

Learning in the Rational Speech Acts Model

Will Monroe and Christopher Potts

Stanford University, California, U.S.A.
wmonroe4@cs.stanford.edu, cgpotts@stanford.edu

Abstract

The Rational Speech Acts (RSA) model treats language use as a recursive process in which probabilistic speaker and listener agents reason about each other’s intentions to enrich the literal semantics of their language along broadly Gricean lines. RSA has been shown to capture many kinds of conversational implicature, but it has been criticized as an unrealistic model of speakers, and it has so far required the manual specification of a semantic lexicon, preventing its use in natural language processing applications that learn lexical knowledge from data. We address these concerns by showing how to define and optimize a trained statistical classifier that uses the intermediate agents of RSA as hidden layers of representation forming a non-linear activation function. This treatment opens up new application domains and new possibilities for learning effectively from data. We validate the model on a referential expression generation task, showing that the best performance is achieved by incorporating features approximating well-established insights about natural language generation into RSA.

1 Pragmatic language use

In the Gricean view of language use [18], people are rational agents who are able to communicate efficiently and effectively by reasoning in terms of shared communicative goals, the costs of production, prior expectations, and others’ belief states. The Rational Speech Acts (RSA) model [11] is a recent Bayesian reconstruction of these core Gricean ideas. RSA and its extensions have been shown to capture many kinds of conversational implicature and to closely model psycholinguistic data from children and adults [7, 2, 23, 30, 33].

Both Grice’s theories and RSA have been criticized for predicting that people are more rational than they actually are. These criticisms have been especially forceful in the context of language production. It seems that speakers often fall short: their utterances are longer than they need to be, underinformative, unintentionally ambiguous, obscure, and so forth [1, 10, 16, 24, 28, 29]. RSA can incorporate notions of bounded rationality [4, 13, 20], but it still sharply contrasts with views in the tradition of [6], in which speaker agents rely on heuristics and shortcuts to try to accurately describe the world while managing the cognitive demands of language production.

In this paper, we offer a substantially different perspective on 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; [34, 15]), showing that it improves on heuristic-driven models and pure RSA by synthesizing the best aspects of both.

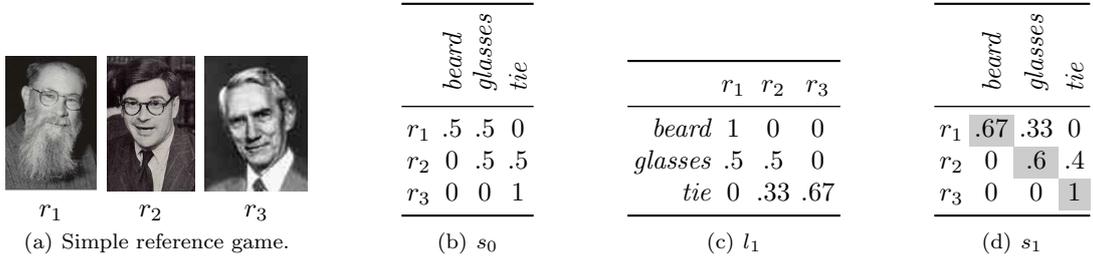


Figure 1: Ambiguity avoidance in RSA.

2 RSA as a speaker model

RSA is a descendent of the signaling systems of [25] and draws on ideas from iterated best response (IBR) models [13, 20], iterated cautious response (ICR) models [21], and cognitive hierarchies [4] (see also [17, 31]). 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.

For instance, suppose a speaker and listener are playing a reference game in the context of the images in Figure 1(a). The speaker S has been privately assigned referent r_1 and must send a message that conveys this to the listener. A literal speaker would make a random choice between *beard* and *glasses*. However, if S places itself in the role of a listener L receiving these messages, then S will see that *glasses* creates uncertainty about the referent whereas *beard* does not, and 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. It assumes a set of messages M , a set of states T , a prior probability distribution P over states T , and a cost function C mapping messages to real numbers. The semantics of messages is defined by a lexicon \mathcal{L} , where $\mathcal{L}(m, t) = 1$ if m is true of t and 0 otherwise. The agents are then defined as follows:

$$s_0(m | t, \mathcal{L}) \propto \exp(\lambda(\log \mathcal{L}(m, t) - C(m))) \quad (1)$$

$$l_1(t | m, \mathcal{L}) \propto s_0(m | t, \mathcal{L})P(t) \quad (2)$$

$$s_1(m | t, \mathcal{L}) \propto \exp(\lambda(\log l_1(t | m, \mathcal{L}) - C(m))) \quad (3)$$

The model that is the starting point for our contribution in this paper is the pragmatic speaker s_1 . It reasons not about the semantics directly but rather about a pragmatic listener l_1 reasoning about a literal speaker s_0 . The strength of this pragmatic reasoning is partly governed by the temperature parameter λ , with higher values leading to more aggressive pragmatic reasoning.

Figure 1 tracks the RSA computations for the reference game in Figure 1(a). Here, the message costs C 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_1 are high, since it derives the unambiguous system highlighted in gray.

The task we seek to model is a language generation task, so we present RSA from a speaker-centric perspective. It has been explored more fully from a listener perspective. In that formulation, the model begins with a literal listener reasoning only in terms of the lexicon \mathcal{L} and state priors. Models of this general form have been shown to capture a wide range of pragmatic behaviors [2, 12, 22, 23, 30] and to increase success in task-oriented dialogues [35, 36].

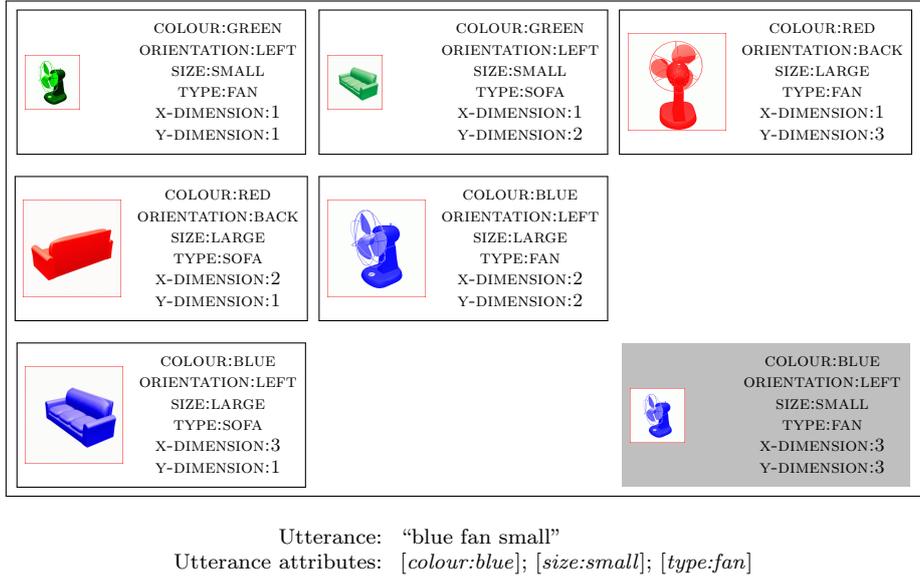


Figure 2: Example item from the TUNA corpus. Target is in gray.

RSA has been criticized on the grounds that it predicts unrealistic speaker behavior [16]. For instance, in Figure 1, we confined our agents to a 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 r_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, and cognitive pressures that make optimal descriptions impossible. One of our central objectives is to allow these factors to guide the core RSA calculation.

3 The TUNA corpus

In Section 6, we evaluate RSA and learned RSA in the TUNA corpus [34, 15], a widely used resource for developing and testing models of natural language generation. We introduce the corpus now because doing so helps clarify the learning task faced by our model, which we define in the next section.

In the TUNA corpus, participants were assigned a target referent or referents in the context of seven other distractors and asked to describe the target(s). 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 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 for reference.)

The task we address is *attribute selection*: reproducing the multiset of attributes in the message produced in each context. Thus, for Figure 2, we would aim to produce {[size:small],

$[colour:blue], [type:fan]\}$. This is less demanding than full natural language generation, since it factors out all morphosyntactic phenomena. Section 6 provides additional details on the nature of this evaluation.

4 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 [17], but differs from theirs in that we directly optimize the performance of the pragmatic speaker in training, whereas [17] 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 a literal listener.

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 [8, 26]. We define our feature representation function ϕ very generally as a map from state–utterance–context triples $\langle t, m, c \rangle$ 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 $\phi(t, m, c)$ 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 m . Thus, the representation of the target entity in Figure 2 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.

Literal speaker. Learned RSA is built on top of a *log-linear model*, standard in the machine learning literature and widely applied to classification tasks [19, 27].

$$S_0(m \mid t, c; \theta) \propto \exp(\theta^T \phi(t, m, c)) \quad (4)$$

This model serves as our literal speaker, analogous to s_0 in (1). The lexicon of this model is embedded in the *parameters* (or *weights*) θ . Intuitively, θ is the direction in feature representation space that the literal 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 **Training**, 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:

$$J_{S_0}(t, m, c, \theta) = \log S_0(m | t, c; \theta) \quad (5)$$

The gradient of this log-likelihood is

$$\begin{aligned} \frac{\partial J_{S_0}}{\partial \theta} &= \phi(t, m, c) - \frac{1}{\sum_{m'} \exp(\theta^T \phi(t, m', c))} \sum_{m'} \exp(\theta^T \phi(t, m', c)) \phi(t, m', c) \\ &= \phi(t, m, c) - \sum_{m'} S_0(m' | t, c; \theta) \phi(t, m', c) \\ &= \phi(t, m, c) - \mathbb{E}_{m' \sim S_0(\cdot | t, c; \theta)} [\phi(t, m', c)] \end{aligned} \quad (6)$$

where the first two equations can be derived by expanding the proportionality constant in the definition of S_0 .

Pragmatic speaker. We now define a pragmatic listener L_1 and a pragmatic speaker S_1 . We will show experimentally (Section 6) that the learned pragmatic speaker S_1 agrees better with human speakers on a referential expression generation task than either the literal speaker S_0 or the pure RSA speaker s_1 .

The parameters for L_1 and S_1 are still the parameters of the literal speaker S_0 ; we wish to update them to maximize the performance of S_1 , the agent that acts according to $S_1(m | t, c; \theta)$, where

$$S_1(m | t, c; \theta) \propto L_1(t | m, c; \theta) \quad (7)$$

$$L_1(t | m, c; \theta) \propto S_0(m | t, c; \theta) \quad (8)$$

This corresponds to the simplest case of RSA in which $\lambda = 1$ and message costs and state priors are uniform: $s_1(m | t, \mathcal{L}) \propto l_1(t | m, \mathcal{L}) \propto s_0(m | t, \mathcal{L})$.

In optimizing the performance of the pragmatic speaker S_1 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 [32]. 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:

$$J_{S_1}(t, m, c, \theta) = \log S_1(m | t, c; \theta) \quad (9)$$

$$J_{L_1}(t, m, c, \theta) = \log L_1(t | m, c; \theta) \quad (10)$$

The log-likelihood of each agent has the same form as the log-likelihood of the literal 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 (6) above, the gradient of these log-likelihoods can thus

be shown to have the same form as the gradient of the literal speaker, but with the gradient of the next lower agent substituted for the feature values:

$$\frac{\partial J_{S_1}}{\partial \theta} = \frac{\partial J_{L_1}}{\partial \theta}(t, m, c, \theta) - \mathbb{E}_{m' \sim S_1(\cdot | t, c; \theta)} \left[\frac{\partial J_{L_1}}{\partial \theta}(t, m', c, \theta) \right] \quad (11)$$

$$\frac{\partial J_{L_1}}{\partial \theta} = \frac{\partial J_{S_0}}{\partial \theta}(t, m, c, \theta) - \mathbb{E}_{t' \sim L_1(\cdot | m, c; \theta)} \left[\frac{\partial J_{S_0}}{\partial \theta}(t', m, c, \theta) \right] \quad (12)$$

The value J_{S_0} in (12) is as defined in (5).

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, 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 [5]. 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 $\langle t_i, m_i, c_i \rangle$, this log posterior is:

$$J(\theta) = -\frac{M}{2} \ell \|\theta\|^2 + \sum_{i=1}^M \log S_1(m_i | t_i, c_i; \theta) \quad (13)$$

The stochastic gradient descent (SGD) family of first-order optimization techniques [3] 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.)

The exact gradient of this objective function is

$$\frac{\partial J}{\partial \theta} = -M\ell\theta + \sum_{i=1}^M \frac{\partial J_{S_1}}{\partial \theta}(t_i, m_i, c_i, \theta) \quad (14)$$

using the per-example gradient $\frac{dJ_{S_1}}{d\theta}$ given in (11). 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 (t, m, c) , the parameters are updated according to the formula

$$\theta := \theta + \alpha \left(-\ell\theta + \frac{\partial J_{S_1}}{\partial \theta}(t, m, c, \theta) \right) \quad (15)$$

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 [9], 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.

5 Example

In Figure 3, 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_1 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 3. 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 4; in TUNA data, properties that the target entities do not possess (e.g., \neg GLASSES) are also included among their “attributes.”

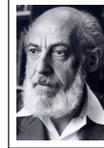
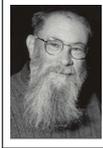
Below the feature representations, we summarize the gradient of the log likelihood ($\frac{\partial J_{S_1}}{\partial \theta}$) for each example, as an $m \times n$ table representing the weight update for each of the mn cross-product features. (We leave out the ℓ_2 regularization and AdaGrad learning rate for simplicity.) Tracing the formula for this gradient (11) back through the RSA layers to the literal listener (5), one can see that the gradient consists of the feature representation of the triple $\langle t, m, c \rangle$ 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 3(b) shows the distribution over utterances given target referent as predicted by the learned pragmatic speaker S_1 after one pass through the data with a fixed learning rate $\alpha = 1$ and no regularization ($\ell = 0$). We compare this distribution with the distribution predicted by the learned literal speaker S_0 and the pure RSA speaker s_1 . 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 3(b) 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_1 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]*.

	Training examples				Test example																																																			
Context																																																								
	r_2	r_3	r_3	r_4	r_1	r_4																																																		
Utterance	[<i>person</i>] with [<i>glasses</i>]		[<i>person</i>] with [<i>beard</i>]		[<i>person</i>] with [<i>glasses</i>]																																																			
Features for true utterance	PERSON \wedge [<i>person</i>] PERSON \wedge [<i>glasses</i>] GLASSES \wedge [<i>person</i>] GLASSES \wedge [<i>glasses</i>] \neg BEARD \wedge [<i>person</i>] \neg BEARD \wedge [<i>glasses</i>]		PERSON \wedge [<i>person</i>] PERSON \wedge [<i>beard</i>] \neg GLASSES \wedge [<i>person</i>] \neg GLASSES \wedge [<i>beard</i>] BEARD \wedge [<i>person</i>] BEARD \wedge [<i>beard</i>]		PERSON \wedge [<i>person</i>] PERSON \wedge [<i>glasses</i>] GLASSES \wedge [<i>person</i>] GLASSES \wedge [<i>glasses</i>] BEARD \wedge [<i>person</i>] BEARD \wedge [<i>glasses</i>]																																																			
Gradient	<table border="1"> <thead> <tr> <th></th> <th>[<i>person</i>]</th> <th>[<i>glasses</i>]</th> <th>[<i>beard</i>]</th> </tr> </thead> <tbody> <tr> <td>PERSON</td> <td>1</td> <td>1</td> <td>-1</td> </tr> <tr> <td>GLASSES</td> <td>2</td> <td>2</td> <td>-2</td> </tr> <tr> <td>BEARD</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>\negGLASSES</td> <td>-1</td> <td>-1</td> <td>1</td> </tr> <tr> <td>\negBEARD</td> <td>1</td> <td>1</td> <td>-1</td> </tr> </tbody> </table>				[<i>person</i>]	[<i>glasses</i>]	[<i>beard</i>]	PERSON	1	1	-1	GLASSES	2	2	-2	BEARD	0	0	0	\neg GLASSES	-1	-1	1	\neg BEARD	1	1	-1	<table border="1"> <thead> <tr> <th></th> <th>[<i>person</i>]</th> <th>[<i>glasses</i>]</th> <th>[<i>beard</i>]</th> </tr> </thead> <tbody> <tr> <td>PERSON</td> <td>1</td> <td>-1</td> <td>1</td> </tr> <tr> <td>GLASSES</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>BEARD</td> <td>2</td> <td>-2</td> <td>2</td> </tr> <tr> <td>\negGLASSES</td> <td>1</td> <td>-1</td> <td>1</td> </tr> <tr> <td>\negBEARD</td> <td>-1</td> <td>1</td> <td>-1</td> </tr> </tbody> </table>				[<i>person</i>]	[<i>glasses</i>]	[<i>beard</i>]	PERSON	1	-1	1	GLASSES	0	0	0	BEARD	2	-2	2	\neg GLASSES	1	-1	1	\neg BEARD	-1	1	-1	(unused)	
	[<i>person</i>]	[<i>glasses</i>]	[<i>beard</i>]																																																					
PERSON	1	1	-1																																																					
GLASSES	2	2	-2																																																					
BEARD	0	0	0																																																					
\neg GLASSES	-1	-1	1																																																					
\neg BEARD	1	1	-1																																																					
	[<i>person</i>]	[<i>glasses</i>]	[<i>beard</i>]																																																					
PERSON	1	-1	1																																																					
GLASSES	0	0	0																																																					
BEARD	2	-2	2																																																					
\neg GLASSES	1	-1	1																																																					
\neg BEARD	-1	1	-1																																																					

(a) Learned S_1 model training. Gradient values given are $6 \frac{\partial J_{S_1}}{\partial \theta}$, evaluated at $\theta = \vec{0}$.

s_1		S_0		S_1		
r_1	r_4	r_1	r_4	r_1	r_4	
.08	.25	.03	.00	.10	.11	\emptyset
.08	.25	.22	.10	.16	.13	[<i>person</i>]
.17	0	.03	.00	.11	.07	[<i>glasses</i>]
.08	.25	.03	.04	.08	.17	[<i>beard</i>]
.17	0	.22	.01	.18	.08	[<i>person</i>], [<i>glasses</i>]
.08	.25	.22	.74	.12	.19	[<i>person</i>], [<i>beard</i>]
.17	0	.03	.00	.10	.11	[<i>glasses</i>], [<i>beard</i>]
.17	0	.22	.10	.16	.11	[<i>person</i>], [<i>glasses</i>], [<i>beard</i>]

(b) Pure RSA (s_1), linear classifier (S_0), and learned RSA (S_1) 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 3: Specificity implicature and overgeneration in learned RSA.

6 Experiments

Data. We report experiments on the TUNA corpus (Section 3 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.

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:

$$\frac{2 \sum_{x \in D} \min [\mathbb{Z}_{a(m_i)}(x), \mathbb{Z}_{a(m_j)}(x)]}{|a(m_i)| + |a(m_j)|} \quad (16)$$

Here, $a(m)$ is the multiset of attributes of message m , D is the non-multiset union of $a(m_i)$ and $a(m_j)$, $\mathbb{Z}_X(x)$ is the number of occurrences of x in the multiset X , and $|a(m_i)|$ is the cardinality of multiset $a(m)$. *Accuracy* is the fraction of examples for which the subset of attributes is predicted perfectly (equivalent to achieving multiset Dice 1).

Experimental set-up. We evaluate all our agents in the same pragmatic contexts: for each trial in the *singular* corpus, we define the messages M to be the powerset of the attributes used in the referential description and the states T to be the set of entities in the trial, including the target. 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) $C(m) = 0$; (ii) $C(m) = |a(m)|$; and (iii) $C(m) = -|a(m)|$.

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 $P(t, m, c)$ and 0 representing its negation. Thus, each vector of real numbers that is the value of $\phi(t, m, c)$ 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., $P(t, m, c) = \text{RED}(t) \wedge [\text{blue}] \in m$). 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., $P(t, m, c) = \text{“}m \text{ contains a color”}$);
- (ii) pair-wise attribute type co-occurrences, where one can be negated (e.g., $\text{“}m \text{ contains a color and a size”}$, $\text{“}m \text{ contains an object type but not a color”}$); and
- (iii) message size in number of attributes (e.g., $\text{“}m \text{ consists of 3 attributes”}$).

For comparison, we also separately train literal speakers S_0 as in (4) (the log-linear model) with each of these feature sets using the same optimization procedure.

Table 1: 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).

Model	Furniture		People		All	
	Acc.	Dice	Acc.	Dice	Acc.	Dice
RSA s_0 (random true message)	1.0%	.475	0.6%	.125	1.7%	.314
RSA s_1	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_1 , basic feats.	23.1%	.789	11.9%	.740	17.9%	.766
Learned S_1 , gen. feats. only	17.4%	.740	1.9%	.712	10.3%	.727
Learned S_1 , basic + gen. feats.	27.6%	.788	22.5%	.764	25.3%	.777

Results. The results (Table 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_1 model significantly outperforms all other models. The value of the model’s flexibility in allowing a variety of feature designs 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 1, all but the last exhibit systematic errors. Pure RSA performs poorly for reasons predicted by [16]—for example, it under-produces color terms and head nouns like *desk*, *chair*, and *person*. This problem is also observed in the trained S_1 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_1 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 [14], namely, .762 multiset Dice, although this comparison must be informal since the test sets are different. (In particular, the Accuracy values given in [14] 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.

7 Conclusion

Our initial experiments 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.

This basic presentation suggests a range of potential next steps. For instance, it would be natural to apply the model to pragmatic interpretation (the listener’s perspective); this requires no substantive formal changes to the model as defined in Section 4, and it opens up new avenues in terms of evaluating pragmatic models in standard classification tasks like sentiment analysis, topic prediction, and natural language reasoning. In addition, for all versions of the model, one could consider including additional hidden speaker and listener layers, incorporating message costs and priors into learning, to capture a wider range of pragmatic phenomena.

References

- [1] Peter Baumann, Brady Clark, and Stefan Kaufmann. Overspecification and the cost of pragmatic reasoning about referring expressions. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Society*. Cognitive Science Society, 2014.
- [2] Leon Bergen, Roger Levy, and Noah D. Goodman. Pragmatic reasoning through semantic inference. Ms., MIT, UCSD, and Stanford, 2014.
- [3] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of the 19th International Conference on Computational Statistics*, pages 177–186. Springer, Berlin, 2010.
- [4] Colin F. Camerer, Teck-Hua Ho, and Juin-Kuan Chong. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3):861–898, 2004.
- [5] Stanley F. Chen and Ronald Rosenfeld. A Gaussian prior for smoothing maximum entropy models. Technical Report CMU-CS-99-108, Carnegie Mellon University, 1999.
- [6] Robert Dale and Ehud Reiter. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263, 1995.
- [7] Judith Degen and Michael Franke. Optimal reasoning about referential expressions. In Sarah Brown-Schmidt, Jonathan Ginzburg, and Staffan Larsson, editors, *Proceedings of the 16th Workshop on the Semantics and Pragmatics of Dialogue*, pages 2–11, 2012.
- [8] Pedro Domingos. A few useful things to know about machine learning. *Communications of ACM*, 55(10):78–87, 2012.
- [9] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, pages 2121–2159, 2011.
- [10] Paul E. Engelhardt, Karl G.D. Bailey, and Fernanda Ferreira. Do speakers and listeners observe the Gricean maxim of quantity? *Journal of Memory and Language*, 54(4):554–573, 2006.
- [11] Michael C. Frank and Noah D. Goodman. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998, 2012.
- [12] Michael C. Frank and Noah D. Goodman. Inferring word meanings by assuming that speakers are informative. *Cognitive Psychology*, 75(1):80–96, 2014.
- [13] Michael Franke. *Signal to Act: Game Theory in Pragmatics*. ILLC Dissertation Series. Institute for Logic, Language and Computation, University of Amsterdam, 2009.
- [14] Albert Gatt, Anja Belz, and Eric Kow. The TUNA Challenge 2008: Overview and evaluation results. In *Proceedings of the Fifth International Natural Language Generation Conference, INLG ’08*, pages 198–206. Association for Computational Linguistics, 2008.
- [15] Albert Gatt, Ielka van der Sluis, and Kees van Deemter. 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. DFKI GmbH, 2007.
- [16] Albert Gatt, Roger P. G. van Gompel, Kees van Deemter, and Emiel Kraemer. Are we Bayesian referring expression generators? In *Proceedings of the Thirty-Fifth Annual Conference of the Cognitive Science Society*, 2013.

- [17] Dave Golland, Percy Liang, and Dan Klein. 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, 2010.
- [18] H. Paul Grice. Logic and conversation. In Peter Cole and Jerry Morgan, editors, *Syntax and Semantics*, volume 3: Speech Acts, pages 43–58. Academic Press, 1975.
- [19] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, Berlin, 2 edition, 2009.
- [20] Gerhard Jäger. Game theory in semantics and pragmatics. In Claudia Maienborn, Klaus von Heusinger, and Paul Portner, editors, *Semantics: An International Handbook of Natural Language Meaning*, volume 3, pages 2487–2425. Mouton de Gruyter, 2012.
- [21] Gerhard Jäger. Rationalizable signaling. *Erkenntnis*, 79(4):673–706, 2014.
- [22] Justine T. Kao, Leon Bergen, and Noah D. Goodman. Formalizing the pragmatics of metaphor understanding. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Society*, pages 719–724. Cognitive Science Society, 2014.
- [23] Justine T. Kao, Jean Y. Wu, Leon Bergen, and Noah D. Goodman. Nonliteral understanding of number words. *Proceedings of the National Academy of Sciences*, 111(33):12002–12007, 2014.
- [24] Willem JM Levelt. *Speaking: From Intention to Articulation*, volume 1. MIT press, 1993.
- [25] David Lewis. *Convention*. Harvard University Press, 1969. Reprinted 2002 by Blackwell.
- [26] Percy Liang and Christopher Potts. Bringing machine learning and compositional semantics together. *Annual Review of Linguistics*, 1(1):355–376, 2015.
- [27] Peter McCullagh and John A. Nelder. *Generalized Linear Models*. Chapman and Hall, London, 1989.
- [28] Brian McMahan and Matthew Stone. A Bayesian model of grounded color semantics. *Transactions of the Association for Computational Linguistics*, 3:103–115, 2015.
- [29] Thomas Pechmann. Incremental speech production and referential overspecification. *Linguistics*, 27(1):89–110, 1989.
- [30] Christopher Potts, Daniel Lassiter, Roger Levy, and Michael C. Frank. Embedded implicatures as pragmatic inferences under compositional lexical uncertainty. To appear in *Journal of Semantics*, 2015.
- [31] Seymour Rosenberg and Bertram D. Cohen. Speakers’ and listeners’ processes in a word communication task. *Science*, 145:1201–1203, 1964.
- [32] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 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, Cambridge, MA, 1986.
- [33] Alex Stiller, Noah D. Goodman, and Michael C. Frank. Ad-hoc scalar implicature in adults and children. In Laura Carlson, Christoph Hoelscher, and Thomas F. Shipley, editors, *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society*, pages 2134–2139. Cognitive Science Society, 2011.
- [34] Kees van Deemter, Ielka van der Sluis, and Albert Gatt. Building a semantically transparent corpus for the generation of referring expressions. In *Proceedings of the Fourth International Natural Language Generation Conference*, pages 130–132. Association for Computational Linguistics, 2006.
- [35] Adam Vogel, Max Bodoia, Christopher Potts, and Dan Jurafsky. 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*, pages 1072–1081. Association for Computational Linguistics, 2013.
- [36] Adam Vogel, Andrés Gómez Emilsson, Michael C. Frank, Dan Jurafsky, and Christopher Potts. Learning to reason pragmatically with cognitive limitations. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Society*, pages 3055–3060. Cognitive Science Society, 2014.