

Learning from omission

Bill McDowell

Duolingo
Pittsburgh, PA 15206
mcdowell@duolingo.com

Noah D. Goodman

Stanford University
Stanford, CA 94305
ngoodman@stanford.edu

Abstract

Pragmatic reasoning allows humans to go beyond the literal meaning when interpreting language in context. Previous work has shown that such reasoning can improve the performance of *already-trained* language understanding systems. Here, we explore whether pragmatic reasoning *during training* can improve the quality of learned meanings. Our experiments on reference game data show that end-to-end pragmatic training produces more accurate utterance interpretation models, especially when data is sparse and language is complex.

1 Introduction

We often draw pragmatic inferences about a speaker’s intentions from what they choose to say, but also from what they choose *not* to say in context. This pragmatic reasoning arises from listeners’ inferences based on speakers’ cooperativity (Grice, 1975), and prior work has observed that such reasoning enables human children to more quickly learn word meanings (Frank and Goodman, 2014). This suggests that pragmatic reasoning might allow modern neural network models to more efficiently learn on grounded language data from cooperative reference games.

As a motivating case, consider an instance of the color reference task from Monroe et al. (2017)—shown in the first row of Table 1. In this task, a speaker communicates a target color to a listener in a context containing two distractor colors; the listener picks out the target based on what the speaker says. In the first instance from Table 1, the speaker utters “dark blue” to describe the target. Whereas “dark” and “blue” also apply to the target, they lose their informativity in the presence of the distractors, and so the speaker pragmatically opts for “dark blue”.

A listener who is learning the language from such examples might draw several inferences from the speaker’s utterance. First, under the assumption that the speaker is informative, a “literal” learner might infer that “dark blue” applies to the target shade more than the distractors. Second, a “pragmatic” learner might consider the cheaper alternatives—“dark” and “blue”—that have occurred in the presence of the same target in prior contexts, and infer that these alternative utterances must also apply to the distractors given the speaker’s failure to use them. The pragmatic learner might thus gain more semantic knowledge from the same training instances than the literal learner: pragmatic reasoning can reduce the data complexity of learning.

The pragmatic learning effects just described depend on the existence of low cost alternative utterances that the learner already knows can apply to the target object. The existence of short alternatives will be more likely when the target objects are more complex (as in row 2 of Table 1), because these objects require longer utterances (with therefore more short alternatives) to individuate. Thus, we further hypothesize that pragmatic inference will reduce data complexity especially in contexts that elicit more complex language.

In light of these arguments, we leverage the pragmatic inference described here in training neural network models to play reference games. For formal, probabilistic representations of contextual reasoning in our training objectives, we embed neural language models within pragmatic listener and speaker distributions, as specified by the Rational Speech Acts (RSA) framework (Goodman and Frank, 2016; Frank and Goodman, 2012). Pragmatic inference allows our models to learn from indirect pragmatic evidence of the sort described above, yielding better calibrated, context-sensitive models and more efficient use





	Target	Distractors	Utterance	Cheaper Alternative Utterances
1.			“dark blue”	“blue”, “dark”...
2.			“left dark blue”	“dark blue”, “left dark”, “right black”...

Table 1: Speaker utterances describing (1) colors and (2) color grids to differentiate them from distractors. A learner might draw inferences about fine-grained linguistic distinctions by explaining the speaker’s failure to use cheaper alternatives in context (e.g. they might infer that “blue” and “dark” apply to some distractors in 1). These inferences have the potential to increase in number and in strength as dimensionality of the referents and utterance complexity increase (as in 2).

of the training data. We compare pragmatic and non-pragmatic models at training and at test, while varying conditions on the training data to test hypotheses regarding the utility of pragmatic inference for learning. In particular, we show that incorporating pragmatic reasoning at training time yields improved, state-of-the-art accuracy for listener models on the color reference task from Monroe et al. (2017), and the effect demonstrated by this improvement is especially large under small training data sizes. We further introduce a new color-grid reference task and data set consisting of higher dimensional objects and more complex speaker language; we find that the effect of pragmatic listener training is even larger in this setting.

2 Related Work

Prior work has shown that neural network models trained to capture the meanings of utterances can be improved using pragmatic reasoning at test time via the RSA framework (Andreas and Klein, 2016; Monroe et al., 2017; Goodman and Frank, 2016; Frank and Goodman, 2012). For instance, Monroe et al. (2017) train context-agnostic (i.e. non-pragmatic) neural network models to learn the meanings of color utterances using a corpus of examples of the form shown in the first line of Table 1. At evaluation, they add an RSA layer on top of the trained model to draw pragmatic, context-sensitive inferences about intended color referents. Other related work explores additional approaches to create context-aware models that generate color descriptions (Meo et al., 2014), image captions (Vedantam et al., 2017), spatial references (Golland et al., 2010), and utterances in simple reference games (Andreas and Klein, 2016). Each of these shows that adding pragmatics at test time improves performance on tasks where context is relevant. Whereas this prior work showed the effectiveness of pragmatic inferences for mod-

els *trained non-pragmatically*, our current work shows that these pragmatic inferences can also inform the training procedure, providing additional gains in performance.

More similar to our work, Monroe and Potts (2015) improve model performance by incorporating pragmatic reasoning into the learning procedure for an RSA pragmatic speaker model. However, in contrast to our work, they consider a much simpler corpus, and a simple non-neural semantics. We consider richer corpora with sequential utterances and continuous referent objects that pose several algorithmic challenges which we solve using neural networks and Monte Carlo methods.

3 Approach

We compare neural nets trained pragmatically and non-pragmatically on a new color-grid reference game corpus as well as the color reference corpus from Monroe et al. (2017). In this section, we describe our tasks and models.

3.1 Reference Game Listener Tasks

The color reference game from Monroe et al. (2017) consists of rounds played between a speaker and a listener. Each round has a context of two distractors and a target color (Figure 1a). Only the speaker knows the target, and must communicate it to the listener—who must pick out the target based on the speaker’s English utterance. Similarly, each round of our new color-grid reference game contains target and distractor color-grid objects, and the speaker must communicate the target grid to the listener (Figure 1b). We train neural network models to play the listener role in these games.

3.2 Models

In both reference games, our listener models reason about a round r represented by a single train-

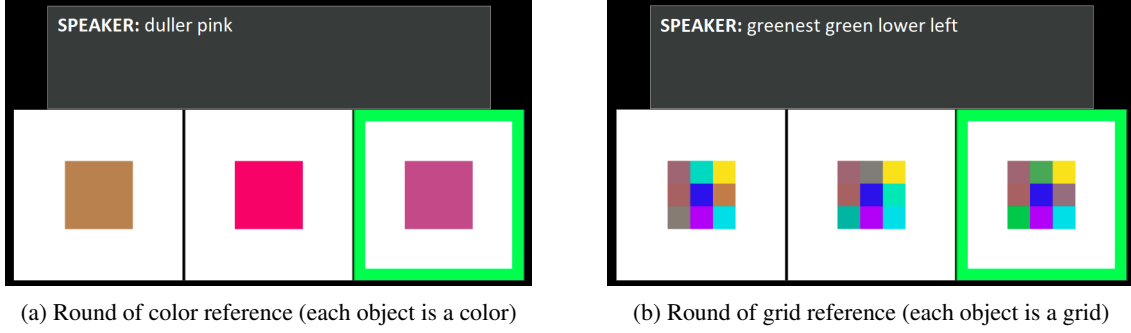


Figure 1: Rounds from the reference game tasks. These rounds consist of messages sent between a speaker and listener. The speaker communicates the target referent object (with a green border) to the listener.

ing/testing example of the form $(O^{(r)}, U^{(r)}, t^{(r)})$ where $O^{(r)}$ is the set of objects observed in the round (colors or color-grids), $U^{(r)}$ is a sequence of utterances produced by the speaker about the target (represented as a token sequence), and $t^{(r)}$ is the target index in $O^{(r)}$. The models predict the most likely referent $O_t^{(r)}$ of an utterance within a context $O^{(r)}$ according to an RSA listener distribution $l(t^{(r)} | U^{(r)}, O^{(r)})$ over targets given the utterances and a context. In pragmatic models, a nested structure allows the listener to form its beliefs about the intended referent by reasoning recursively about speaker intentions with respect to a hypothetical “literal” (non-pragmatic) listener’s interpretations of utterances. This recursive reasoning allows listener models to account for the speaker’s context-sensitive, pragmatic adjustments to the semantic content of utterances.

Formally, our pragmatic RSA model l_1 , with learnable semantic parameters θ , for target referent t , given an observed context O and speaker utterances U , is computed as:

$$\begin{aligned}
 l_1(t | U, O; \theta) &= \frac{s_1(U | t, O; \theta)p(t)}{\sum_{t'} s_1(U | t', O; \theta)p(t')} \\
 s_1(U | t, O; \theta) &= \frac{l_0(t | U, O; \theta)^\alpha p(U | O)}{\sum_{U'} l_0(t | U', O; \theta)^\alpha p(U' | O)} \\
 l_0(t | U, O; \theta) &= \frac{\mathcal{L}_{U, O_t}^\theta p(t)}{\sum_{t'} \mathcal{L}_{U, O_{t'}}^\theta p(t')}
 \end{aligned}$$

In these equations, the top-level l_1 listener model estimates the target referent by computing a pragmatic speaker s_1 and a target prior $p(t)$. Similarly, the pragmatic speaker s_1 computes an utterance distribution with respect to a literal listener l_0 , an utterance prior $p(U | O)$, and a rationality parameter α . Finally, the “literal” listener computes its expectation about the target referent from

the target prior $p(t)$ and the literal meaning, $\mathcal{L}_{U, O_t}^\theta$, which captures the extent to which utterance U applies to O_t . In both the l_0 and l_1 distributions, we take $p(t)$ to be a uniform distribution over target indices.

Literal meanings The literal meanings $\mathcal{L}_{U, O_t}^\theta$ in l_0 are computed by an LSTM (Hochreiter and Schmidhuber, 1997) that takes an input utterance and an object (color or color-grid), and produces output in the interval $(0, 1)$ representing the degree to which the utterance applies to the object (see Figure 2b). The object is represented as a single continuous vector, and is mapped into the initial hidden state of the LSTM by a dense linear layer in the case of colors, and an average-pooled, convolutional layer in the case of grids (with weights shared across the grid-cell representations described in Section 4.1.2). Given the initialized hidden state, the LSTM runs over embeddings of the tokens of an utterance. The final hidden state is passed through an affine layer, and squished by a sigmoid to produce output in $(0, 1)$. This neural net contains all learnable parameters θ of our listeners.

Utterance prior The utterance prior $p(U | O)$ in s_1 is a non-uniform distribution over sequences of English tokens—represented by a *pre-trained* LSTM language model conditioned on an input color or grid (see Figure 2a). Similar to the literal meaning LSTM, we apply a linear transformation to the input object to initialize the LSTM hidden state. Then, each step of the LSTM applies to and outputs successive tokens of an utterance. In addition, when operating over grid inputs, we apply a layer of multiplicative attention given by the “general” scoring function in (Luong et al., 2015) between the LSTM output and the convolutional

grid output before the final Softmax. This allows the language model to “attend” to individual grid cells when producing output tokens, yielding an improvement in utterance prior sample quality.

The language model is pre-trained over speaker utterances paired with targets, but the support of the distribution encoded by this LSTM is too large for the s_1 normalization term within the RSA listener to be computed efficiently. Similar to Monroe et al. (2017), we resolve this issue by taking a small set of samples from the pre-trained LSTM applied to each object in a context, to approximate $p(U | O)$, each time l_1 is computed during training and evaluation.

3.3 Learning

The full l_1 neural RSA architecture for computing pragmatic predictions over batches of input utterances and contexts is given by Algorithm 1.¹ During training, we backpropagate gradients through the full architecture, including the RSA layers, and optimize the pragmatic likelihood $\max_{\theta} \log l_1(t | U, O; \theta)$. For clarity, we can rewrite this optimization problem for a single (O, U, t) training example in the following simplified form by manipulating the RSA distributional equations from the previous section:

$$\max_{\theta} \left(\log \mathcal{L}_{U, O_t}^{\theta} - \log Z_{l_0}(U | O; \theta) - \log Z_{s_1}(t | O; \theta) - \log Z_{l_1}(U | O; \theta) \right)$$

Here, Z_{l_1} , Z_{s_0} , and Z_{l_0} are the normalization terms in the denominators of the nested RSA distributions, which we can rewrite using the log-sum-exp function (LSE) as:

$$\begin{aligned} \log Z_{l_0}(U | O; \theta) &= \text{LSE}_{t'} \left(\log \mathcal{L}_{U, O_{t'}}^{\theta} + \log p(t') \right) \\ \log Z_{s_1}(t | O; \theta) &= \text{LSE}_{U'} \left(\log l_0(t | U', O; \theta)^{\alpha} + \log p(U' | O) \right) \\ \log Z_{l_1}(U | O; \theta) &= \text{LSE}_{t'} \left(\log s_1(U | t', O; \theta) + \log p(t') \right) \end{aligned}$$

¹Note that this algorithm listing provides careful annotation of the dimensionality of various distributional tensors, which we hope might aid future research in reproducing our model implementations.

Given this representation of the optimization problem, we can see its relationship to the intuitive characterization of pragmatic learning that we gave in the introduction. First, the two terms $\log \mathcal{L}_{U, O_t}^{\theta} - \log Z_{l_0}(U | O; \theta)$ can be seen as finding the optimal *non-pragmatic* parameters; the first $\log \mathcal{L}_{U, O_t}^{\theta}$ term upweights the model’s estimate of the literal applicability of the observed U to its intended target referent, and the $-\log Z_{l_0}(U | O; \theta)$ term maximizes the margin between this estimate and the applicability of U to the contextual distractors.² Next, the $-\log Z_{s_1}(t | O; \theta)$ term makes pragmatic adjustments to the parameter estimates by enforcing a margin between the l_0 predictions given by low cost alternatives U' and the observed utterance U on a referent object t . The enforcement of this margin pushes $\mathcal{L}_{U', O_{t'}}^{\theta}$ upward for distractors t' , simulating the pragmatic reasoning described in the introduction, and drawing additional information about the low cost alternative utterances from their omission in context. Finally, the $-\log Z_{l_1}(U | O; \theta)$ term enforces a margin between the speaker prediction $s_1(U | t, O; \theta)$ and predictions on the true utterance U given distractors $O_{t'}$. This ensures that the true utterance is down-weighted on distractor objects following the speaker’s pragmatic adjustments, such that our l_1 listener predictions are well-calibrated with respect to the s_1 distribution’s cost-sensitive adjustments learned through $-\log Z_{s_1}(t | O; \theta)$.

4 Experiments

We investigate the value of pragmatic training by estimating the parameters θ in the RSA “literal meaning” function \mathcal{L}^{θ} for l_1 (pragmatic) and l_0 (non-pragmatic) distributions according to the maximal likelihood of the training data for the color and grid reference tasks. We then evaluate meanings \mathcal{L}^{θ} from each training procedure using pragmatic l_1 inference (and non-pragmatic l_0 inference, for completeness). We perform this comparison repeatedly to evaluate the value of pragmatics at training and test under various data conditions. In particular, we evaluate the hypotheses that (1) the pragmatic inferences enabled by the l_1 training will reduce sample complexity, leading to more accurate meaning functions especially under small data sizes, and (2) the effectiveness

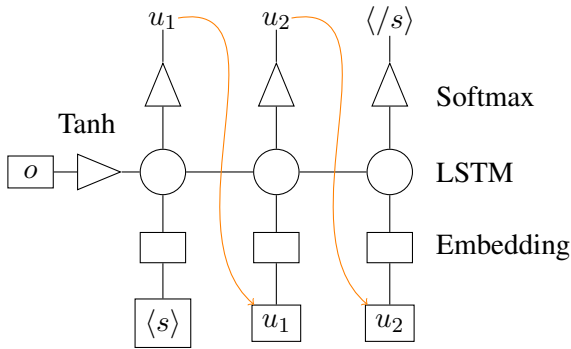
²Here, we think informally of a margin by considering the LSE as an approximation of max.

Algorithm 1 RSA pragmatic listener (l_1) neural network forward computation. The I_1 function is applied to batches of input utterances and observed contexts, and produces batches of distributions over objects in the contexts, representing the listener’s beliefs about intended utterance referents.

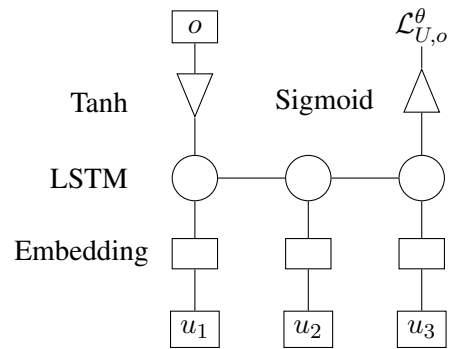
```

1:  $b \leftarrow$  data batch size
2:  $l \leftarrow$  maximum utterance length
3:  $k \leftarrow$  number of objects per context (i.e. colors or color-grids)
4:  $d \leftarrow$  dimension of each object
5:  $u \leftarrow$  number utterances to sample per object in context to make speaker distribution supports
6:  $z \leftarrow ku + 1$  number of utterances in each support including input utterance
7:  $s_0 \leftarrow$  pre-trained LSTM language model (Figure 2a)
8:  $\mathbf{L} \leftarrow$  LSTM meaning function architecture (Figure 2b)
9: function  $I_1$ (utterances  $U \in \mathbb{R}^{b \times l}$ , observations  $O \in \mathbb{R}^{b \times k \times d}$ )
10:    $P_t \leftarrow (\mathbb{S} = (0, \dots, k-1)^b, \mathbb{P} = \mathbf{1}^{b \times k}/k)$  ▷ batch of uniform target priors of size  $b \times k$ 
11:    $S_1 \leftarrow s_1(\mathbb{S}[P_t], O, U)^T$  ▷ speaker utterance distributions of size  $b \times z \times k$ 
12: return Normalize-Rows( $S_1 \cdot \text{Repeat}(\mathbb{P}[P_t], z)$ )[ $U$ ] ▷ target distributions conditioned on utterances in  $U$ 
13:
14: function  $s_1$ (possible targets  $T \in \mathbb{R}^{b \times k}$ , observations  $O \in \mathbb{R}^{b \times k \times d}$ , fixed input utterances  $U \in \mathbb{R}^{b \times l}$ )
15:    $P_{utt} \leftarrow \text{SAMPLE-UTTERANCE-PRIORS}(U, O)$  ▷ sample batch of utterance priors of size  $b \times z$ 
16:    $L_0 \leftarrow \mathbf{l}_0(\mathbb{S}[P_u], O)^T$  ▷ batch of distributions over targets of size  $b \times k \times z$ 
17: return Normalize-Rows( $L_0 \cdot \text{Repeat}(\mathbb{P}[P_u], k)$ ) ▷ speaker utterance distributions of size  $b \times k \times z$ 
18:
19: function  $\mathbf{l}_0$ (possible utterances  $U \in \mathbb{R}^{b \times z \times l}$ , observations  $O \in \mathbb{R}^{b \times k \times d}$ )
20:    $P_t \leftarrow (\mathbb{S} = (0, \dots, k-1)^b, \mathbb{P} = \mathbf{1}^{b \times k}/k)$  ▷ batch of uniform target priors of size  $b \times k$ 
21:    $\mathcal{L} \leftarrow \text{COMPUTE-MEANINGS}(U, \mathbb{S}[P_t], O)$  ▷ batch of meaning matrices of size  $b \times z \times k$ 
22: return Normalize-Rows( $\mathcal{L} \cdot \text{Repeat}(\mathbb{P}[P_t], z)$ ) ▷ batch of distributions over targets of size  $b \times z \times k$ 
23:
24: function SAMPLE-UTTERANCE-PRIORS(fixed input utterances  $U \in \mathbb{R}^{b \times l}$ ,  $O \in \mathbb{R}^{b \times k \times d}$ )
25:    $P_{utt} \leftarrow (\mathbb{S} = \mathbf{0}^{b \times z \times l}, \mathbb{P} = \frac{\mathbf{1}^{b \times z}}{z})$  ▷ initialize supports and probabilities in utterance prior tensor
26:   for  $i = 1$  to  $b$  do ▷ for each round in batch
27:     for  $j = 1$  to  $k$  do ▷ for each object in a round
28:       Sample  $\mathbf{u}^{(v)}$  from  $s_0(\langle s \rangle, O[i, j])$  for  $v = 1, \dots, u$  ▷ sample utterances for object  $O[i, j]$ 
29:        $\mathbb{S}[P_{utt}][i, (j-1)u : ju] \leftarrow \mathbf{u}$  ▷ add sampled utterance to supports
30:        $\mathbb{S}[P_{utt}][:, ku, :] \leftarrow U$  ▷ add input utterances to supports
31: return  $P_{utt}$ 
32:
33: function COMPUTE-MEANINGS( $U \in \mathbb{R}^{b \times z \times l}, T \in \mathbb{R}^{b \times k}, O \in \mathbb{R}^{b \times k \times d}$ )
34:    $\mathcal{L} \leftarrow \mathbf{0}^{b \times z \times k}$  ▷ initialize meaning tensor to be filled
35:   for  $i = 1$  to  $b$  do ▷ for each round in batch
36:      $(U_i, T_i) \leftarrow$  cartesian product of utterances in  $U[i]$  and targets in  $T[i]$ 
37:      $\mathcal{L}[i] \leftarrow \text{Reshape}(\mathbf{L}(U_i, O[i, T_i]), z, k)$  ▷ degrees to which each utterance applies to each object
38: return  $\mathcal{L}$  ▷ batch of meaning matrices (one per example context)

```



(a) Architecture for the language model from which we sample for utterance priors. We recursively sample utterance tokens from a dense Softmax layer applied to the LSTM output.

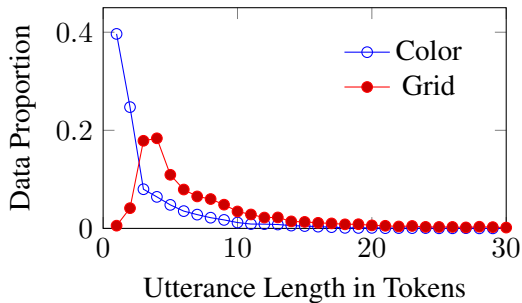


(b) Architecture for computing the literal meaning $\mathcal{L}_{U,o}^\theta$ within RSA. A dense sigmoid layer computes the output meaning $\mathcal{L}_{U,o}^\theta \in (0, 1)$ based on the final state of the LSTM that was applied to u and o .

Figure 2: Neural networks for (a) the speaker language model used to construct utterance priors, and (b) the meaning function $\mathcal{L}_{U,o}^\theta$ within the RSA listener distributions (diagram style inspired by Monroe et al. (2017)). Both architectures apply a tanh layer to an input object o (a grid or color), and use the result as the initial hidden state of an LSTM layer. In each case, the LSTM operates over embeddings of tokens u_1, u_2, \dots from utterance U .

Model	Color Dev	Color Test	Grid Dev	Grid Test
l_0 training, l_0 test	0.8455 ± 0.0011	0.8656 ± 0.0012	0.5714 ± 0.0068	0.5443 ± 0.0122
l_0 training, l_1 test	0.8472 ± 0.0013	0.8671 ± 0.0017	0.5694 ± 0.0075	0.5455 ± 0.0123
l_1 training, l_1 test	0.8587 ± 0.0008	0.8771 ± 0.0008	0.6329 ± 0.0045	0.6200 ± 0.0063
Monroe et al. (2017)	0.8484	0.8698		

Table 2: Listener accuracies on the color and grid data. All accuracies are reported with \pm SE.



(a) Length distributions of speaker utterances for colors and grids. Grids usually have longer descriptions.

Color Utterances	Grid Utterances
blue	top left blue
purple	purple top left
green	purple top right

(b) Top three most frequent speaker utterances in the color and grid data. Top color descriptions are single words. Multi-word utterances like “dark green” are less frequent. Top grid utterances tend to specify colors and locations.

Data	Color Accuracy	Grid Accuracy
Full	0.9003	0.9318
Close	0.8333	0.9024
Split	0.8970	0.9291
Far	0.9696	0.9642

(c) Human accuracies on full color and grid data, and in close, split, and far conditions. Grid accuracy is higher than color accuracy, possibly because there are more properties for speakers to describe when referring to grids.

Figure 3: Comparison of the color and grid data sets

of the l_1 training over l_0 training will increase on a more difficult reference game task containing higher-dimensional objects and utterances—i.e. pragmatic training will help more in the grids task than in the colors task.

4.1 Data

4.1.1 Color Reference

For the color reference task, we use the data collected by Monroe et al. (2017) from human play on the color reference task through Amazon Mechanical Turk using the framework of Hawkins

(2015). Each game consists of 50 rounds played by a human speaker and listener. In each round, the speaker describes a target color surrounded by a context of two other distractor colors, and a listener clicks on the targets based on the speaker’s description (see Figure 1a). The resulting data consists of 46,994 rounds across 948 games, where the colors of some rounds are sampled to be more likely to require pragmatic reasoning than others. In particular, 15,516 trials are *close* with both distractors within a small distance to the target color in RGB space, 15,782 are *far* with both distractors far from the target, and 15,693 are *split* with one distractor near the target and one far from the target. For model development, we use the train/dev/test split from Monroe et al. (2017) with 15,665 training, 15,670 dev, and 15,659 test rounds.

Within our models, we represent color objects using a 3-dimensional CIELAB color space—normalized so that the values of each dimension are in $[-1, 1]$. Our use of the CIELAB color space departs from prior work on the color data which used a 54-dimensional Fourier space (Monroe et al., 2017, 2016; Zhang and Lu, 2002). We found that both the CIELAB and Fourier spaces gave similar model performance, so we chose the CIELAB space due to its smaller dimensionality. Our speaker utterances are represented as sequences of cleaned English token strings. Following Monroe et al. (2017), we preprocess the tokens by lowercasing, splitting off punctuation, and replacing tokens that appear only once with *[unk]*. In the color data, we also follow the prior work and split off *-er*, *-est*, and *-ish* suffixes. Whereas the prior work concatenated speaker messages into a single utterance without limit, we limit the full sequence length to 40 tokens for efficiency.

4.1.2 Grid Reference

Because initial simulations suggested that pragmatic training would be more valuable in more complex domains (where data sparsity is a greater issue), we collected a new data set from human

play on the color-grid reference task described in Section 3.1. Data was collected on Amazon Mechanical Turk using an open source framework for collaborative games (Hawkins, 2015). Each game consists of 60 rounds played between a human speaker and listener, where the speaker describes a target grid in the presence of two distractor grids (see Figure 1b), resulting in a data set of 10,666 rounds spread across 197 games.³ Each round consists of three 3×3 grid objects, with the grid colors at each cell location sampled according to the same *close*, *split*, and *far* conditions as the in color reference data—yielding 3,575 *close*, 3,549 *far*, and 3,542 *split* rounds. We also varied the number of cells that differ between objects in a round from 1 to 9. As shown in Figure 3, these grid trials result in more complex speaker utterances than the color data. We partitioned this data into 158 train, 21 dev, 18 test games containing 8,453 training, 1,236 dev, and 977 test rounds. In our models, we represent a single color-grid object from the data as a concatenation of 9 vectors representing the 9 grid cells. Each of the 9 cell vectors consists of the normalized CIELAB representation used in the color data appended to a one-hot vector representing the position of the cell within the grid. For speaker utterances, we use the same representation as in the color data, except that we do not split off the *-er*, *-est*, and *-ish* endings.

4.2 Model Training Details

We implement our models in PyTorch (Paszke et al., 2017), and train them using the Adam variant of stochastic gradient descent (Kingma and Ba, 2015) with default parameters $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$. We train with early-stopping based on dev set log-likelihood (for speaker) or accuracy (for listener) model evaluations. Before training our listeners, we pre-train an LSTM language model to provide samples for the utterance priors on target colors paired with speaker utterances of length at most 12 on examples where human listeners picked the correct color. We follow Monroe et al. (2017) for language model hyper-parameters, with embedding and LSTM layers of size 100. Also following this prior work, we use a learning rate of 0.004, batch size 128, and apply 0.5 dropout to each layer. We train for 7,000 iterations, evaluating the model’s

³The grid data and our model implementations are available at <https://github.com/forkunited/ltprg>.

accuracy on the dev set every 100 iterations. We pick the model with the best dev set log-likelihood from evaluations at 100 iteration intervals.

To train and compare various listeners, we optimize likelihoods under non-pragmatic l_0 and pragmatic l_1 with a literal meaning function computed by the LSTM architecture described in Section 3.2, sampling new utterance priors for each mini-batch from our pre-trained language model applied to the round’s three colors for use within the s_1 module of RSA (see Algorithm 1). We draw 30 samples per round (10 per color or grid) at a maximum length of 12. We generally use speaker rationality $\alpha = 8.0$ based on dev set tuning, and we follow Monroe et al. (2017) for other hyper-parameters—with embedding size of 100 and LSTM size of 100 in our meaning functions. Also following this prior work, we allow the LSTM to be bidirectional with learning rate of 0.005, batch size 128, and gradient clipping at 5.0. We train listeners for 10,000 iterations on the color data and 15,000 iterations on grid data, evaluating dev set accuracy every 500 iterations. We pick the model with the best accuracy from those evaluated at 500 iteration intervals.

4.3 Results

4.3.1 Color Reference

The accuracies of color target predictions by l_0 and l_1 models under both l_0 and l_1 training are shown in the left columns of Table 2. For robustness, average accuracies and standard errors were computed by repeatedly retraining and evaluating with different weight initializations and training data orderings using 4 different random seeds. The results in the top left panel of Table 2 show that l_1 pragmatic training coupled with l_1 pragmatic evaluation gives the best average accuracies. The previously studied l_1 pragmatic usage with l_0 non-pragmatic training is next best. These results provide evidence that literal meanings estimated through pragmatic training are better calibrated for pragmatic usage than meanings estimated through non-pragmatic l_0 training. Furthermore, relative to state-of-the-art in Monroe et al. (2017), Table 2 shows that our pragmatically trained model yields improved accuracy over their best “blended” pragmatic L_e model which computed predictions based on the product of two separate non-pragmatically trained models.

The effect sizes are small for the pragmatic to

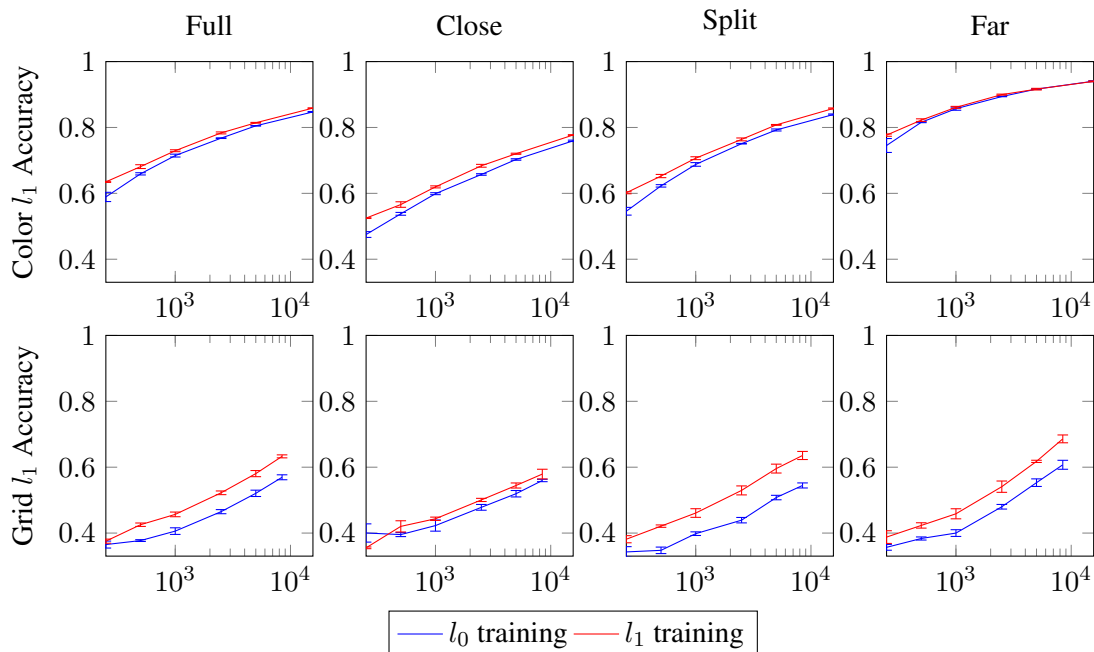


Figure 4: Listener accuracies on the color and grid dev data for models learned under training data subsets of various sizes (given by the horizontal axes). The separate plots give accuracies over the full data and the *close*, *split*, and *far* conditions.

non-pragmatic comparisons when training on the full color data (though approaching the ceiling 0.9108 human accuracy), but we hypothesized that the effect of pragmatic training would increase for training with smaller data sizes (as motivated by arguments in the introduction). To test this, we trained the listener models on smaller subsets of the training data, and evaluated accuracy. As shown by the top left plot of Figure 4, pragmatic training results in a larger gain in accuracy when less data is available for training.

Lastly, we also considered the effect of pragmatic training under the varying *close*, *split*, and *far* data conditions. As shown in the three plots at the right of the top row of Figure 4, the effect of l_1 training over l_0 is especially pronounced for inferences on *close* and *split* data conditions where the target is more similar to the distractors, and the language is more context-dependent. This makes sense, as these conditions contain examples where the pragmatic, cost-sensitive adjustments to the learned meanings would be the most necessary.

4.3.2 Grid Reference

For the more complex grid reference task, the listener accuracies in the right columns of Table 2 show an even larger gain from pragmatic l_1 training, and no gain is seen for pragmatic evaluation

with *non-pragmatic* training. This result is consistent with the hypothesis motivated by arguments in the introduction that pragmatic training should be more effective in contexts containing targets and distractors for which many low-cost alternative utterance are applicable. Furthermore, the grid-reference data-complexity exploration in the bottom row of Figure 4 shows that this improvement given by pragmatic training remains large across data sizes; the exception is the smallest amount of training data under the most difficult *close* condition, where the language is so sparse that meanings may be difficult to estimate, even with pragmatic adjustments. Altogether, these results suggest that pragmatic training helps with an intermediate amount of data relative to the domain complexity—with too little data, pragmatics has no signal to work with, but with too much data, the indirect evidence provided by pragmatics is less necessary. Since real-world linguistic contexts are more complex than either of our experimental domains, we hypothesize that they often fit into this intermediate data regime.

4.3.3 Literal Meaning Comparisons

To improve our understanding of the quantitative results, we also investigate qualitative differences between meaning functions \mathcal{L}^θ estimated under

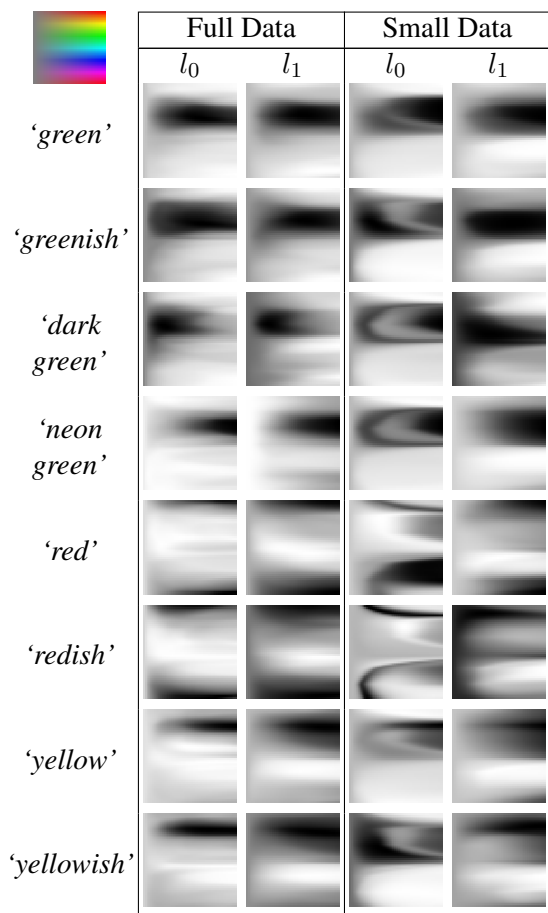


Table 3: Extensions of various utterances according to the literal meaning functions (\mathcal{L}) from listener models l_0 and l_1 learned on full (left) and smaller (250 example) subsets (right) of the color data. Each cell shows a learned color utterance extension in the *Hue* \times *Saturation* color space depicted at the top left, with the degree of darkness corresponding to the learned values of $\mathcal{L}_{U,O}^\theta$ (given by the architecture in Figure 2b) for an utterance U on colors O for regions of color space.

l_0 and l_1 on the color reference task. Table 3 shows representations of these meaning functions for several utterances. For each utterance U , we plot the extension \mathcal{L}_U^θ estimated under l_0 and l_1 , with the darkness of a pixel at c representing $\mathcal{L}_{U,c}^\theta$ —the degree to which an utterance U applies to a color c within a *Hue* \times *Saturation* color space. In these plots, the larger areas of medium gray shades for l_1 extensions suggest that the pragmatic training yields more permissive interpretations for a given utterance. This makes sense, as pragmatics allows looser meanings to be effectively tightened at interpretation time. Furthermore, the meanings learned by the l_1 also have lower curvature across the color space, consistent with a view of pragmatics as providing a regularizer (Section 3.3)—

preventing overfitting. This view is further supported by the plots on the right-hand side of Table 3, which show that the meanings learned by l_0 from smaller amounts of training data tend to overfit to idiosyncratic regions of the color space, whereas the pragmatic l_1 training tends to smooth out these irregularities. These qualitative observations are also consistent with the data complexity results shown in Figure 4, where the l_1 training gives an especially large improvement over l_0 for small data sizes.

5 Conclusion

Our experiments provide evidence that using pragmatic reasoning during training can yield improved neural semantics models. This was true in the existing color reference corpus, where we achieved state-of-the-art results, and even more so in the new color-grid corpus. We thus found that pragmatic training is more effective when data is relatively sparse and the domain yields complex, high-cost utterances and low-cost omissions over which pragmatic inferences might proceed.

Future work should provide further exploration of the data regime in which pragmatic learning is most beneficial and its correspondence to real-world language use. This might include scaling with linguistic complexity and properties of referents. In particular, the argument in our introduction suggests that especially frequent objects and low-cost utterances are the seed from which pragmatic inference can proceed over more complex language and infrequent objects. This asymmetry in object reference rates is expected for long-tail, real-world regimes consistent with Zipf’s law (Zipf, 1949).

Overall, we have shown that pragmatic reasoning regarding alternative utterances provides a useful inductive bias for learning in grounded language understanding systems—leveraging inferences over what speakers choose *not* to say to reduce the data complexity of learning.

Acknowledgments

We thank Leon Bergen for guidance in setting the up the initial versions of the pragmatic learning models, Katherine Hermann for help with some initial experiments on the color reference task, and Sahil Chopra for collecting a small batch of pilot data for the color grid task.

References

- 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)*, pages 1173–1182.
- 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*, 75:80–96.
- 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 on Natural Language Processing (EMNLP)*, pages 410–419.
- Noah D. Goodman and Michael C. Frank. 2016. Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*, 20(11):818–829.
- H Paul Grice. 1975. Logic and conversation. 1975, pages 41–58.
- Robert X. D. Hawkins. 2015. Conducting real-time multiplayer experiments on the web. *Behavior Research Methods*, 47(4):966–976.
- Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. *ICLR*.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- 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)*, pages 107–115.
- Will Monroe, Noah D Goodman, and Christopher Potts. 2016. Learning to generate compositional color descriptions. In *Proc. EMNLP*.
- Will Monroe, Robert XD 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*.
- Will Monroe and Christopher Potts. 2015. Learning in the Rational Speech Acts model. In *Proceedings of the 20th Amsterdam Colloquium*, pages 1–12.
- Adam Paszke, Sam Gross, and Soumith Chintala. 2017. Pytorch.
- Ramakrishna Vedantam, Samy Bengio, Kevin Murphy, Devi Parikh, and Gal Chechik. 2017. Context-aware captions from context-agnostic supervision. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1070–1079.
- Dengsheng Zhang and Guojun Lu. 2002. Shape-based image retrieval using generic fourier descriptor. *Signal Processing: Image Communication*, 17(10):825–848.
- George Kingsley Zipf. 1949. Human behavior and the principle of least effort. *Oxford, England: Addison-Wesley Press*.