Reasoning about Pragmatics with Neural Listeners and Speakers

Jacob Andreas and **Dan Klein**
Computer Science Division
University of California, Berkeley
{jda,klein}@cs.berkeley.edu

Abstract

We present a model for contrastively describing scenes, in which context-specific behavior results from a combination of inference-driven pragmatics and learned semantics. Like previous learned approaches to language generation, our model uses a simple feature-driven architecture (here a pair of neural "listener" and "speaker" models) to ground language in the world. Like inference-driven approaches to pragmatics, our model actively reasons about listener behavior when selecting utterances. For training, our approach requires only ordinary captions, annotated *without* demonstration of the pragmatic behavior the model ultimately exhibits. In human evaluations on a referring expression game, our approach succeeds 81% of the time, compared to 69% using existing techniques.

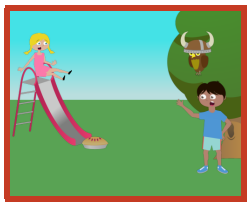
1 Introduction

We present a model for describing scenes and objects by reasoning about context and listener behavior. By incorporating standard neural modules for image retrieval and language modeling into a probabilistic framework for pragmatics, our model generates rich, contextually appropriate descriptions of structured world representations.

This paper focuses on a *reference game* RG played between a listener L and a speaker S.

1. Reference candidates r_1 and r_2 are revealed to both players.
2. S is secretly assigned a random target $t \in \{1, 2\}$.
3. S produces a description $d = S(t, r_1, r_2)$, which is shown to L.
4. L chooses $c = L(d, r_1, r_2)$.
5. Both players win if $c = t$.

(RG)



(a) target

(b) distractor

the owl is sitting in the tree

(c) description

Figure 1: Sample output from our model. When presented with a target image (a) in contrast with a distractor image (b), the model generates a description (c). This description mentions a *tree*, the distinguishing object present in (a) but not in (b), and situates it with respect to other objects and events in the scene.

Figure 1 shows an example drawn from a standard captioning dataset (Zitnick et al., 2014).

In order for the players to win, S's description d must be *pragmatic*: it must be informative, fluent, concise, and must ultimately encode an understanding of L's behavior. In Figure 1, for example, *the owl is wearing a hat* and *the owl is sitting in the tree* are both accurate descriptions of the target image, but only the second allows a human listener to succeed with high probability. RG is the focus of many papers in the computational pragmatics literature: it provides a concrete generation task while eliciting a broad range of pragmatic behaviors, including conversational implicature (Benotti and Traum, 2009) and context dependence (Smith et al., 2013). Existing computational models of pragmatics can be divided into two broad lines of work, which we term the *direct* and *derived* approaches.

Direct models (see Section 2 for examples) are based on a representation of S. They learn pragmatic behavior by example. Beginning with datasets annotated for the specific task they are trying to

solve (e.g. examples of humans playing RG), direct models use feature-based architectures to predict appropriate behavior without a listener representation. While quite general in principle, such models require training data annotated specifically with pragmatics in mind; such data is scarce in practice.

Derived models, by contrast, are based on a representation of L. They first instantiate a *base listener* L0 (intended to simulate a naïve, non-pragmatic listener). They then form a *reasoning speaker* S1, which chooses a description that causes L0 to behave correctly. Existing derived models couple hand-written grammars and hand-engineered listener models with sophisticated inference procedures. They exhibit complex behavior, but are restricted to small domains where grammar engineering is practical.

The approach we present in this paper aims to capture the best aspects of both lines of work. Like direct approaches, we use machine learning to acquire a complete grounded generation model from data, without domain knowledge in the form of a hand-written grammar or hand-engineered listener model. But like derived approaches, we use this learning to construct a *base* model, and embed it within a higher-order model that reasons about listener responses. As will be seen, this reasoning step allows the model to make use of weaker supervision than previous data-driven approaches, while exhibiting robust behavior in a variety of contexts.

Our goal is to build a derived model that scales to real-world datasets without domain engineering. Independent of the application to RG, our model also belongs to the family of neural image captioning models that have been a popular subject of recent study (Xu et al., 2015). Nevertheless, our approach appears to be:

- the first such captioning model to reason explicitly about listeners
- the first learned approach to pragmatics that requires only *non-pragmatic* training data

Following previous work, we evaluate our model on RG, though the general architecture could be applied to other tasks where pragmatics plays a core role. Using a large dataset of abstract scenes like the one shown in Figure 1, we run a series of games with humans in the role of L and our system in the role of S. We find that the descriptions generated by our model result in correct interpretation 17% more often than a recent learned baseline system. We use these experiments to explore various other aspects of computational pragmatics, including tradeoffs between adequacy and fluency, and between computational efficiency and expressive power.[1]

2 Related Work

Direct pragmatics As an example of the direct approach mentioned in the introduction, FitzGerald et al. (2013) collect a set of human-generated referring expressions about abstract representations of sets of colored blocks. Given a set of blocks to describe, their model directly learns a maximum-entropy distribution over the set of logical expressions whose denotation is the target set. Other research, focused on referring expression generation from a computer vision perspective, includes that of Mao et al. (2015) and Kazemzadeh et al. (2014).

Derived pragmatics Derived approaches, sometimes referred to as "rational speech acts" models, include those of Smith et al. (2013), Vogel et al. (2013), Golland et al. (2010), and Monroe and Potts (2015). These couple template-driven language generation with probabilistic or game-theoretic reasoning frameworks to produce contextually appropriate language: intelligent listeners reason about the behavior of reflexive speakers, and even higher-order speakers reason about these listeners. Experiments (Frank et al., 2009) show that derived approaches explain human behavior well, but both computational and representational issues restrict their application to simple reference games. They require domain-specific engineering, controlled world representations, and pragmatically annotated training data.

An extensive literature on computational pragmatics considers its application to tasks other than RG, including instruction following (Anderson et al., 1991) and discourse analysis (Jurafsky et al., 1997).

[1] Models, human annotations, and code to generate all tables and figures in this paper can be found at `http://github.com/jacobandreas/pragma`.

Representing language and the world In addition to the pragmatics literature, the approach proposed in this paper relies extensively on recently developed tools for multimodal processing of language and unstructured representations like images. These includes both image retrieval models, which select an image from a collection given a textual description (Socher et al., 2014), and neural conditional language models, which take a content representation and emit a string (Donahue et al., 2015).

3 Approach

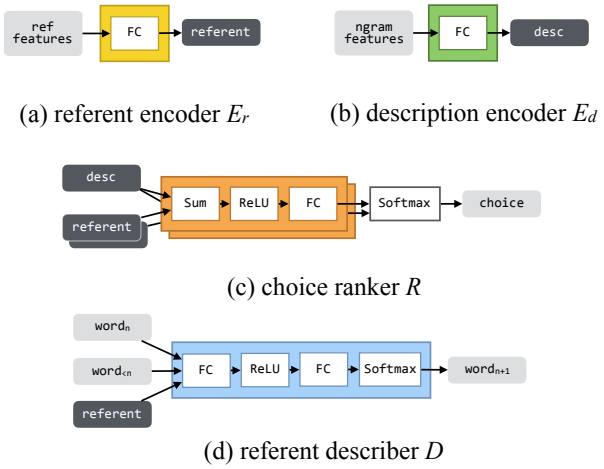
Our goal is to produce a model that can play the role of the speaker S in RG. Specifically, given a target referent (e.g. scene or object) r and a distractor r', the model must produce a description d that uniquely identifies r. For training, we have access to a set of *non-contrastively* captioned referents $\{(r_i, d_i)\}$: each training description d_i is generated for its associated referent r_i in isolation. There is no guarantee that d_i would actually serve as a good referring expression for r_i in any particular context. We must thus use the training data to ground language in referent representations, but rely on reasoning to produce pragmatics.

Our model architecture is compositional and hierarchical. We begin in Section 3.2 by describing a collection of "modules": basic computational primitives for mapping between referents, descriptions, and reference judgments, here implemented as linear operators or small neural networks. While these modules appear as substructures in neural architectures for a variety of tasks, we put them to novel use in constructing a reasoning pragmatic speaker.

Section 3.3 describes how to assemble two base models: a *literal speaker*, which maps from referents to strings, and a *literal listener*, which maps from strings to reference judgments. Section 3.4 describes how these base models are used to implement a top-level *reasoning speaker*: a learned, probabilistic, derived model of pragmatics.

3.1 Preliminaries

Formally, we take a description d to consist of a sequence of words $d_1, d_2, \ldots, d_n$, drawn from a vocabulary of known size. For encoding, we also assume access to a feature representation $f(d)$ of the sentence (for purposes of this paper, a vector of indicator features on n-grams). These two views—as a sequence of words d_i and a feature vector $f(d)$—form the basis of module interactions with language.



(a) referent encoder E_r (b) description encoder E_d

(c) choice ranker R

(d) referent describer D

Figure 2: Diagrams of modules used to construct speaker and listener models. "FC" is a fully-connected layer (a matrix multiply) and "ReLU" is a rectified linear unit. The encoder modules (a,b) map from feature representations (in gray) to embeddings (in black), while the ranker (c) and describer modules (d) respectively map from embeddings to decisions and strings.

Referent representations are similarly simple. Because the model never generates referents—only conditions on them and scores them—a vector-valued feature representation of referents suffices. Our approach is completely indifferent to the nature of this representation. While the experiments in this paper use a vector of indicator features on objects and actions present in abstract scenes (Figure 1), it would be easy to instead use pre-trained convolutional representations for referring to natural images. As with descriptions, we denote this feature representation $f(r)$ for referents.

3.2 Modules

All listener and speaker models are built from a kit of simple building blocks for working with multimodal representations of images and text:

1. a **referent encoder** E_r
2. a **description encoder** E_d
3. a **choice ranker** R
4. a **referent describer** D

These are depicted in Figure 2, and specified more formally below. All modules are parameterized by weight matrices, written with capital letters W_1, W_2, etc.; we refer to the collection of weights for all modules together as W.

Encoders The referent and description encoders produce a linear embedding of referents and descriptions in a common vector space.

$$\text{Referent encoder:} \quad E_r(r) = W_1 f(r) \tag{1}$$

$$\text{Description encoder:} \quad E_d(d) = W_2 f(d) \tag{2}$$

Choice ranker The choice ranker takes a string encoding and a collection of referent encodings, assigns a score to each (string, referent) pair, and then transforms these scores into a distribution over referents. We write $R(e_i|e_{-i}, e_d)$ for the probability of choosing i in contrast to the alternative; for example, $R(e_2|e_1, e_d)$ is the probability of answering "2" when presented with encodings e_1 and e_2.

$$\begin{aligned} s_1 &= w_3^\top \rho(W_4 e_1 + W_5 e_d) \\ s_2 &= w_3^\top \rho(W_4 e_2 + W_5 e_d) \\ R(e_i|e_{-i}, e_d) &= \frac{e^{s_i}}{e^{s_1} + e^{s_2}} \end{aligned} \tag{3}$$

(Here ρ is a rectified linear activation function.)

Referent describer The referent describer takes an image encoding and outputs a description using a (feedforward) conditional neural language model. We express this model as a distribution $p(d_{n+1}|d_n, d_{<n}, e_r)$, where d_n is an indicator feature on the last description word generated, $d_{<n}$ is a vector of indicator features on all other words previously generated, and e_r is a referent embedding. This is a "2-plus-skip-gram" model, with local positional history features, global position-independent history features, and features on the referent being described. To implement this probability distribution, we first use a multilayer perceptron to compute a vector of scores s (one s_i for each vocabulary item): $s = W_6\rho(W_7[d_n, d_{<n}, e_i])$. We then normalize these to obtain probabilities: $p_i = e^{s_i} / \sum_j e^{s_j}$. Finally, $p(d_{n+1}|d_n, d_{<n}, e_r) = p_{d_{n+1}}$.

3.3 Base models

From these building blocks, we construct a pair of base models. The first of these is a **literal listener** L0, which takes a description and a set of referents, and chooses the referent most likely to be described. This serves the same purpose as the base listener in the general derived approach described in the introduction. We additionally construct a **literal speaker** S0, which takes a referent in isolation and outputs a description. The literal speaker is used for efficient inference over the space of possible descriptions, as described in Section 3.4. L0 is, in essence, a retrieval model, and S0 is neural captioning model.

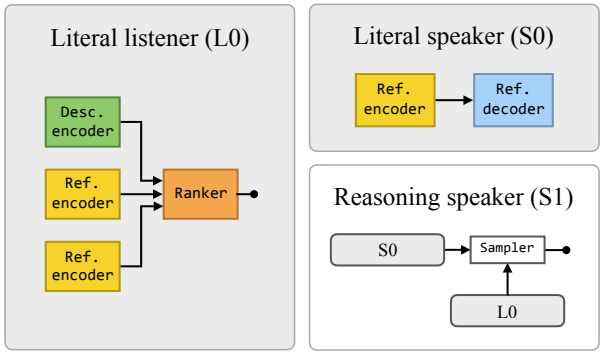


Figure 3: Schematic depictions of models. The literal listener L0 maps from descriptions and reference candidates to reference decisions. The literal speaker S0 maps directly from scenes to descriptions, ignoring context, while the reasoning speaker uses samples from S0 and scores from both L0 and S0 to produce contextually-appropriate captions.

Both of the base models are probabilistic: L0 produces a distribution over referent choices, and S0 produces a distribution over strings. They are depicted with shaded backgrounds in Figure 3.

Literal listener Given a description d and a pair of candidate referents r_1 and r_2, the literal listener embeds both referents and passes them to the ranking module, producing a distribution over choices i.

$$\begin{aligned} e_d &= E_d(d) \\ e_1 &= E_r(r_1) \\ e_2 &= E_r(r_2) \\ p_{\text{L0}}(i|d, r_1, r_2) &= R(e_i|e_{-i}, e_d) \end{aligned} \tag{4}$$

That is, $p_{\text{L0}}(1|d, r_1, r_2) = R(e_1|e_2, e_d)$ and vice-versa. This model is trained contrastively, by solving the following optimization problem:

$$\max_W \sum_j \log p_{\text{L0}}(1|d_j, r_j, r') \tag{5}$$

Here r' is a random distractor chosen uniformly from the training set. For each training example (r_i, d_i), this objective attempts to maximize the probability that the model chooses r_i as the referent of d_i over a random distractor.

This contrastive objective ensures that our approach is applicable even when there is not a naturally-occurring source of target–distractor pairs, as previous work (Golland et al., 2010; Monroe and Potts, 2015) has required. Note that this can also be viewed as a version of the loss described by Smith and Eisner (2005), where it approximates a likelihood objective that encourages L0 to prefer r_i to every other possible referent simultaneously.

Literal speaker As in the figure, the literal speaker is obtained by composing a referent encoder with a describer, as follows:

$$e = E_r(f(r))$$
$$p_{\text{S0}}(d|r) = D_d(d|e)$$

As with the listener, the literal speaker should be understood as producing a distribution over strings. It is trained by maximizing the conditional likelihood of captions in the training data:

$$\max_W \sum_i \log p_{\text{S0}}(d_i|r_i) \tag{6}$$

These base models are intended to be the minimal learned equivalents of the hand-engineered speakers and hand-written grammars employed in previous derived approaches (Golland et al., 2010). The neural encoding/decoding framework implemented by the modules in the previous subsection provides a simple way to map from referents to descriptions and descriptions to judgments without worrying too much about the details of syntax or semantics. Past work amply demonstrates that neural conditional language models are powerful enough to generate fluent and accurate (though not necessarily pragmatic) descriptions of images or structured representations (Donahue et al., 2015).

3.4 Reasoning model

As described in the introduction, the general derived approach to pragmatics constructs a base listener and then selects a description that makes it behave correctly. Since the assumption that listeners will behave deterministically is often a poor one, it is common for such derived approaches to implement *probabilistic* base listeners, and maximize the probability of correct behavior.

The neural literal listener L0 described in the preceding section is such a probabilistic listener. Given a target i and a pair of candidate referents r_1 and r_2, it is natural to specify the behavior of a reasoning speaker as simply:

$$\max_d p_{\text{L0}}(i|d, r_1, r_2) \tag{7}$$

At a first glance, the only thing necessary to implement this model is the representation of the literal listener itself. When the set of possible utterances comes from a fixed vocabulary (Vogel et al., 2013) or a grammar small enough to exhaustively enumerate (Smith et al., 2013) the operation $\max_d$ in Equation 7 is practical.

For our purposes, however, we would like the model to be capable of producing arbitrary utterances. Because the score p_{L0} is produced by a discriminative listener model, and does not factor along the words of the description, there is no dynamic program that enables efficient inference over the space of all strings.

We instead use a sampling-based optimization procedure. The key ingredient here is a good *proposal distribution* from which to sample sentences likely to be assigned high weight by the model listener. For this we turn to the literal speaker S0 described in the previous section. Recall that this speaker produces a distribution over plausible descriptions of isolated images, while ignoring pragmatic context. We can use it as a source of candidate descriptions, to be reweighted according to the expected behavior of L0. The full specification of a sampling neural reasoning speaker is as follows:

1. Draw samples $d_1, \ldots d_n \sim p_{\text{S0}}(\cdot|r_i)$.
2. Score samples: $p_k = p_{\text{L0}}(i|d_k, r_1, r_2)$.
3. Select d_k with $k = \arg\max p_k$.

While primarily to enable efficient inference, we can also use the literal speaker to serve a different purpose: "regularizing" model behavior towards choices that are adequate and fluent, rather than exploiting strange model behavior. Past work has re-

stricted the set of utterances in a way that guarantees fluency. But with an imperfect learned listener model, and a procedure that optimizes this listener's judgments directly, the speaker model might accidentally discover the kinds of pathological optima that neural classification models are known to exhibit (Goodfellow et al., 2014)—in this case, sentences that cause exactly the right response from L0, but no longer bear any resemblance to human language use. To correct this, we allow the model to consider two questions: as before, "how likely is it that a listener would interpret this sentence correctly?", but additionally "how likely is it that a speaker would produce it?"

Formally, we introduce a parameter λ that trades off between L0 and S0, and take the reasoning model score in step 2 above to be:

$$p_k = p_{\text{S0}}(d_k|r_i)^{\lambda} \cdot p_{\text{L0}}(i|d_k, r_1, r_2)^{1-\lambda} \quad (8)$$

This can be viewed as a weighted *joint* probability that a sentence is both uttered by the literal speaker and correctly interpreted by the literal listener, or alternatively in terms of Grice's conversational maxims (Grice, 1970): L0 encodes the maxims of *quality* and *relation*, ensuring that the description contains enough information for L to make the right choice, while S0 encodes the maxim of *manner*, ensuring that the description conforms with patterns of human language use. Responsibility for the maxim of *quantity* is shared: L0 ensures that the model doesn't say too little, and S0 ensures that the model doesn't say too much.

4 Evaluation

We evaluate our model on the reference game RG described in the introduction. In particular, we construct instances of RG using the Abstract Scenes Dataset introduced by Zitnick and Parikh (2013). Example scenes are shown in Figure 1 and Figure 4. The dataset contains pictures constructed by humans and described in natural language. Scene representations are available both as rendered images and as feature representations containing the identity and location of each object; as noted in Section 3.1, we use this feature set to produce our referent representation $f(r)$. This dataset was previously used for a variety of language and vision tasks (e.g. Ortiz et al. (2015), Zitnick et al. (2014)). It consists of 10,020 scenes, each annotated with up to 6 captions.

The abstract scenes dataset provides a more challenging version of RG than anything we are aware of in the existing computational pragmatics literature, which has largely used the TUNA corpus of isolated object descriptions (Gatt et al., 2007) or small synthetic datasets (Smith et al., 2013). By contrast, the abstract scenes data was generated by humans looking at complex images with numerous objects, and features grammatical errors, misspellings, and a vocabulary an order of magnitude larger than TUNA. Unlike previous work, we have no prespecified in-domain grammar, and no direct supervision of the relationship between scene features and lexemes.

We perform a human evaluation using Amazon Mechanical Turk. We begin by holding out a development set and a test set; each held-out set contains 1000 scenes and their accompanying descriptions. For each held-out set, we construct two sets of 200 paired (target, distractor) scenes: **All**, with up to four differences between paired scenes, and **Hard**, with exactly one difference between paired scenes. (We take the number of differences between scenes to be the number of objects that appear in one scene but not the other.)

We report two evaluation metrics. *Fluency* is determined by showing human raters isolated sentences, and asking them to rate linguistic quality on a scale from 1–5. *Accuracy* is success rate at RG: as in Figure 1, humans are shown two images and a model-generated description, and asked to select the image matching the description.

In the remainder of this section, we measure the tradeoff between fluency and accuracy that results from different mixtures of the base models (Section 4.1), measure the number of samples needed to obtain good performance from the reasoning listener (Section 4.2), and attempt to approximate the reasoning listener with a monolithic "compiled" listener (Section 4.3). In Section 4.4 we report final accuracies for our approach and baselines.

# samples	1	10	100	1000
Accuracy (%)	66	75	83	85

Table 1: S1 accuracy vs. number of samples.

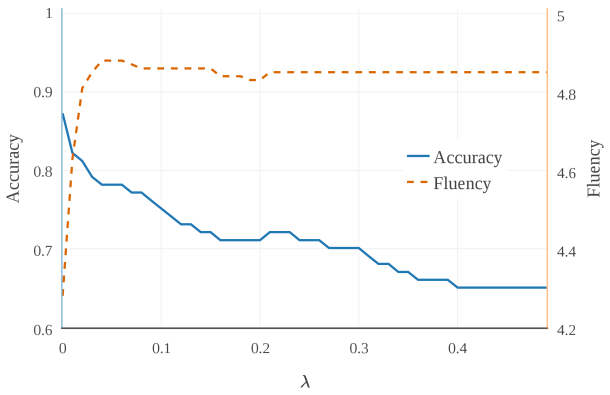


Figure 5: Tradeoff between speaker and listener models, controlled by the parameter λ in Equation 8. With $\lambda = 0$, all weight is placed on the literal listener, and the model produces highly discriminative but somewhat disfluent captions. With $\lambda = 1$, all weight is placed on the literal speaker, and the model produces fluent but generic captions.

4.1 How good are the base models?

To measure the performance of the base models, we draw 10 samples d_{jk} for a subset of 100 pairs $(r_{1,j}, r_{2,j})$ in the Dev-All set. We collect human fluency and accuracy judgments for each of the 1000 total samples. This allows us to conduct a post-hoc search over values of λ: for a range of λ, we compute the average accuracy and fluency of the highest scoring sample. By varying λ, we can view the tradeoff between accuracy and fluency that results from interpolating between the listener and speaker model—setting $\lambda = 0$ gives samples from p_{L0}, and $\lambda = 1$ gives samples from p_{S0}.

Figure 5 shows the resulting accuracy and fluency for various values of λ. It can be seen that relying entirely on the listener gives the highest accuracy but degraded fluency. However, by adding only a very small weight to the speaker model, it is possible to achieve near-perfect fluency without a substantial decrease in accuracy. Example sentences for an individual reference game are shown in Figure 5; increasing λ causes captions to become more generic. For the remaining experiments in this paper, we take $\lambda = 0.02$, finding that this gives excellent performance on both metrics.

On the development set, $\lambda = 0.02$ results in an **average fluency of 4.8** (compared to 4.8 for the literal speaker $\lambda = 1$). This high fluency can be confirmed by inspection of model samples (Figure 4). We thus focus on **accuracy** or the remainder of the evaluation.

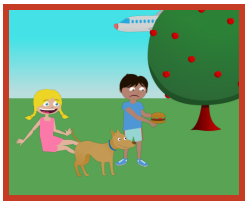
4.2 How many samples are needed?

Next we turn to the computational efficiency of the reasoning model. As in all sampling-based inference, the number of samples that must be drawn from the proposal is of critical interest—if too many samples are needed, the model will be too slow to use in practice. Having fixed $\lambda = 0.02$ in the preceding section, we measure accuracy for versions of the reasoning model that draw 1, 10, 100, and 1000 samples. Results are shown in Table 1. We find that gains continue up to 100 samples.

4.3 Is reasoning necessary?

Because they do not require complicated inference procedures, direct approaches to pragmatics typically enjoy better computational efficiency than derived ones. Having built an accurate derived speaker, can we bootstrap a more efficient direct speaker?

To explore this, we constructed a "compiled" speaker model as follows: Given reference candidates r_1 and r_2 and target t, this model produces embeddings e_1 and e_2, concatenates them together into a "contrast embedding" $[e_t, e_{-t}]$, and then feeds this whole embedding into a string decoder module. Like S0, this model generates captions without the need for discriminative rescoring; unlike S0, the contrast embedding means this model can in principle learn to produce pragmatic captions, if given access to pragmatic training data. Since no such training data exists, we train the compiled model on

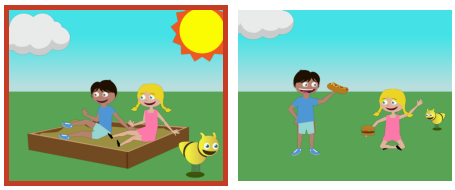


(a) target

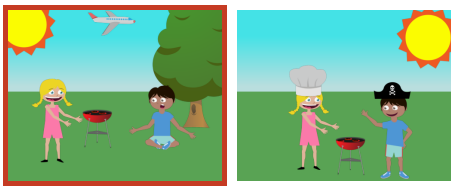
(b) distractor

(prefer L0)	`0.0`	*a hamburger on the ground*
	`0.1`	*mike is holding the burger*
(prefer S0)	`0.2`	*the airplane is in the sky*

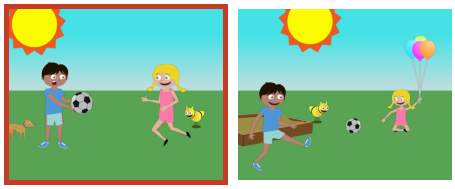
Figure 5: Captions for the same pair with varying λ. Changing λ alters both the naturalness and specificity of the output.



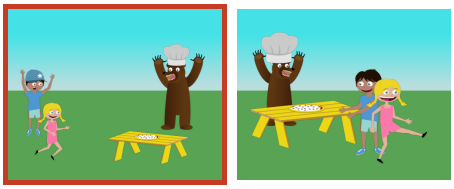
(a) *the sun is in the sky*
[contrastive]



(d) *the plane is flying in the sky*
[contrastive]



(c) *the dog is standing beside jenny*
[contrastive]



(b) *mike is wearing a chef's hat*
[non-contrastive]

Figure 4: Figure 4: Four randomly-chosen samples from our model. For each, the target image is shown on the left, the distractor image is shown on the right, and description generated by the model is shown below. All descriptions are fluent, and generally succeed in uniquely identifying the target scene, even when they do not perfectly describe it (e.g. (c)). These samples are broadly representative of the model's performance (Table 2).

	Dev acc. (%)		Test acc. (%)	
Model	All	Hard	All	Hard
Literal (S0)	66	54	64	53
Contrastive	71	54	69	58
Reasoning (S1)	**83**	**73**	**81**	**68**

Table 2: Success rates at RG on abstract scenes. "Literal" is a captioning baseline corresponding to the base speaker S0. "Contrastive" is a reimplementation of the approach of Mao et al. (2015). "Reasoning" is the model from this paper. All differences between our model and baselines are significant ($p < 0.05$, Binomial).

	# of differences				
	1	2	3	4	Mean
Literal (S0)	50	66	70	78	66 (%)
Reasoning	64	86	88	94	83
Compiled (S1)	44	72	80	80	69

Table 3: Comparison of the "compiled" pragmatic speaker model with literal and explicitly reasoning speakers. The models are evaluated on subsets of the development set, arranged by difficulty: column headings indicate the number of differences between the target and distractor scenes.

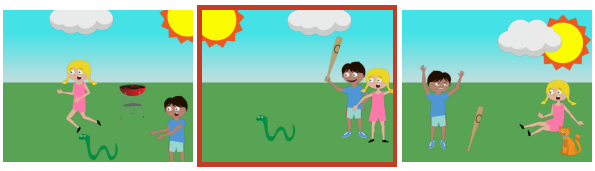
captions sampled from the reasoning speaker itself.

This model is evaluated in Table 3. While the distribution of scores is quite different from that of the base model (it improves noticeably over S0 on scenes with 2–3 differences), the overall gain is negligible (the difference in mean scores is not significant). The compiled model significantly underperforms the reasoning model. These results suggest either that the reasoning procedure is not easily approximated by a shallow neural network, or that example descriptions of randomly-sampled training pairs (which are usually easy to discriminate) do not provide a strong enough signal for a reflex learner to recover pragmatic behavior.

4.4 Final evaluation

Based on the following sections, we keep $\lambda = 0.02$ and use 100 samples to generate predictions. We evaluate on the test set, comparing this **Reasoning** model S1 to two baselines: **Literal**, an image captioning model trained normally on the abstract scene captions (corresponding to our L0), and **Contrastive**, a model trained with a soft contrastive objective, and previously used for visual referring expression generation (Mao et al., 2015).

Results are shown in Table 2. Our reasoning model outperforms both the literal baseline and previous work by a substantial margin, achieving an improvement of 17% on all pairs set and 15% on hard



(a) (b) (c)

(b vs. a) *mike is holding a baseball bat*
(b vs. c) *the snake is slithering away from mike and jenny*

Figure 6: Descriptions of the same image in different contexts. When the target scene (b) is contrasted with the left (a), the system describes a bat; when the target scene is contrasted with the right (c), the system describes a snake.

pairs.[2] Figures 4 and 6 show various representative descriptions from the model.

5 Conclusion

We have presented an approach for learning to generate pragmatic descriptions about general referents, even without training data collected in a pragmatic context. Our approach is built from a pair of simple neural base models, a listener and a speaker, and a high-level model that reasons about their outputs in order to produce pragmatic descriptions. In an evaluation on a standard referring expression game, our model's descriptions produced correct behavior in human listeners significantly more often than existing baselines.

It is generally true of existing derived approaches to pragmatics that much of the system's behavior requires hand-engineering, and generally true of direct approaches (and neural networks in particular) that training is only possible when supervision is available for the precise target task. By synthesizing these two approaches, we address both problems, obtaining pragmatic behavior without domain knowledge and without targeted training data. We believe that this general strategy of using reasoning to obtain novel contextual behavior from neural decoding models might be more broadly applied.

[2] For comparison, a model with hand-engineered pragmatic behavior—trained using a feature representation with indicators on only those objects that appear in the target image but not the distractor—produces an accuracy of 78% and 69% on all and hard development pairs respectively. In addition to performing slightly worse than our reasoning model, this alternative approach relies on the structure of scene representations and cannot be applied to more general pragmatics tasks.

References

Anne H. Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, et al. 1991. The HCRC map task corpus. *Language and speech*, 34(4):351–366.

Luciana Benotti and David Traum. 2009. A computational account of comparative implicatures for a spoken dialogue agent. In *Proceedings of the Eighth International Conference on Computational Semantics*, pages 4–17. Proceedings of the Annual Meeting of the Association for Computational Linguistics.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2625–2634.

Nicholas FitzGerald, Yoav Artzi, and Luke Zettlemoyer. 2013. Learning distributions over logical forms for referring expression generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Michael C Frank, Noah D Goodman, Peter Lai, and Joshua B Tenenbaum. 2009. Informative communication in word production and word learning. In *Proceedings of the 31st annual conference of the cognitive science society*, pages 1228–1233.

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*, pages 49–56. Proceedings of the Annual Meeting of the Association for Computational Linguistics.

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*, pages 410–419. Association for Computational Linguistics.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Herbert P Grice. 1970. Logic and conversation.

Daniel Jurafsky, Rebecca Bates, Noah Coccaro, Rachel Martin, Marie Meteer, Klaus Ries, Elizabeth Shriberg, Audreas Stolcke, Paul Taylor, Van Ess-Dykema, et al. 1997. Automatic detection of discourse structure for speech recognition and understanding. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 88–95. IEEE.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 787–798.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. 2015. Generation and comprehension of unambiguous object descriptions. *arXiv preprint arXiv:1511.02283*.

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

Luis Gilberto Mateos Ortiz, Clemens Wolff, and Mirella Lapata. 2015. Learning to interpret and describe abstract scenes. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1505–1515.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Nathaniel J Smith, Noah Goodman, and Michael Frank. 2013. Learning and using language via recursive pragmatic reasoning about other agents. In *Advances in Neural Information Processing Systems*, pages 3039–3047.

Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218.

Adam Vogel, Max Bodoia, Christopher Potts, and Daniel Jurafsky. 2013. Emergence of Gricean maxims from multi-agent decision theory. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1072–1081.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*.

C Zitnick and Devi Parikh. 2013. Bringing semantics into focus using visual abstraction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 3009–3016.

C Lawrence Zitnick, Ramakrishna Vedantam, and Devi Parikh. 2014. Adopting abstract images for semantic scene understanding.