturation) and *green* (near 120 on the hue spectrum) but not *blue* (near 240 in hue). The utterance does not constrain the value (roughly, brightness–darkness) component, so here we sum over this component to summarize the 3-dimensional distribution in 2 dimensions.

The L_0 model correctly interprets both of these constraints: it gives higher probability to low-saturation colors and greens, while avoiding bluer colors. However, the result is a probability distribution nearly centered at distractor 2, the brighter green.

In fact, if we were not comparing it to the other colors in the context, distractor 2 would be a very good example of a drab green that is not bluish.

The speaker S_0, however, produces utterances conditioned on the context; it has successfully learned that *drab* would be more likely as a description of the grayish green than as a description of the yellowish one in this context. The speaker-based listener L_1 therefore predicts the true target, with greater confidence than L_0 or L_2. This prediction results in the blends L_a and L_e preferring the true target, allowing the speaker's perspective to override the listener's.

4.6 Related work

Meo et al. (2014) evaluate a model of color description generation (McMahan and Stone, 2015) on the color reference data of Baumgaertner et al. (2012) by creating an $L(S_0)$ listener. Their model requires enumerating the set of possible utterances for each context, which is infeasible when utterances are as varied as those in our dataset.

Andreas and Klein (2016) also combine neural speaker and listener models in a reference game setting. They propose a pragmatic speaker, $S(L_0)$, sampling from a neural S_0 model to limit the search space and regularize the model toward human-like utterances. We show these techniques help in listener (understanding) tasks as well. Approaching pragmatics from the listener side requires either inverting the pragmatic reasoning (i.e., deriving a listener from a speaker), or adding another step of recursive reasoning, yielding a two-level derived pragmatic model $L(S(L_0))$. We show both approaches contribute to an effective listener.

4.7 Discussion

This chapter presents a newly-collected corpus of color descriptions from reference games, and shows that a pragmatic reasoning agent incorporating neural listener and speaker models interprets color descriptions in context better than the listener alone.

An important distinction between the model described here and that in Chapter 2

is that here the pragmatic reasoning scheme is applied on top of pre-trained models. Since the human data is presumed to be pragmatic, we expect that training a model in an end-to-end fashion, the way it is done in Chapter 2, would be advantageous. The gradient of the log-likelihood of the L_2 model given an example pair and a set of alternative utterances is well-defined, but using it directly in SGD training is not as theoretically well motivated as it is for the L_0 model, since the bias introduced by the procedure of sampling a finite alternative set must be taken into account. Doing so would seem to require a more integrated solution to the problem of the intractable alternative utterance space.

The separation of referent and utterance representation in our base speaker and listener models in principle allows easy substitution of referents other than colors (for example, images), although the performance of the listener agents could be limited by the representation of utterance semantics as a Gaussian distribution in referent representation space. Our pragmatic agents also rely on the ability to enumerate the set of possible referents. Avoiding this enumeration, as would be necessary in tasks with intractably large referent spaces, is a challenging theoretical problem for RSA-like models.

Another important next step is to pursue multi-turn dialogue. As noted in Section 4.1, both participants in our reference game task could use the chat window at any point, and more than half of dyads had at least one two-way interaction. Chapter 6 discusses some of the challenges involved in modeling dialogue in more detail.

We have made the dataset described in Section 4.1 publicly available[11] with the expectation that others may find interest in these challenges as well.

[11] `https://cocolab.stanford.edu/datasets/colors.html`

Chapter 5

Generating bilingual pragmatic references

A feature of pragmatic reasoning that distinguishes it from syntax and semantics is that it has strong claims to cross-lingual generalization, in that it depends far less on arbitrary conventions of a particular language. However, the ways in which these contrasts are expressed depend heavily on language-specific syntax and semantics. This chapter is concerned with developing a model of contextual language production that captures language-specific syntax and semantics while also exhibiting responsiveness to contextual differences. It evaluates a sequence-to-sequence speaker agent based on that of Section 3.3 on the same color reference game described in Section 4.1 played in both English and Mandarin Chinese, using a new corpus of Chinese data collected using the same protocol.

While English and Chinese both use fairly similar syntax for color descriptions, the reference game setting is designed to elicit constructions that make reference to the context, and these constructions—particularly comparatives and negation—differ morpho-syntactically and pragmatically between the two languages. Additionally, Chinese is considered to have a smaller number of basic color terms (Berlin and Kay, 1969), which predicts that more specific descriptions will be interpreted as pragmatically marked.

Our primary goal in this chapter is to examine the effects of *bilingual* training:

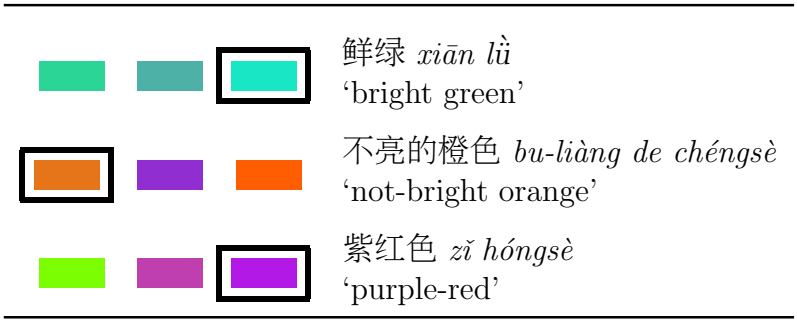


Figure 5.1: Reference game contexts and utterances from our Chinese corpus. The boxed color is the target. Some color terms show differences between Chinese and English, such as 绿 *lǜ* 'green' in the first example for a color that might be referred to with 'blue' or 'aqua' in English.

building one speaker trained on both English and Chinese data with a shared vocabulary, so that it can produce utterances in either language. The reference game setting offers an objective measure of success on the grounded language task, namely, the speaker's ability to guide the listener to the target. We use this to address the tricky problem of speaker evaluation in natural language generation. (The metrics used in Chapter 2 are less helpful here, since utterances are not pre-annotated with semantic attributes, and a speaker can do an adequate job communicating while hardly ever exactly duplicating a human reference utterance.) Specifically, we judge a speaker model on the accuracy of an RSA L_1 model built from it. We refer to this metric as *pragmatic informativeness* because it requires not only accuracy but also effectiveness at meeting the players' shared goal (Grice, 1975). A more formal definition and a discussion of alternatives are given in Section 5.3.2.

We show that a bilingually-trained model produces distributions over Chinese utterances that have higher pragmatic informativeness than a monolingual model. An analysis of the learned word embeddings reveals that the bilingual model learns color synonyms between the two languages without being directly exposed to labeled pairs. However, using a context-independent color term elicitation task from Berlin and Kay (1969) on our models, we show that the learned lexical meanings are largely faithful to each language's basic color system, with only minor cross-lingual influences. This suggests that the improvements due to adding English data are not primarily due

to better representations of the input colors or lexical semantics alone. The bilingual model does better resemble human patterns of utterance length as a function of contextual difficulty, suggesting the pragmatic level as one possible area of cross-lingual generalization.

5.1 Data collection

We adapted the open-source reference game framework of Hawkins (2015) to Chinese and followed the data collection protocols described in Chapter 4 as closely as possible, in the hope that this can be the first step in a broader multilingual color reference project. We again recruit pairs of players on Amazon Mechanical Turk in real time, randomly assigning one the role of the speaker and the other the listener in the reference game. Players are self-reported Chinese speakers, but they must pass a series of Chinese comprehension questions in order to proceed, with instructions in a format preventing copy-and-paste translation. After filtering out extremely long messages (number of tokens greater than 4σ above the mean), spam games,[1] and players who self-reported confusion about the game, we have a new corpus of 5,774 Chinese messages in color reference games, which we release publicly.

The contexts are again divided into *far* (1,421 contexts), *split* (1,412 contexts), and *close* (1,425 contexts) condition, using a threshold of $\theta = 20$ by the CIEDE2000 color-difference formula (Sharma et al., 2005).

5.2 Human data analysis

As we mentioned earlier, our main goal with this work is to investigate the effects of bilingual training on pragmatic language use. We first examine the similarities and differences in pragmatic behaviors between the English and Chinese corpora we use, using analyses similar to those in Section 4.2.2. The picture that emerges accords well with our expectations about pragmatics: the broad patterns are aligned across

[1]Some players found they could advance through rounds by sending duplicate messages. Games were considered spam if the game contained 25 or more duplicates.

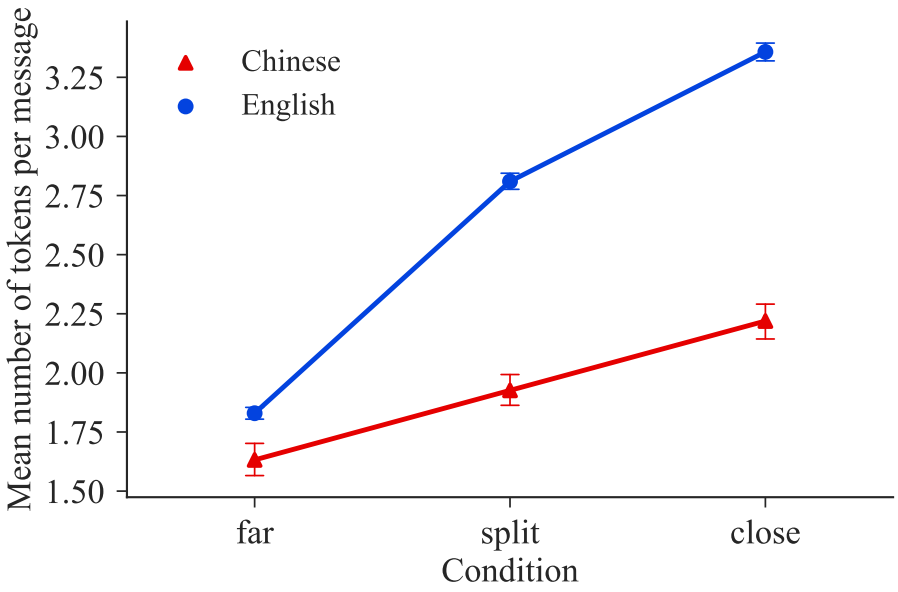


Figure 5.2: Comparison of mean length of messages in English and Chinese. The *split* and *close* conditions have more similar context colors (Section 5.1).

the two languages, with the observed differences mostly tracing to the details of their lexicons and morphosyntactic constructions.

5.2.1 Message length

Like in Section 4.2.2, we expect message length to correlate with the difficulty of the context: as the target becomes harder to distinguish from the distractors, the speaker will produce more complex messages, and length is a rough indicator of such complexity. For this analysis, we used the Natural Language Toolkit (NLTK; Bird et al. 2009) and Jieba (Junyi, 2015) to tokenize English and Chinese messages, respectively, and counted the number of tokens in both languages as a measure of message length. The results (Figure 5.2) confirm that in both languages, players become more verbose in more difficult conditions.[2]

[2]We do not believe that the overall drop in message length from English to Chinese reflects a fundamental difference between the languages; this has a few possible explanations, from Chinese messages taking the form of "sentence segments" (Wang and Qin, 2010) to differences in tokenization.

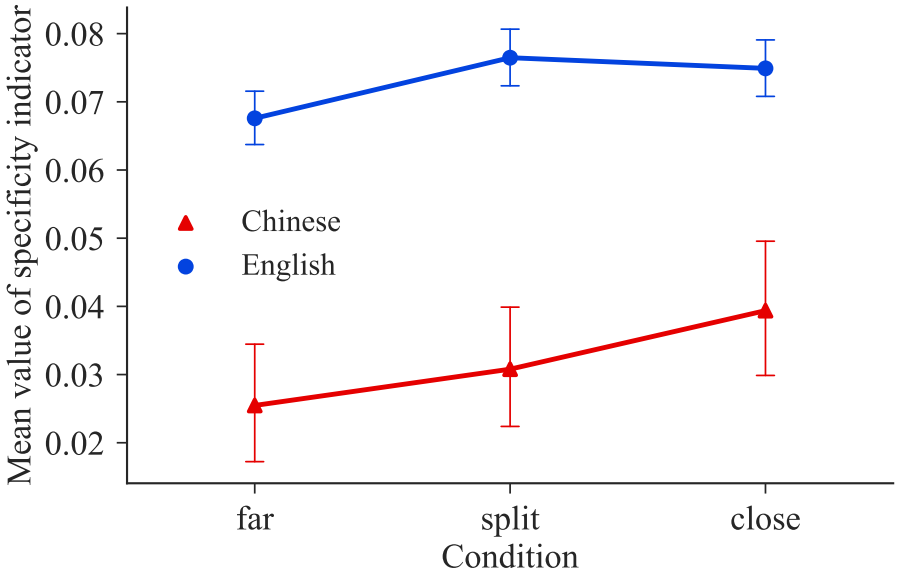


Figure 5.3: Comparison of WordNet specificity in Chinese and English.

5.2.2 Specificity

Section 4.2.2 also discussed the relationship between specificity and difficulty: in the *split* and *far* conditions, the speaker must make fine-grained distinctions. We repeat the analysis of WordNet specificity from that section on our Chinese data. We first translate to English via Google Translate, then evaluate the specificity of the translated word. It should be noted that this method has the drawback of obscuring differences between the two languages' color systems, as well as the potential for introducing noise due to errors in automatic translation. Though Mandarin variations of WordNet exist, we chose this translation method to standardize hypernym paths for both languages. Differences in ontology decisions between lexical resources prevent straightforward cross-lingual comparisons of hypernym depths, while automatic translation to a common language ensures the resulting hypernym paths are directly comparable.

Figure 5.3 summarizes the results of this measurement. In general, the usage of high-specificity color words increases in more difficult conditions, as expected. However, we see that Chinese speakers use them significantly less than English speakers.

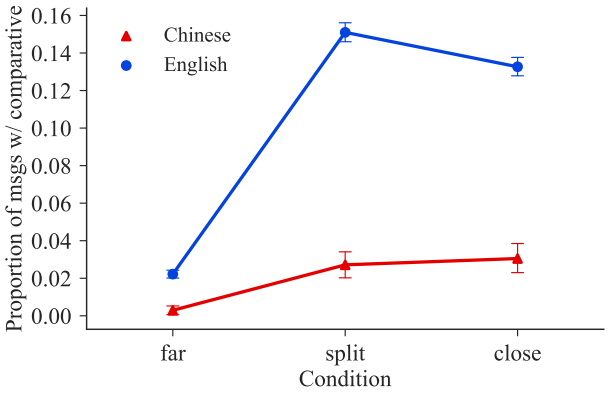
Instead, Chinese speakers use nominal modifiers, such as 草 *cǎo* 'grass' and 海 *hǎi* 'ocean', which do not contain "color" in their hypernym paths and are thus not marked as high-specificity. To quantify this observation, we annotated random samples of 200 messages from each language for whether they contained nominal color descriptions, and found that 3.5% of the English messages contain such nominals versus 13.5% of the Chinese messages.

The use of nominal modifiers as opposed to adjectives ('dark orange', 'dull brown') is arguably expected given the claims of Berlin and Kay (1969) and others that Chinese has fewer basic color terms than English, thus requiring more visually evocative modifiers to clarify distinctions between similar hues. (This isn't a complete explanation, since Chinese is rich in narrow but rare non-basic color terms. For the cases where Chinese has an appropriate narrow color term, it is possible that speakers make a pragmatic decision to avoid obscure vocabulary in favor of more familiar nouns.)

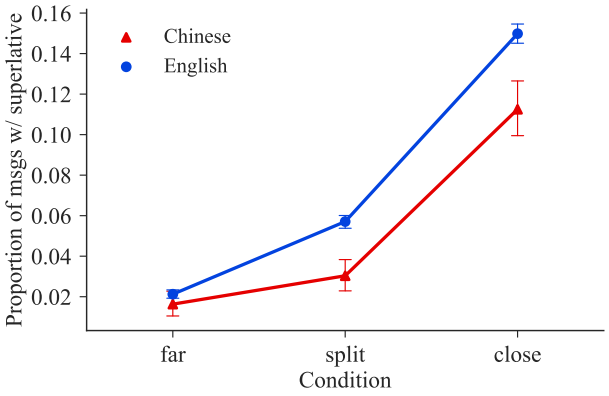
5.2.3 Comparatives, superlatives, and negation

To detect comparative and superlative adjectives in English, we use NLTK POS-tagging, which outputs JJR and RBR for comparatives, and JJS and RBS for superlatives. In Chinese, we look for the tokens 更 *gèng* 'more' and 比 *bǐ* 'comparatively' to detect comparatives and 最 *zuì* 'most' to detect superlatives. We detect negation by tokenizing messages with NLTK and Jieba and then looking for the tokens *not* and *n't* in English and corresponding 不 *bù* and 没 *méi* in Chinese.

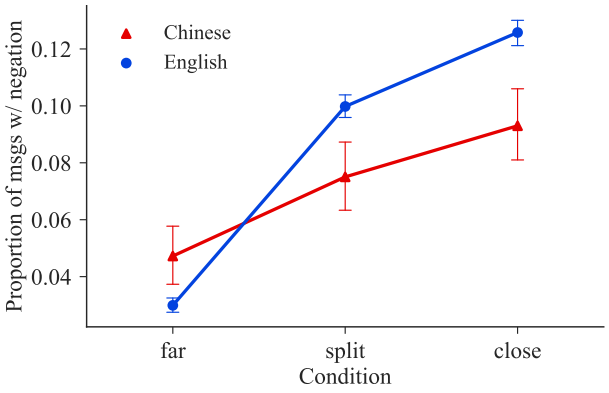
Both languages exhibit similar trends for superlative adjectives. In English, comparatives are used most frequently in the *split* condition and second most frequently in the *close* condition, while in Chinese, they occur at around the same rate in the *split* and *close* conditions. The literature is not conclusive about the source of these differences. Xia (2014) argues that complex attributives are rarely used and sound "syntactically deviant or Europeanized" (Zhu, 1982; Xie, 2001) in Chinese, citing the left-branching nature of the language as restricting attributives in length and complexity. There are also conflicting theories on the markedness of gradable adjectives in Chinese (Grano, 2012; Ito, 2008); such markedness may contribute to the frequency



(a) Usage of comparative adjectives in Chinese and English.



(b) Usage of superlative adjectives in Chinese and English.



(c) Usage of negation in Chinese and English.

Figure 5.4: Comparison of usage of comparatives, superlatives, and negation in English and Chinese.

at which comparative forms are used.

We also see that both languages follow the same general trend of using negation more frequently as the condition becomes more difficult.

5.3 Models and evaluation metrics

5.3.1 Monolingual and bilingual speaker models

We build and evaluate three artificial agents on this reference game task, two trained on monolingual descriptions (one for each language) and one on bilingual descriptions. We base these models on the basic speaker architecture from Chapter 4. The monolingual speakers represent the context by passing all the context colors as input to an LSTM sequence encoder, then concatenating this representation with a word vector for each previous output token as the input to an LSTM decoder that produces a color description token-by-token. This defines a distribution over descriptions u conditioned on the target and context, $S(u \mid c_t, C)$.

To accommodate bilingual training with this architecture, we expand the vocabulary to include English and Chinese words, and we add a flag ℓ to the input specifying whether the model's output should be in English ($\ell = 0$) or Chinese ($\ell = 1$):

$$S(u \mid \ell, c_t, C) = \prod_{i=1}^{|u|} s(u_i \mid u_{1..i-1}, \ell, c_t, C)$$

The flag ℓ is embedded as a single additional dimension that is concatenated alongside the context and input (previous token) vectors for the encoder. See Appendix A for additional training details.

5.3.2 Pragmatic informativeness

As mentioned at the start of this chapter, we evaluate the two models on a measure of *pragmatic informativeness*: how well does the model represent a human speaker, such that a generative model of a listener can be built from it to interpret utterances?

Formally, for a speaker $S_0(u \mid \ell, t, C)$ and an example consisting of an utterance, language identifier, and color context $\langle u, \ell, C \rangle$, we identify the t^* that maximizes the probability of u according to S_0:

$$t^* = \arg\max_t S_0(u \mid t, C)$$

This is the same as choosing the highest-likelihood prediction according to an RSA L_1 model. The pragmatic informativeness of a speaker is the proportion of target colors in a test set correctly identified by t^*, or the accuracy of the L_1 listener.

One drawback of this metric is it does not evaluate how faithful the model is to the overall distribution of human utterances, only the relative conditional likelihoods of human utterances for different target colors. In practice, since the agents are trained to minimize log likelihood, we do not observe our agents frequently producing wildly unhumanlike utterances; however, this is a caveat to keep in mind for evaluating agents that do not naturally approximate a language model.

The understanding model implied in this metric is equivalent to the Rational Observer model of McMahan and Stone (2015), with the difference that our model is a neural network that makes a combined judgment of applicability (semantic appropriateness) and availability (utterance prior), instead of modeling the two components separately. However, we stop short of directly predicting the referent of an expression discriminatively, as is done by e.g. Kennington and Schlangen (2015), so as to require a model that is usable as a speaker.

A related metric is *communicative success* as defined by Golland et al. (2010), which judges the speaker by the accuracy of a human listener when given model-produced utterances. Our pragmatic informativeness metric instead gives a model-derived listener human utterances and assesses its accuracy at identifying colors. Pragmatic informativeness has the advantage of not requiring additional expensive human labeling in response to model outputs; it can be assessed on an existing collection of human utterances, and can therefore be considered an automatic metric.

5.3.3 A note on perplexity

Perplexity is a common intrinsic evaluation metric for generation models.[3] However, for comparing monolingual and bilingual models, we found perplexity to be unhelpful, owing largely to its vocabulary-dependent definition. Specifically, if we fix the vocabulary in advance to include tokens from both languages, then the monolingual model performs unreasonably poorly, and bilingual training helps immensely. However, this is an unfair comparison: the monolingual model's high perplexity is dominated by low probabilities assigned to rare tokens in the opposite-language data that it did not see. Thus, perplexity ceases to be a measure of language modeling ability and assumes the role of a proxy for the out-of-vocabulary rate.

On the other hand, if we define the output vocabulary to be the set of tokens seen at least n times in training ($n = 1$ and 2 are common), then monolingual training yields better perplexity than bilingual training, but mainly because including opposite-language training data forces the bilingual model to predict more rare words that would otherwise be replaced with ⟨unk⟩.[4] This produces the counterintuitive result that perplexity initially goes *up* (gets worse) when increasing the amount of training data. (As a pathological case, with no training data, a model can get a perfect perplexity of 1 by predicting ⟨unk⟩ for every token.)

5.4 Model results and analysis

Pragmatic informativeness of the models on English and Chinese data is shown in Table 5.1. The main result is that training a bilingual model helps compared to a Chinese monolingual one; however, the benefit is asymmetrical, as training on monolingual English data is superior for English data to training on a mix of Chinese and English. All differences in Table 5.1 are significant at $p < 0.001$ (approximate

[3]Two other intrinsic metrics, word error rate (WER) and BLEU (Papineni et al., 2002), were at or worse than chance despite qualitatively adequate speaker outputs, due to high diversity in valid outputs for similar contexts. This problem is common in dialogue tasks, for which BLEU is known to be an ineffective speaker evaluation metric (Liu et al., 2016).

[4]The rare words that make this difference are primarily the small number of English words that were used by the Chinese-language participants; no Chinese words were observed in the English data.

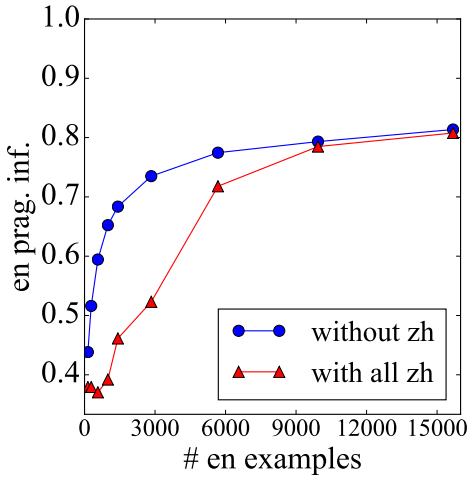
test	train	dev acc	test acc
en	en	**80.51**	**83.06**
	en+zh	79.73	81.43
zh	zh	67.16	67.75
	en+zh	**71.81**	**72.89**

Table 5.1: Pragmatic informativeness scores (%) for monolingual and bilingual speakers.

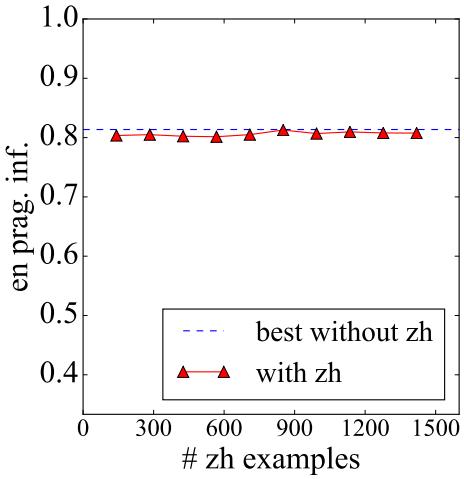
permutation test, 10,000 samples; Padó, 2006), except for the decrease on the English dev set, which is significant at $p < 0.05$.

An important difference between our corpora is that the English dataset is an order of magnitude larger than the Chinese. Intuitively, we expect adding more training data on the same task will improve the model, regardless of language. However, we find that the effect of dataset size is not so straightforward. In fact, the differences in training set size convey a non-linear benefit. Figure 5.5 shows the pragmatic informativeness of the monolingual and bilingual speakers on the development set as a function of dataset size (number of English and Chinese utterances). The blue curves (circles) in the plots on the left, Figure 5.5a and Figure 5.5c, are standard learning curves for the monolingual models, and their parallel red curves (triangles) show the pragmatic informativeness of the bilingual model with the same amount of in-language data plus all available data in the opposite language. The plots on the right, Figure 5.5b and Figure 5.5d, show the effect of gradually adding opposite-language data to the bilingual model starting with all of the in-language data.

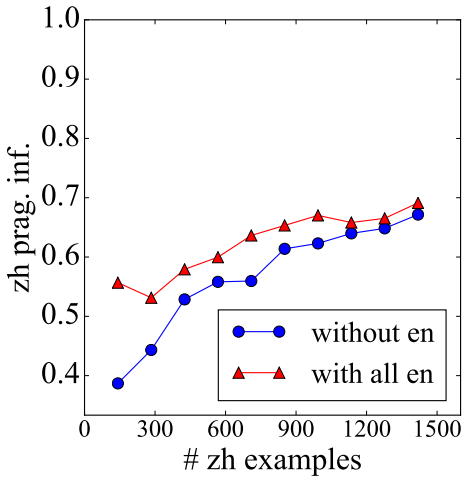
Overall, we see that adding all English data consistently helps the Chinese monolingual model, whereas adding all Chinese data consistently hurts the English monolingual model (though with diminishing effects as the amount of English data increases). Adding small amounts of English data—especially amounts comparable to the size of the Chinese dataset—*decreases* accuracy of the Chinese model dramatically. This suggests an interaction between the total amount of data and the effect of bilingual training: a model trained on a moderately small number of in-language



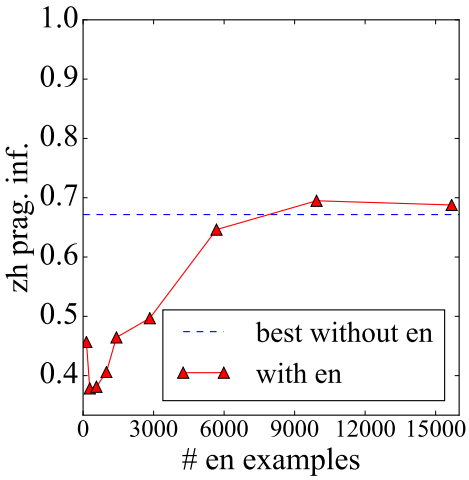
(a) English without any / with all Chinese



(b) All Eng., varying amount of Ch. data



(c) Chinese without any / with all English



(d) All Ch., varying amount of Eng. data

Figure 5.5: Pragmatic informativeness (dev set) for different amounts and languages of training data.

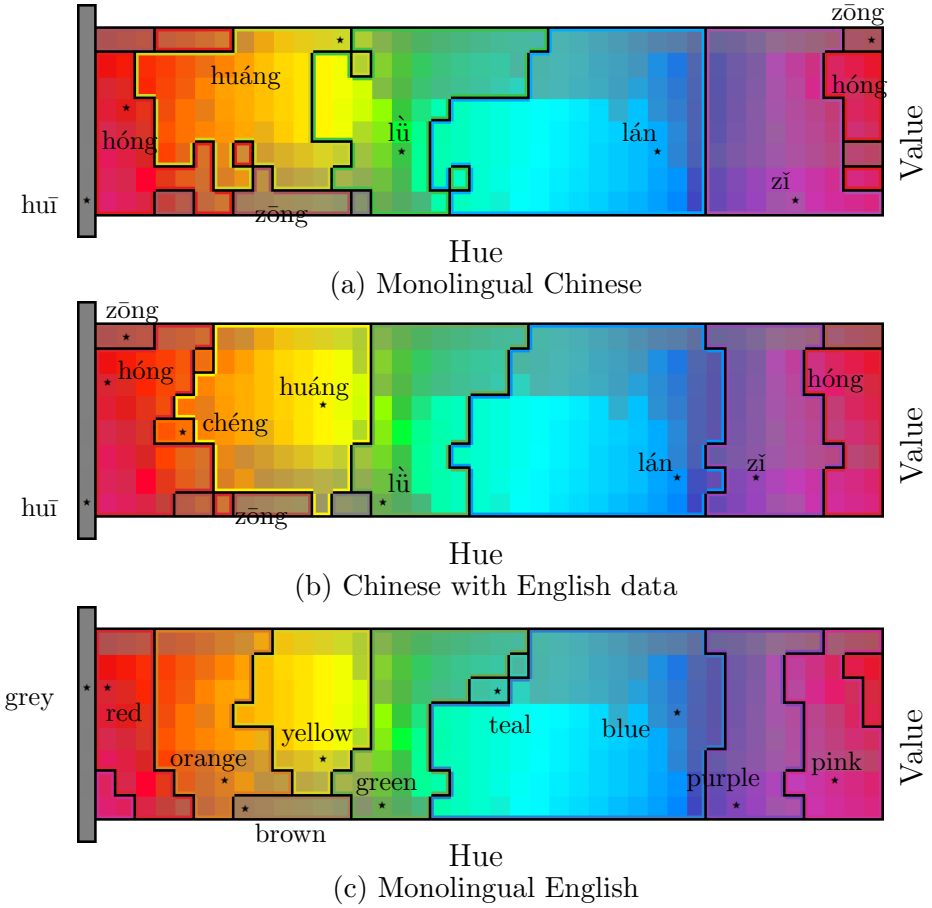
examples can benefit from a much larger training set in another language, but combining data in two languages is detrimental when both datasets are very small and has very little effect when the in-language training set is large. This implies a benefit primarily in low-resource settings, which agrees with the findings of Johnson et al. (2016) using a similar architecture for machine translation.

zh		**en**	**en**	**zh**	
绿色	'green'	**green**	green	**绿**	**'green'**
紫色	'purple'	**purple**	blue	**蓝**	**'blue'**
蓝色	'blue'	purple	purple	蓝	'blue'
灰色	'grey'	**grey**	bright	**鲜艳**	**'bright'**
亮	'bright'	**bright**	pink	**粉色**	**'pink'**
灰	'grey'	-er	grey	**灰**	**'grey'**
蓝	'blue'	*teal*	dark	**暗**	**'dark'**
绿	'green'	**green**	gray	**灰**	**'grey'**
紫	'purple'	**purple**	yellow	**黄色**	**'yellow'**
草	'grass'	*green*	light	最	'most'

Table 5.2: Bilingual lexicon induction from Chinese to English (first two columns) and vice versa (last two). Correct translations in **bold**, semantically close words in *italic*.

5.4.1 Bilingual lexicon induction

To get a better understanding of the influence of the bilingual training on the model's lexical representations in the two languages, we extracted the weights of the final softmax layer of the bilingual speaker model and used them to induce a bilingual lexicon with a word vector analogy task. For two pairs of lexical translations, 蓝色 *lánsè* → "blue" and "red" → 红 *hóng*, we took the difference between the source language word vector and the target language word vector. To "translate" a word, we added this "translation vector" to the word vector for the source word, and found the word in the opposite language with the largest inner product to the resulting vector. The results are presented in Table 5.2. We identified the 10 most frequent color-related words in each language to translate. (In other words, we did not use this process to find translations of function words like "the" or the Chinese nominalization/genitive particle 的 *de*, but we show proposed translations that were not color-related, such as 灰 *huī* being translated as the English comparative ending "-er".) The majority of common color words are translated correctly by this simple method, showing that the vectors in the softmax layer do express a linear correspondence between the representation of synonyms in the two languages.



(a) Monolingual Chinese

(b) Chinese with English data

(c) Monolingual English

Figure 5.6: Color term lexicons: colors in the World Color Survey palette grouped by highest-probability description, averaged over 10 randomly-generated pairs of distractor colors. The color that results in the highest probability of each description is marked with a star. English influences on the bilingual model include the appearance of 橙色 *chéngsè* 'orange' and narrowing of 黄色 *huángsè* 'yellow' and 绿色 *lǜsè* 'green'.

5.4.2 Color term semantics

The above experiment suggests that the bilingual model has learned word semantics in ways that discover translation pairs. However, we wish to know whether bilingual training has resulted in changes to the model's output distribution reflecting differences in the two languages' color systems. To evaluate this, we performed an experiment similar to the basic color term elicitations in the World Color Survey

(WCS; Berlin and Kay, 1969) on our models. For each of the 330 colors in the original WCS, we presented that color to our monolingual and bilingual models and recorded the most likely color description according to the conditional language model. Our models require a three-color context to produce a description; as an approximation to eliciting context-insensitive color terms, we gave the model ten contexts with randomly generated (uniform in H, S, and V) distractor colors and averaged the language model probabilities. We also identified, for each color term produced as the most likely description of one or more colors, the color that resulted in the highest probability of producing that term.

The results are in Figure 5.6. The charts use the layout of the WCS stimulus, in which the two axes represent dimensions of color variation similar to hue and lightness. Each region represents a set of colors that the model labeled with the same color term, and a star marks the color that resulted in the highest probability of producing that term. The Chinese terms, except for 红 *hóng*, are abbreviated by deleting the final morpheme 色 *sè* 'color'.

The charts agree with Berlin and Kay (1969) on most of the differences between the two languages: *orange* and *pink* have clear regions of dominance in English, whereas in the Mandarin monolingual model *pink* is subsumed by 红 *hóng* 'red', and *orange* is subsumed by 黄色 *huángsè* 'yellow'. Our models produce three colors not in the six-color system[5] identified by Berlin and Kay for Mandarin: 灰色 *huīsè* 'grey', 紫色 *zǐsè* 'purple', and 棕色 *zōngsè* 'brown'. We do not specifically claim these should be considered *basic* color terms, since Berlin and Kay give a theoretical definition of "basic color term" that is not rigorously captured by our model. In particular, they explicitly exclude 灰色 *huīsè* from the set of basic color terms, despite its frequency, because it has a meaning that refers to an object ('ashes'). The other two may have been excluded for the same reason, or they may represent a change in the language or the influence of English on the participants' usage.[6]

[5]Notably absent are 'black' and 'white'. Our collection methodology restricted colors to a single lightness, so black and white are not in the data. For these charts, we replaced the World Color Survey swatches with the closest color used in our data collection.

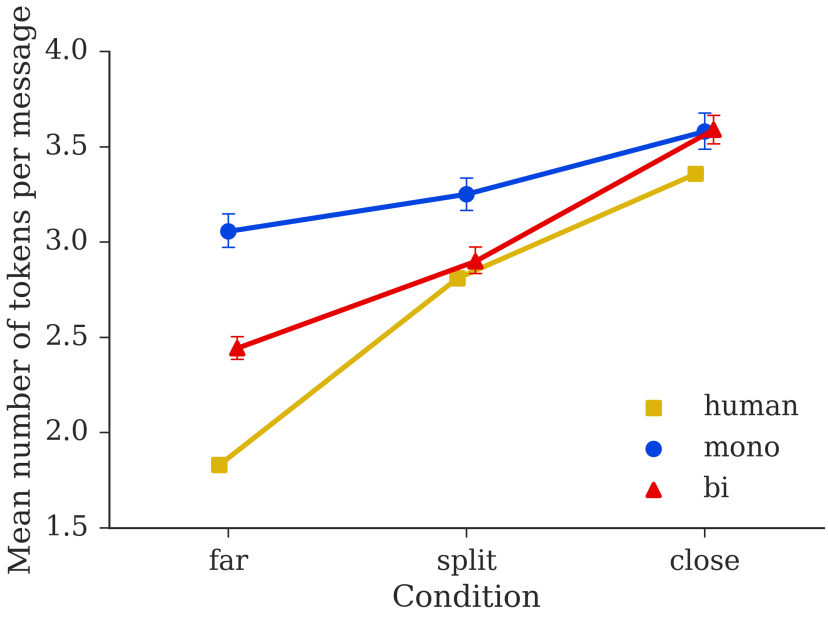
[6]MTurk's restriction to US workers makes English influence more likely than would otherwise be expected.

A few differences between the monolingual and bilingual models can be characterized as an influence of one language's color system on the other. First, *teal* appears as a common description of a few color swatches from the English monolingual model, but the bilingual model, like the Chinese model, does not feature a common word for teal. Second, the Chinese monolingual model does not include a common word for orange, but the bilingual model identifies 橙色 *chéngsè* 'orange'. Finally, the English *green* is semantically narrower than the Chinese 绿色 *lǜsè*, and the Chinese bilingual model exhibits a corresponding narrowing of the range of 绿色 *lǜsè*. Overall, however, the monolingual models capture largely accurate maps of each language's basic color system, and the bilingual model retains the major contrasts between them, rather than "averaging" between the two. This suggests that the bilingual model learns a representation of the input colors that encodes their categorization in both languages, and that these lexical semantic representations largely do not influence each other.

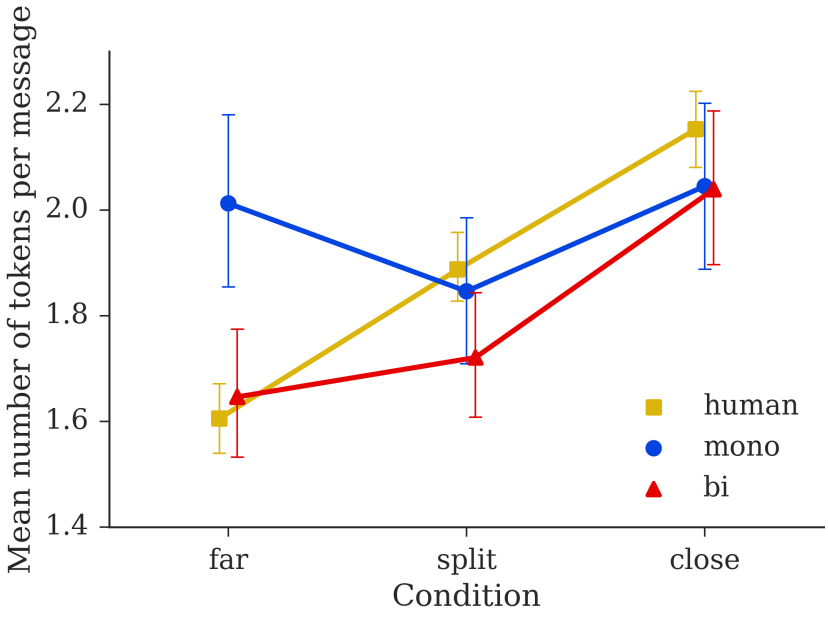
5.4.3 Comparing model and human utterances

The observations of previous section lead us to conclude that the differences resulting from bilingual training do not primarily result from differences in lexical semantic interpretation. Instead, a different observation indicates that the improvements in the bilingually-trained model are primarily at the pragmatic (context-dependent) level of language production. Figure 5.7 reveals that the bilingually-trained model better captures the main pragmatic pattern we observe in the human data, that of increasing message length in harder conditions. In both languages, the monolingual model uses longer utterances in the easy *far* condition than human speakers do, whereas the bilingual model is significantly closer on that condition to the human statistics. We see similar results in the use of negations and comparatives; the use of superlatives is not substantially different between the monolingual and bilingual models.

We note that this result does not rule out several competing hypotheses. In particular, we do not exclude improvements in compositional semantics or syntax, nor



(a) Human and model utterance lengths in English.



(b) Human and model utterance lengths in Chinese.

Figure 5.7: Comparison of mean length of messages between human and model utterances.

do we distinguish improvements in specific linguistic areas from broader regularization effects of having additional data in general. Preliminary experiments involving augmentation of the data by duplicating and deleting constituents show no gains, suggesting that the improvement depends on certain kinds of regularities in the English data that are not provided by artificial manipulations. However, more investigation is needed to thoroughly assess the role of general-purpose regularization in our observations.

5.5 Related work

The method we use to build a bilingual model involves adding a single dimension to the previous-token vectors in the encoder representing the language (Section 5.3). In essence, the two languages have separate vocabulary representation at the input and output but shared hidden representations. Adding a hard constraint on the output vocabulary would make this equivalent to a simple form of multitask learning (Caruana, 1997; Collobert and Weston, 2008). However, allowing the model to use tokens from either language at any time is simpler and results in better modeling of mixed-language data, which is more common in non-English environments. In fact, our model occasionally ignores the flag and "code-switches" between the two languages within a single output, which is not possible in typical multitask architectures.

Using shared parameters for cross-lingual representation transfer has a large literature. Klementiev et al. (2012) and Hermann and Blunsom (2014) use multitask learning with multilingual document classification to build cross-lingual word vectors, and observe accurate lexical translations from linear vector analogy operations. They include predicting translations for words in parallel data as one of their tasks. Our translations from vector relationships (Section 5.4.1) derive their cross-lingual relationships from the non-linguistic input of our grounded task, without parallel data.

Huang et al. (2013) note gains in speech recognition from cross-lingual learning with shared parameters. In machine translation, Johnson et al. (2016) add the approach of setting the output language using a symbol in the input. Kaiser et al.

(2017) extend this to image captioning, speech recognition, and parsing in one multitask system. Our work complements these efforts with an in-depth analysis of bilingual training on a grounded generation task and an exploration of the relationship between cross-lingual semantic differences and pragmatics. In general, we see grounding in non-linguistic input, including images and sensory input from real and simulated worlds, as an intriguing substitute for direct linguistic supervision in low-resource settings. We encourage evaluation of multitask and multilingual models on tasks that require reference to the context for effective language production and understanding.

5.6 Discussion

This chapter studied the effects of training on bilingual data in the color reference game setting. It provides evidence that bilingual training can be helpful, but with a non-obvious effect of dataset size: accuracy as a function of opposite-language data follows a U-shaped curve. The resulting model is more human-like in measures of sensitivity to contextual difficulty (pragmatics), while exhibiting language-specific lexical learning in the form of vector relationships between lexical pairs and differences between the two languages in common color-term extensions (semantics).

It should be noted that color descriptions in English and Chinese are similar both in their syntax and in the way they divide up the semantic space. We might expect that for languages like Arabic and Spanish (with their different placement of modifiers), or Waorani and Pirahã (with their much smaller color term inventories), adding English data could have detrimental effects that outweigh the language-general gains. An investigation across a broader range of languages is desirable.

Our contribution includes a new dataset of human utterances in a color reference game in Mandarin Chinese, which is available to the public alongside the corresponding English data[7]. Code and trained model parameters are also available online for experiment replication.[8]

[7] `https://cocolab.stanford.edu/datasets/colors.html`
[8] `https://github.com/futurulus/colors-in-context`

Chapter 6

Conclusion

This dissertation has demonstrated benefits of modeling the relationship between speaker and listener using a combination of machine learning models and the RSA insight. Chapter 1 summarized the successes of RSA and similar models in modeling human behavior in restricted settings. Chapter 2 showed how to train a machine learning model with the RSA recursive structure embedded in it, allowing the model to learn word meanings and patterns of speaker behavior from examples, while improving its ability to adapt to the context. Here, a speaker model guides a listener's learning. In Chapter 3 we saw the benefits of neural models of grounded language for tasks involving high diversity of both utterances and referents, and Chapter 4 incorporated neural models into an RSA-based listener that is able to make use of training on the speaker task. Finally, Chapter 5 proposed using a speaker-based listener as a way of evaluating the speaker on its ability to describe referents distinctively. Along the way, these lines of work have yielded better speakers and listeners in reference game settings.

The final sections of each of the preceding chapters have mentioned shortcomings of the models they introduce and specific areas that have the potential for improvement. In this chapter, I conclude with a broader discussion of promising topics of future research in this area.

6.1 Future directions

6.1.1 Extensions of RSA

The definition of RSA in Section 1.2 is a simplification of the broader family of models used in the cognitive science literature, and can only capture a limited subset of the phenomena that have been studied with RSA-like approaches. More commonly, RSA is extended in some way that broadens the scope of the reasoning performed by the agents.

One extension that has proven to be useful for modeling a variety of pragmatic effects is the addition of *lexical uncertainty*. In a lexical uncertainty RSA model, the semantic interpretation function is not fixed (whether specified a priori or learned from examples), but rather is modeled as a distribution over possible lexicons. A listener model then marginalizes over possible lexicons conditioned on its observations of the speaker's behavior to arrive at a probabilistic interpretation. Such a model accurately predicts pragmatic phenomena such as Horn implicature (that costly utterances are likely to express unusual situations; Bergen et al., 2012) and nonce understanding of new words in context (Smith et al., 2013; Frank and Goodman, 2014).

The use of continuous outputs from a machine learning model instead of a binary-valued semantic interpretation function is related to the notion of a distribution over lexicons, in that it can be imagined as a graded or probabilistic judgment of semantic compatibility. It even avoids some shortcomings of fuzzy logic, by providing the ability to produce compositional semantic judgments that depend on more than just semantic judgments about the parts (Kamp and Partee, 1995). However, the RNN models used in this dissertation cannot do inference on a global distribution over lexicons, instead assigning a "probability" of each utterance being true in context independently of other utterances.

Another approach that has been fruitful for predicting pragmatic behavior in cognitive science experiments is modeling joint reasoning about language, the objective state of the world, and subjective processes such as the other agent's goals or emotions. A model of a pragmatic listener that reasons about the speaker's goals yields accurate predictions of metaphor (Kao et al., 2014a) and numerical hyperbole (Kao

et al., 2014b). That is, when a speaker uses an utterance such as "John is a bear" to mean that John is a large and hairy human, or "this laptop cost a million dollars" if the laptop cost $5,000, a listener can consider the possibility that the speaker's goal is to convey a limited subset of John's attributes, or a broad understanding that the laptop is expensive, and understand the intended meaning without necessarily accepting the literal meaning.

These extensions involve marginalizing over latent variables that are tractable to enumerate in restricted settings but intractably large in general. Implementing these with machine learning approaches is an enticing project, but will likely require an approximation to this marginalization.

6.1.2 Further expansion of referent and utterance spaces

As was briefly mentioned at the end of Chapter 4, the use of neural network representations makes the models presented here readily amenable to referents with high-dimensional structure, as long as a scheme for building an embedding of such a referent exists. This includes replacing the colors of Chapters 3 through 5 with images, videos, or natural language documents. However, all of the models used in this dissertation still require enumerating over a finite set of such complicated referents. This enumeration is easy in three-object language games, but it gets considerably harder when the context gets larger.

For example, consider a virtual world constructed from triangular meshes, as is common in games and animation. In such a world, one might want to make reference to parts of such meshes ("the front left corner of the desk in the back", "the handle of the teacup"). In general, this requires a referent space consisting of all subsets of the triangles in the meshes. An important open question in grounded language understanding is how to avoid the exponential blowup in the size of the search space that comes with considering referents composed of multiple objects or parts of objects.

With more complicated referents comes more complicated utterances. Chapter 3 analyzed the ability of a RNN model to produce utterances employing certain kinds

of compositional structure, motivating the use of RNNs as the foundation of RSA-based models in the color description task. However, this analysis focused only on one type of composition common in color descriptions, the combination of an adjectival modifier with a basic color word. The variety of compositional structures in language in general is vast, and much of this variety comes from the need to describe referents in the world more complicated than colors. The interaction between RSA modeling of pragmatics and compositional semantics is the subject of recent research (Potts et al., 2015; Bergen et al., 2016). Testing the implications of RSA for verb phrases, prepositional phrases, and clauses, just to name some of the most common forms of linguistic composition, requires a task that elicits such structures.

Image captioning is one promising task for this (Karpathy and Fei-Fei, 2015; Vinyals et al., 2015), as it requires expressing actions, spatial relations, and other kinds of interactions between objects in a scene. Andreas and Klein (2016), who introduced the utterance sampling method used in Chapter 4, applied their model (equivalent to L_a) to a task of captioning clip-art scenes (Zitnick and Parikh, 2013) for a reference game. Cohn-Gordon et al. (2018) use natural images and apply an RSA approach at each token or character in decoding. They find that this incremental approach produces substantial gains using a metric similar to pragmatic informativeness (Section 5.3.2), but evaluating a speaker's generated utterances (instead of human utterances) using the accuracy of an L_1 model trained on disjoint data.

Another type of language that is mostly unattested in color descriptions but essential for a general language understanding system is temporal language. Accounting for temporal language requires a dynamic context, such as a story, a video, or a virtual world with moving characters. Fried et al. (2018) apply the approach of Andreas and Klein (2016) to a dynamic instruction-following environment, and observe improvements over their base RNN L_0.

6.1.3 Partial information

In the work described in this dissertation, the speaker has complete knowledge of the state of the world (the context and the identity of the target), while the listener

knows everything except the identity of the target. This property is not a fundamental assumption of the machine learning models that serve as the L_0 and S_0 agents in this work—the RNN models in particular do not need to know how much knowledge of the world state is hidden from them to optimize their objective functions or make classification decisions (though making accurate decisions becomes more difficult as more necessary information is hidden). However, the RSA equations as written in Chapter 1 assume the same context for speaker and listener. Partial observability breaks this symmetry: a speaker reasoning about a listener cannot simply use a model trained from the listener's perspective, since that model needs information that the speaker does not know.

Formally, if the listener receives an observation o_L conditioned on the world state t according to a probability distribution $\Omega_L(o_L \mid t)$, and the speaker receives an observation o_S according to $\Omega_S(o_S \mid t)$, then the RSA recursion equations (1.2) and (1.1) must be modified to compute expected utility under uncertainty about the world. If the goal of both agents is for the listener to guess the complete state of the world t, a full specification of the relationship between speaker and listener could be given by:

$$S_n(u \mid o_S) \propto \exp\left(\lambda\left(\mathbb{E}_{t|o_S}\left[\mathbb{E}_{o_L \sim \Omega_L(\cdot|t)}\left[\log L_{n-1}(t \mid u, o_L)\right]\right] - K(u)\right)\right) \tag{6.1}$$

$$L_n(t \mid u, o_L) \propto P(t)\Omega_L(o_L \mid t) \cdot \mathbb{E}_{o_S \sim \Omega_S(\cdot|t)}\left[S_{n-1}(u \mid o_S)\right] \tag{6.2}$$

Alternatively, the expectations in these computations could be folded into a "lower-information" model that does not explicitly consider its own observation, but represents the other agent's distribution over choices as seen by an external observer who does not have access to the agent's observations:

$$S_n(u \mid o_S) \propto \exp\left(\lambda\left(\mathbb{E}_{t|o_S}\left[\log \tilde{L}_{n-1}(t \mid u)\right] - K(u)\right)\right) \tag{6.3}$$

$$L_n(t \mid u, o_L) \propto P(t)\Omega_L(o_L \mid t) \cdot \tilde{S}_{n-1}(u \mid t) \tag{6.4}$$

This has the same form as the equations in Section 1.2, except for the inclusion of the level n agent's observation probabilities. However, the choice of whether to

model the actual listener and speaker's pragmatic context or maintain separate models of the other agent marginalizing over observations is a design decision for machine learning–based models that could impact the resulting agents' performance.

In the presence of partial information, this becomes equivalent to a decentralized partially-observable Markov decision process (Dec-POMDP). Choosing optimal actions in Dec-POMDPs is infamously intractable: NEXP-complete (Bernstein et al., 2002, cited in Vogel et al., 2013).

6.1.4 Partial reward alignment

Another simplifying assumption that was implicit in the use of a reference game is that speaker and listener have the same goal (for the listener to correctly identify the target). This assumption is wrong in many useful scenarios, and predecessors of RSA (Camerer et al., 2004; Franke, 2010; Jäger, 2011) have devoted considerable attention to communicative settings in which agents have opposing or partially aligned incentives.

An example of a setting that emphasizes this aspect, in addition to many of the other challenges mentioned above, is the negotiation task of Lewis et al. (2017). Their task, which is based on that of DeVault et al. (2015), features two players in equal roles taking turns as speaker and listener, who have an interest in acquiring a set of objects that are "in play" (not yet belonging to either player). The objects, which in this game are hats, books, and balls, have different values for each player; for example, one player may get a payoff of 5 for each book the player receives, while the other only gets a payoff of 2 per book. Players can only see their own values; the other player's values are hidden. In order to receive a payoff, the players must settle on an agreement to divide up the objects between the two of them. The values of the objects are specified such that is is not possible for both players to receive the maximum payoff, so they must negotiate a compromise.

This task is not a game of perfect cooperation, but neither is it a zero-sum game: by sharing information about which objects are more valuable to which player, the players can arrive at a deal that is mutually beneficial. The main modification to

an RSA model that is needed to handle partially aligned incentives is to replace the speaker's utility function: instead of prioritizing the likelihood that the listener recovers the true world state, the speaker should maximize its own expected reward. In many cases, the optimal action may involve producing informative messages, but where the agents' goals conflict, keeping quiet or lying may be preferable.

6.1.5 Dialogue planning

One-off reference game tasks present little need for speakers and listeners to model back-and-forth dialogue. However, dialogue is a central reason the RSA insight is valuable: the symmetry between speaker and listener arises because people are so often on equal footing and able to act in both roles.

Dialogue agents are more challenging to model than isolated speakers and listeners, requiring long-term planning, remembering previous utterances, and (for the listener) deciding when to ask for clarification or commit to a referent (Lewis, 1979; Brown and Yule, 1983; Clark, 1996; Roberts, 1996). When using RSA models, multi-turn dialogues introduce several new complications. One is the need to update a listener agent's distribution over world states in response to each new speaker utterance, maintaining the information provided by previous utterances. This can be handled with a straightforward Bayesian update rule, if the size of the state space is small enough to enumerate. Another is that the speaker's action space is no longer a single utterance but a sequence of them, with the other agent's utterances interspersed among them. Learning to choose the right utterance involves dealing with delayed reward, suggesting the use of reinforcement learning approaches.

The above negotiation task also involves multi-turn dialogues: exchanging information, making offers and counter-offers, and clarifying the final deal. Lewis et al. (2017) implement an RNN dialogue agent trained with reinforcement learning and use random rollouts from this model as a means of planning ahead in dialogue production, estimating the expected reward for various candidate utterances and choosing the one with the highest estimate. Their agent models both sides of the conversation based only on one agent's private values. This corresponds to building the

"low-information" models of equations (6.3) and (6.4).

An alternative approach would be to build an explicit model of the opponent's values given the dialogue so far and use that as input to the model of the opponent, following the formulation in (6.1) and (6.2). This approach has several potential benefits: it convinces the model to pay attention to a latent variable that is known to be important, and it allows the model to take advantage of supervision from the opponent's true motives, which are available in the training data from human games. Finally, it improves transparency of the model: when the model makes a mistake, one can examine the inferred opponent's goals to determine if they were incorrect and leading the model astray. The main downside is that the model is no longer free to choose a representation of the goals and uncertainty about those goals, but instead must work with the inferred goals. Experiments are necessary to determine whether this "hard inference" procedure would be beneficial.

The use of rollouts by Lewis et al. (2017) does not allow taking into account the prospect of an opponent that also plans ahead to maximize reward. A more systematic treatment is given by Khani et al. (2018). They combine the recursive pragmatic reasoning of RSA, inference of the other agent's hidden knowledge from past utterances, and planning of future utterances to maximize expected reward, to produce a unified model of pragmatic dialogue production. This very recent work is an exciting example of the continuing success resulting from adapting RSA models to increasingly challenging domains. My hope in publishing this dissertation is to spur further similar work that can expand the boundaries of the linguistic phenomena that can be captured by general-purpose computational language systems.

Appendix A

Hyperparameters and other model details

Hyperparameters for the main models described in this dissertation are given in Table A.1. The learned RSA model hyperparameters were tuned by grid search and cross-validation. The RNN color description model of Chapter 3 and the base speaker and listener in Chapter 4 were tuned for perplexity on a held-out subset of the training set by random search. Both Chinese monolingual and bilingual model were tuned by random search followed by a local search from the best candidate until no single parameter change produced a better result. However, the tuned settings for the Chinese monolingual model did not outperform the settings from Chapter 4's base speaker for the English model on the development set, so the monolingual models in the final experiments shown in Chapter 5 used the same parameters.

Starting in Chapter 4, the vocabulary for each model consisted of all tokens that were seen at least twice in training. The bilingual model's vocabulary in is larger than the union of the words in each monolingual model because some tokens occurred once in each language (largely meta-commentary—e.g., *dunno*, *HIT*, *xD*—and some English color word typos).

hyperparameter	S_0 (TUNA)	S_1 (TUNA)	L_0 (colors)	S_0 (colors, mono.)	S_0 (colors, biling.)
optimizer	AdaGrad	AdaGrad	AdaGrad	ADAM	RMSProp
training epochs	50	10	30	15	15
learning rate	0.01	0.01	0.1	0.004	0.004
dropout	–	–	0.2	0.1	0.1
ℓ_2 regularization ℓ	0.01	0.01	–	–	–
gradient clip norm	–	–	–	–	1
LSTM cell size	–	–	20	100	50
embedding/feat. size	497–663 (people) 1244–1344 (furn.)		20	100	100
initial forget bias	–	–	5	0	5
nonlinearity σ_y	–	–	ReLU	tanh	sigmoid
vocabulary size	24 (people) 32 (furn.)		367	895 (en) 260 (zh)	1,326

Table A.1: Values of hyperparameters for each of the trained models described in the previous chapters, plus vocabulary sizes.

Bibliography

Hirotugu Akaike. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, et al. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*.

Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. In *Proceedings of the 2016 Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Bert Baumgaertner, Raquel Fernandez, and Matthew Stone. 2012. Towards a flexible semantics: Colour terms in collaborative reference tasks. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155.

Leon Bergen, Noah D. Goodman, and Roger Levy. 2012. That's what she (could have) said: How alternative utterances affect language use. In *Proceedings of the 34th Annual Meeting of the Cognitive Science Society*.

Leon Bergen, Roger Levy, and Noah D. Goodman. 2016. Pragmatic reasoning through semantic inference. *Semantics and Pragmatics*, 9:20.

Brent Berlin and Paul Kay. 1969. *Basic Color Terms: Their Universality and Evolution*. University of California Press.

Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819–840.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the 19th International Conference on Computational Statistics*.

Gillian Brown and George Yule. 1983. *Discourse Analysis*. Cambridge University Press.

Sarah Brown-Schmidt and Michael K. Tanenhaus. 2008. Real-time investigation of referential domains in unscripted conversation: A targeted language game approach. *Cognitive Science*, 32(4):643–684.

Colin F. Camerer, Teck-Hua Ho, and Juin-Kuan Chong. 2004. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3):861–898.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28:41–75.

Stanley F. Chen and Ronald Rosenfeld. 1999. A Gaussian prior for smoothing maximum entropy models. Technical Report CMU-CS-99-108, Carnegie Mellon University.

Gennaro Chierchia. 2004. Scalar implicatures, polarity phenomena, and the syntax/pragmatics interface. *Structures and Beyond: The Cartography of Syntactic Structures*, 3:39.

Herbert H. Clark. 1996. *Using Language*. Cambridge University Press.

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

Reuben Cohn-Gordon, Noah D. Goodman, and Christopher Potts. 2018. Pragmatically informative image captioning with character-level inference. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

James N. Collins. 2016. Reasoning about definiteness in a language without articles. *Semantics and Linguistic Theory*, 26:82–102.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*.

Richard S. Cook, Paul Kay, and Terry Regier. 2005. The World Color Survey database. *Handbook of Categorization in Cognitive Science*, pages 223–241.

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

Kees van Deemter, Ielka van der Sluis, and Albert Gatt. 2006. Building a semantically transparent corpus for the generation of referring expressions. In *Proceedings of the Fourth International Natural Language Generation Conference*.

Judith Degen and Michael K. Tanenhaus. 2015. Availability of alternatives and the processing of scalar implicatures: A visual world eye-tracking study. *Cognitive Science*, 40(1):172–201.

David DeVault, Johnathan Mell, and Jonathan Gratch. 2015. Toward natural turn-taking in a virtual human negotiation agent. In *AAAI Spring Symposium on Turn-Taking and Coordination in Human-Machine Interaction*.

Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, et al. 2015. Lasagne: First release.

Pedro Domingos. 2012. A few useful things to know about machine learning. *Communications of ACM*, 55(10):78–87.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, pages 2121–2159.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998.

Michael C. Frank and Noah D. Goodman. 2014. Inferring word meanings by assuming that speakers are informative. *Cognitive Psychology*, 75(1):80–96.

Michael Franke. 2009. *Signal to Act: Game Theory in Pragmatics*. ILLC Dissertation Series. Institute for Logic, Language and Computation, University of Amsterdam.

Michael Franke. 2010. Semantic meaning and pragmatic inference in non-cooperative conversation. In *Interfaces: Explorations in Logic, Language and Computation*, pages 13–24. Springer.

Daniel Fried, Jacob Andreas, and Dan Klein. 2018. Unified pragmatic models for generating and following instructions. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Albert Gatt, Anja Belz, and Eric Kow. 2008. The TUNA Challenge 2008: Overview and evaluation results. In *Proceedings of the Fifth International Natural Language Generation Conference*.

Albert Gatt, Roger P. G. van Gompel, Kees van Deemter, and Emiel Krahmer. 2013. Are we Bayesian referring expression generators? In *Proceedings of the 35th Annual Conference of the Cognitive Science Society*.

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

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*.

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

Noah D. Goodman and Michael C. Frank. 2016. Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*, 20(11):818–829.

Noah D. Goodman and Andreas Stuhlmüller. 2013. Knowledge and implicature: Modeling language understanding as social cognition. *Topics in Cognitive Science*, 5(1):173–184.

Thomas Grano. 2012. Mandarin 'hen' and universal markedness in gradable adjectives. *Natural Language & Linguistic Theory*, 30(2):513–565.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

H. Paul Grice. 1975. Logic and conversation. In Peter Cole and Jerry Morgan, editors, *Syntax and Semantics*, volume 3: Speech Acts, pages 43–58. Academic Press.

Joy E. Hanna, Michael K. Tanenhaus, and John C. Trueswell. 2003. The effects of common ground and perspective on domains of referential interpretation. *Journal of Memory and Language*, 49(1):43–61.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*, 2nd edition. Springer.

Robert X.D. Hawkins. 2015. Conducting real-time multiplayer experiments on the web. *Behavior Research Methods*, 47(4):966–976.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. 2013. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Satomi Ito. 2008. Typology of comparatives. In *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation (PACLIC)*.

Gerhard Jäger. 2011. Game theory in semantics and pragmatics. In Claudia Maienborn, Klaus von Heusinger, and Paul Portner, editors, *Semantics: An International Handbook of Natural Language Meaning*, pages 2487–2425. De Gruyter Mouton.

Gerhard Jäger. 2014. Rationalizable signaling. *Erkenntnis*, 79(4):673–706.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Sun Junyi. 2015. Jieba Python Library.

Łukasz Kaiser, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. 2017. One model to learn them all. *arXiv preprint arXiv:1706.05137*.

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

Justine T. Kao, Leon Bergen, and Noah D. Goodman. 2014a. Formalizing the pragmatics of metaphor understanding. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Society*.

Justine T. Kao, Jean Y. Wu, Leon Bergen, and Noah D. Goodman. 2014b. Nonliteral understanding of number words. *Proceedings of the National Academy of Sciences*, 111(33):12002–12007.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kazuya Kawakami, Chris Dyer, Bryan Routledge, and Noah A. Smith. 2016. Character sequence models for colorful words. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Casey Kennington and David Schlangen. 2015. Simple learning and compositional application of perceptually grounded word meanings for incremental reference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Fereshte Khani, Noah D. Goodman, and Percy Liang. 2018. Planning, inference, and pragmatics in sequential language games. *Transactions of the Association for Computational Linguistics (TACL)*, 6.

Diederik P. Kingma and Jimmy Lei Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*.

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

Emiel Krahmer and Kees Van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

Robert M. Krauss and Sidney Weinheimer. 1964. Changes in reference phrases as a function of frequency of usage in social interaction: A preliminary study. *Psychonomic Science*, 1(1–12):113–114.

Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. 2011. Baby talk: Understanding and generating image descriptions. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

David Lewis. 1969. *Convention*. Harvard University Press. Reprinted 2002 by Blackwell.

David Lewis. 1979. Scorekeeping in a language game. *Journal of Philosophical Logic*, 8(1):339–359.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? End-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Percy Liang and Christopher Potts. 2015. Bringing machine learning and compositional semantics together. *Annual Review of Linguistics*, 1(1):355–376.

Chia-Wei Liu, Ryan Lowe, Iulian V. Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

R. Duncan Luce. 1959. *Individual Choice Behavior: A Theoretical Analysis*. Wiley.

Peter McCullagh and John A. Nelder. 1989. *Generalized Linear Models*. Chapman and Hall.

Richard D. McKelvey and Thomas R. Palfrey. 1995. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10(1):6–38.

Brian McMahan and Matthew Stone. 2015. A Bayesian model of grounded color semantics. *Transactions of the Association for Computational Linguistics (TACL)*, 3:103–115.

Timothy Meo, Brian McMahan, and Matthew Stone. 2014. Generating and resolving vague color references. In *Proceedings of the 18th Workshop on the Semantics and Pragmatics of Dialogue (SemDial)*.

Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, Kota Yamaguchi, Tamara Berg, Karl Stratos, and Hal Daumé III. 2012. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Will Monroe, Noah D. Goodman, and Christopher Potts. 2016. Learning to generate compositional color descriptions. In *Proceedings of the 2016 Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Will Monroe, Robert X.D. Hawkins, Noah D. Goodman, and Christopher Potts. 2017. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics (TACL)*, 5:325–338.

Will Monroe, Jennifer Hu, Andrew Jong, and Christopher Potts. 2018. Generating bilingual pragmatic color references. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Will Monroe and Christopher Potts. 2015. Learning in the Rational Speech Acts model. In *Proceedings of the 20th Amsterdam Colloquium*.

Randall Munroe. 2010. Color survey results. Online at http://blog.xkcd.com/2010/05/03/color-surveyresults.

Sebastian Padó. 2006. *User's guide to* ***sigf****: Significance testing by approximate randomisation*. `http://www.nlpado.de/~sebastian/software/sigf.shtml`.

Maike Paetzel, David Nicolas Racca, and David DeVault. 2014. A multimodal corpus of rapid dialogue games. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Christopher Potts, Daniel Lassiter, Roger Levy, and Michael C. Frank. 2015. Embedded implicatures as pragmatic inferences under compositional lexical uncertainty. *Journal of Semantics*.

Josef Raviv. 1967. Decision making in Markov chains applied to the problem of pattern recognition. *IEEE Transactions on Information Theory*, 13(4):536–551.

Craige Roberts. 1996. Information structure in discourse: Towards an integrated formal theory of pragmatics. *Working Papers in Linguistics—Ohio State University Department of Linguistics*, pages 91–136.

Seymour Rosenberg and Bertram D. Cohen. 1964. Speakers' and listeners' processes in a word communication task. *Science*, 145:1201–1203.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning internal representations by error propagation. In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1: Foundations, pages 318–362. MIT Press.

Gaurav Sharma, Wencheng Wu, and Edul N. Dalal. 2005. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30.

Nathaniel J. Smith, Noah D. Goodman, and Michael C. Frank. 2013. Learning and using language via recursive pragmatic reasoning about other agents. In *Advances in Neural Information Processing Systems 26 (NIPS)*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS)*.

Michael K. Tanenhaus and Sarah Brown-Schmidt. 2008. Language processing in the natural world. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 363(1493):1105–1122.

Stefanie Tellex, Ross A. Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. 2014. Asking for help using inverse semantics. In *Robotics: Science and Systems*.

Joshua B. Tenenbaum, Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman. 2011. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Adam Vogel, Max Bodoia, Christopher Potts, and Dan Jurafsky. 2013. Emergence of Gricean maxims from multi-agent decision theory. In *Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.

Adam Vogel, Andrés Gómez Emilsson, Michael C. Frank, Dan Jurafsky, and Christopher Potts. 2014. Learning to reason pragmatically with cognitive limitations. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Society*.

Kefei Wang and Hongwu Qin. 2010. A parallel corpus-based study of translational chinese. In Richard Xiao, editor, *Using Corpora in Contrastive and Translation Studies*, pages 164–181. Cambridge Scholars Publishing.

Richard M. Warren. 1970. Perceptual restoration of missing speech sounds. *Science*, 167(3917):392–393.

Yun Xia. 2014. *Normalization in Translation: Corpus-based Diachronic Research into Twentieth-century English Chinese Fictional Translation*. Cambridge Scholars Publishing.

Yaoji Xie. 2001. 汉语语法欧化综述 / Hànyǔ yǔfǎ Ōuhuà zōngshù (A review of Europeanized Chinese grammar). 语文研究 *Yǔwén Yánjiū (Chinese Language Research)*, 1:17–22.

Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhihong Zeng, Maja Pantic, Glenn I. Roisman, and Thomas S. Huang. 2009. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58.

Dengsheng Zhang and Guojun Lu. 2002. Shape-based image retrieval using generic Fourier descriptor. *Signal Processing: Image Communication*, 17(10):825–848.

Dexi Zhu. 1982. 语法讲义 / *Yǔfǎ Jiǎngyì (Lectures on Grammar)*. The Commercial Press.

C. Lawrence Zitnick and Devi Parikh. 2013. Bringing semantics into focus using visual abstraction. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.