A simple repair mechanism can alleviate computational demands of pragmatic reasoning: simulations and complexity analysis

Jacqueline van Arkel[1] **Marieke Woensdregt**[2] **Mark Dingemanse**[2] **Mark Blokpoel**[3]

[1]Faculty of Science and Engineering, University of Groningen, Groningen, the Netherlands
[2]Centre for Language Studies, Radboud University, Nijmegen, the Netherlands
[3]Donders Centre for Cognition, Radboud University, Nijmegen, the Netherlands
jacquelinevanarkel@gmail.com, m.woensdregt@let.ru.nl
m.dingemanse@let.ru.nl, m.blokpoel@donders.ru.nl

Abstract

How can people communicate successfully while keeping resource costs low in the face of ambiguity? We present a principled theoretical analysis comparing two strategies for disambiguation in communication: (i) pragmatic reasoning, where communicators reason about each other, and (ii) other-initiated repair, where communicators signal and resolve trouble interactively. Using agent-based simulations and computational complexity analyses, we compare the efficiency of these strategies in terms of communicative success, computation cost and interaction cost. We show that agents with a simple repair mechanism can increase efficiency, compared to pragmatic agents, by reducing their computational burden at the cost of longer interactions. We also find that efficiency is highly contingent on the mechanism, highlighting the importance of explicit formalisation and computational rigour.

1 Introduction

Natural languages are rife with ambiguity (Wasow et al., 2005), yet people seem to communicate efficiently regardless. How can people communicate successfully in the face of ambiguity while keeping resource costs low? There seem to be at least three strategies communicators have at their disposal. First, contextual information can be used to disambiguate the speaker's intended meaning (Piantadosi et al., 2012; Sperber and Wilson, 1986; MacDonald et al., 1994), though context-sensitive computations are notorious in computational cognitive science for the astronomical demands they make on computation time (Fodor, 2000; Haselager, 1997; van Rooij et al., 2011). Second, pragmatic reasoning allows taking into account the speaker's goal (e.g. 'being informative') (Grice, 1975; Sperber and Wilson, 1986; Goodman and Frank, 2016), but this alone is not always enough to fully disambiguate meaning (Schegloff, 1992). Finally, communicators can leverage the interaction itself by explicitly requesting clarification (e.g. by asking 'Huh?' or 'Who?') in a process known as other-initiated repair (Schegloff et al., 1977; Purver et al., 2018). This provides a possible way for communicators to reduce their computational burden through interaction, potentially increasing communicative efficiency (Dingemanse, 2020).

To investigate the computational plausibility of this potential gain in communicative efficiency we present a theoretical analysis of other-initiated repair and pragmatic reasoning. Following Gibson et al. (2019), we define efficient communication as communication in which participants reach mutual understanding while requiring minimal effort in terms of resource costs (deconstructed here as the sum of computational and interactional cost). We compare a novel agent-based model of other-initiated repair with one of pragmatic reasoning (Goodman and Frank, 2016) for both their communicative success and use of computational and interactional resources. Simulations are used to evaluate the models' success and interactional resource costs while a computational complexity analysis is used to determine the computational resource demands (van Rooij, 2008; van Rooij et al., 2019).

The results show that, on roughly equal terms of communicative success, agents with a simple repair mechanism can reduce their computational burden compared to pragmatic agents, at the cost of longer interactions. While this shows that an efficiency-increasing trade-off is in principle possible, the question remains whether the computational advantage scales to more complex forms of other-initiated repair. The work we present here makes two contributions: 1) a proof of concept that a simple form of repair can help communicators outsource computational demands in interaction, and 2) a framework for the careful theoretical anal-

Proceedings of the 24th Conference on Computational Natural Language Learning, pages 177–194

https://doi.org/10.18653/v1/2020.conll-1.14

ysis of the interplay of cognitive and interactional resources in human communication.

2 Background

A computational model of pragmatic reasoning in communication that is widely used and has been shown to fit empirical data of human communicative behaviour well, is the rational speech act (RSA) model (Frank and Goodman, 2012; Goodman and Frank, 2016). This model formalises communication as rational behaviour in which a speaker chooses an utterance by maximising its utility, where utility is defined as the probability that the listener will correctly infer the speaker's communicative intention[1]. This means that the speaker reasons about a listener when choosing an utterance. Likewise, the listener in the RSA model reasons about a speaker by inverting this model of rational utterance production: inferring what the speaker's most likely communicative intention is given the utterance produced (using Bayesian inference). Thus, both RSA production and RSA interpretation consist of a chain of recursive social reasoning, eventually bottoming out in a literal (i.e. zero-order) speaker or listener, which is where the interaction is grounded in semantic meaning. We take this model as our basis to implement pragmatic reasoning for disambiguation in communication.

As mentioned above, another mechanism that human communicators use to reach mutual understanding is repair (Schegloff et al., 1977; Clark and Schaefer, 1987). Cross-linguistic work on informal face-to-face conversation has shown that repair is frequent (on average once every 1.4 minutes) and that it is highly similar in form and function across unrelated languages (Dingemanse et al., 2015). Attested repair initiations fall into three basic types, which differ in the grasp they display of the trouble source: (i) *open request* (e.g. 'Huh?'), (ii) *restricted request* (e.g. 'Who?') and (iii) *restricted offer* (e.g. 'At the market?'). These types are used according to similar principles across languages, with participants requesting clarification when necessary and reusing material when possible, resulting in repair sequences that appear to minimise the joint effort of speaker and listener (Dingemanse et al., 2015; Clark and Wilkes-Gibbs, 1986).

Interactive repair is a universal and frequently used mechanism for resolving trouble in communication. Here we hypothesise that it provides an affordance that inference based on context or pragmatic reasoning does not: it allows at least part of the computational burden of making inferences to be offloaded onto interaction, in effect distributing the process of reaching mutual understanding over multiple interactional turns (Dingemanse, 2020). This can be seen as a form of cognitive offloading (Risko and Gilbert, 2016), with turns at talk constituting material symbols that can augment cognitive processes (Clark, 2006). In this paper we combine agent-based simulations with a computational complexity analysis to investigate the relative resource demands of pragmatic reasoning and interactive repair. We aim to find out whether other-initiated repair can increase communicative efficiency by relieving communicators of the computational demands of pragmatic reasoning, without that causing a decrease in communicative success.

3 Methods

3.1 Computational models[2]

We use agent-based simulations to compare the communicative efficiency (in terms of both success and resource costs) of other-initiated repair (OIR) and pragmatic reasoning. As reviewed above, people use both strategies for disambiguation in natural conversation. Here, however, we separate them in order to create a baseline comparison between the two. We design two separate models: (i) an interactional model, in which agents have the ability to use repair, but do not use pragmatic reasoning, and (ii) a pragmatic model, in which agents use pragmatic reasoning, but do not have the ability to use repair.

Both models of communication start from a lexicon consisting of binary signal-referent mappings (see Table 1 for an example). Depending on the model of communication (interactional or pragmatic), speakers and listeners use this lexicon in different ways in order to arrive at signal productions and interpretations.

3.1.1 Interactional model

In the interactional model, agents are literal communicators who do not use pragmatic reasoning but can initiate repair. The main innovation we present here is a model of other-initiated repair

[1] We use the conventional 'speaker' and 'listener', though we are aware that natural languages are produced and perceived in diverse modalities.

[2] The implementation code and simulation data are available at: https://osf.io/fxphv/.

	r_1	r_2	r_3	r_4
s_1	0	1	1	0
s_2	1	0	1	0
s_3	1	1	0	0
s_4	1	0	0	1

Table 1: Example of a simple lexicon. s denotes a signal, and r a referent. This lexicon has an ambiguity level of 0.5: every signal is associated with half of the referents.

governed by the listener's level of certainty about the speaker's intended referent. Our model consists of three parts. First, after each signal production by the speaker, we measure the listener's uncertainty as the conditional entropy of the probability distribution over referents given the signal (MacKay, 2003). Second, we define an entropy threshold parameter which simulates the amount of uncertainty that the listener is willing to tolerate: when a listener's uncertainty falls above this threshold (i.e. uncertainty is too high), they initiate repair using an open request (which one can think of as saying 'Huh?' or 'What did you say?') (for a related use of entropy as a trigger for repair, see de Ruiter and Cummins, 2012). Finally, we provide a simple mechanism for solving the ambiguity problem indicated by the listener: the speaker can send another signal associated with the intended referent, and the listener then performs a conjunction operation to determine what referents are in the intersection of the current signal and the previous signal(s), thereby (potentially) reducing referential uncertainty. When the conditional entropy of the listener's probability distribution over referents given the signal(s) received falls below the entropy threshold (i.e. when uncertainty is low enough), an interpretation is reached by choosing the referent that has maximum posterior probability.

For example, imagine a speaker with an intention to communicate referent 2 who has just uttered signal 3 based on the lexicon in Table 1. After the listener has initiated repair, the speaker utters signal 1, which leads to the association vector of $[0, 1, 0, 0]$ after conjunction, and now the listener can be certain referent 2 is the speaker's intended referent. Below we give a computational-level description of production and interpretation in this interactional model.

PRODUCTION

Input: A set of signals S, a set of referents R, a lexicon $\mathcal{L} : S \times R \to \mathbb{B}$ mapping signal-referent pairs to a Boolean value. We write $\mathcal{L}(s)$ to denote the list of values for all referents given signal s. A dialogue history D_r which is a set of signals produced earlier in a conversation $\{s, \dots\}$. The dialogue history D_r is relative to the intended referent r by the speaker. An order of pragmatic inference $n = 0$. And finally an intended referent $r \in R$.

Output: The signal s that maximizes the probability $\Pr^S_{S_0}(s \mid r, \mathcal{L}_{D_r})$, where

$$\mathcal{L}_{D_r}(s, r) = \mathcal{L}(s, r) \bigwedge_{s' \in D_r} \mathcal{L}(s', r)$$

For interactional production, the following equations are relevant:

$$\Pr^S_{S_0}(s \mid r, \mathcal{L}_{D_r}) = \delta_S(s|r, \mathcal{L}_{D_r}) \quad (1)$$

$$\delta_S(s|r, \mathcal{L}_{D_r}) = \frac{\mathcal{L}_{D_r}(s, r)}{\sum_{s' \in S} \mathcal{L}_{D_r}(s', r)} \quad (2)$$

Equation 1 shows the probability of a signal s given the intended referent r and the lexicon updated according to the dialogue history $\mathcal{L}_{D_r}$. For an interactional speaker (who uses literal production), this probability is given by Equation 2, which normalises the lexicon over signals, given the intended referent.

INTERPRETATION

Input: $\mathcal{L}$, $\mathcal{L}(s)$, and D_r as defined for the production model above. An order of pragmatic inference $n = 0$. An entropy threshold H_t determining whether the entropy H is too high or sufficiently low. And finally an observed signal $s \in S$.

Output: $\mathcal{L}_{D_r}(s, r)$ as defined for the production model above. Let $\Pr^L_{L_0}(r \mid s, \mathcal{L}_{D_r})$ provide the posterior distribution over referents given s and $\mathcal{L}_{D_r}$, and let $H(R|s, \mathcal{L}_{D_r})$ be the conditional entropy (i.e. uncertainty) of that distribution. The output is of one of two types: a repair signal, or an inferred referent given the signal and dialogue history:

$$\begin{cases} \text{repair signal} & \text{if } H(R|s, \mathcal{L}_{D_r}) > H_t \\ \arg\max_{r \in R} \Pr^L_{L_0}(r \mid s, \mathcal{L}_{D_r}) & \text{if } H(R|s, \mathcal{L}_{D_r}) \leq H_t \end{cases}$$

For interactional interpretation, the following equations are relevant:

$$\Pr^L_{L_0}(r \mid s, \mathcal{L}_{D_r}) = \delta_L(r|s, \mathcal{L}_{D_r}) \quad (3)$$

$$\delta_L(r|s, \mathcal{L}_{D_r}) = \frac{\mathcal{L}_{D_r}(s, r)}{\sum_{r' \in R} \mathcal{L}_{D_r}(s, r')} \quad (4)$$

$$H(R|s, \mathcal{L}_{D_r}) = \sum_{r \in R} \Pr(r \mid s, \mathcal{L}_{D_r}) \times \log_2 \frac{1}{\Pr(r \mid s, \mathcal{L}_{D_r})} \quad (5)$$

Equation 3 shows the probability of a referent r given the received signal s and the lexicon updated according to the dialogue history $\mathcal{L}_{D_r}$. For an interactional listener (who uses literal interpretation), this probability is given by Equation 4, which normalises the lexicon over referents given the received signal. Finally, the conditional entropy of the probability distribution over referents given the signal and the lexicon updated according to the dialogue history is shown in Equation 5.

3.1.2 Pragmatic model

The pragmatic model is based on the RSA framework (Frank and Goodman, 2012; Goodman and Frank, 2016). This framework models pragmatic reasoning as a chain of social recursion, in which the speaker reasons about the listener when choosing a signal, and the listener reasons about the speaker when interpreting a signal. Figure 1 shows the chain of reasoning used in the current model.

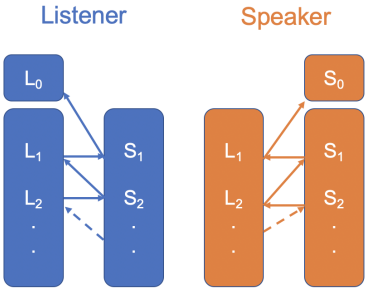

Figure 1: Pragmatic reasoning model for listener and speaker. Arrow direction represents a 'reasons about' relationship, illustrating the recursive reasoning being done by the agents. Agents reason about increasingly lower levels, eventually bottoming out in a literal listener or speaker respectively.

In order to not stack the deck against the pragmatic agents in terms of computational burden, we further distinguish between two subtypes of pragmatic agents: 'frugally pragmatic' and 'fully pragmatic'. A frugally pragmatic listener starts out at a low level of social recursion (order $n = 1$), and only 'levels up' to a higher order of pragmatic reasoning $(n + 1)$ when too uncertain about the speaker's intended referent. Thus, they decide how to proceed based on their own uncertainty, somewhat analogously to how the interactional listener decides whether to initiate repair. In contrast, a fully pragmatic listener starts at the maximum order of pragmatic reasoning straight away (here we cap pragmatic reasoning at order 2, as previous simulation work has shown that orders higher than 2 yield diminishing returns in terms of communicative success; Blokpoel et al., 2020). As this paper focuses on disambiguation by the listener, we keep the speaker model that these two subtypes of pragmatic listener interact with constant: a 'fully pragmatic' speaker who starts at the maximum order of pragmatic reasoning straight away. Below we give a computational-level description of production and interpretation in this pragmatic model.

PRODUCTION

Input: $\mathcal{L}$ and $\mathcal{L}(s)$ as defined above (see Production in Interactional Model; Section 3.1.1). An order of pragmatic inference $n = 2$, and an intended referent $r \in R$.

Output: The signal s that maximizes the probability $\Pr^S_{S_n}(s \mid r, \mathcal{L})$.

$$\Pr^S_{S_n}(s \mid r, \mathcal{L}) = \frac{\Pr^S_{L_n}(r \mid s, \mathcal{L})}{\sum_{s' \in S} \Pr^S_{L_n}(r \mid s', \mathcal{L})} \quad (6)$$

$$\Pr^S_{L_n}(r \mid s, \mathcal{L}) = \frac{\Pr^S_{S_{n-1}}(s \mid r, \mathcal{L})}{\sum_{r' \in R} \Pr^S_{S_{n-1}}(s \mid r', \mathcal{L})} \quad (7)$$

$$\Pr^S_{S_0}(s \mid r, \mathcal{L}) = \delta_S(s|r, \mathcal{L}) \quad (8)$$

$$\delta_S(s|r, \mathcal{L}) = \frac{\mathcal{L}(s, r)}{\sum_{s' \in S} \mathcal{L}(s', r)} \quad (9)$$

For pragmatic production, the speaker reasons about the listener (Equation 6), who in turn reasons about the speaker being one order of pragmatic reasoning below (Equation 7). Finally, this bottoms out to reasoning about a literal (zero-order) speaker (Equation 8), where the normalised lexicon comes into play (Equation 9).

INTERPRETATION

Input: $\mathcal{L}$ and $\mathcal{L}(s)$ as defined above (see Production in Interactional Model; Section 3.1.1). An order of pragmatic inference n with a maximum at $n_{max} = 2$. An entropy threshold H_t determining whether the entropy H is too high or sufficiently

low. And finally an observed signal $s \in S$.

Output: Let $\Pr^L_{L_n}(r \mid s, \mathcal{L})$ be the posterior distribution over referents given s and $\mathcal{L}$, and let $H(R|s, \mathcal{L})$ be the conditional entropy (i.e. uncertainty) of that distribution. The output is an inferred referent r given the signal, if needed by moving a level up on the order of pragmatic reasoning:

$$\begin{cases} \text{RSA INTERPRETATION}(n+1) & \text{if } H(R|s,\mathcal{L}) > H_t, \text{ and } n < n_{max} \\ \arg\max_{r \in R} \Pr^L_{L_n}(r \mid s, \mathcal{L}) & \text{if } H(R|s,\mathcal{L}) \leq H_t, \text{ or } n = n_{max} \end{cases}$$

$$\Pr^L_{L_n}(r \mid s, \mathcal{L}) = \frac{\Pr^L_{S_n}(s \mid r, \mathcal{L})}{\sum_{r' \in R} \Pr^L_{S_n}(s \mid r', \mathcal{L})} \quad (10)$$

$$\Pr^L_{S_n}(s \mid r, \mathcal{L}) = \frac{\Pr^L_{L_{n-1}}(r \mid s, \mathcal{L})}{\sum_{s' \in S} \Pr^L_{L_{n-1}}(r \mid s', \mathcal{L})} \quad (11)$$

$$\Pr^L_{L_0}(r \mid s, \mathcal{L}) = \delta_L(r|s, \mathcal{L}) \quad (12)$$

$$\delta_L(r|s, \mathcal{L}) = \frac{\mathcal{L}(s, r)}{\sum_{r' \in R} \mathcal{L}(s, r')} \quad (13)$$

$$H(R|s, \mathcal{L}) = \sum_{r \in R} \Pr(r \mid s, \mathcal{L}) \times \log_2 \frac{1}{\Pr(r \mid s, \mathcal{L})} \quad (14)$$

For pragmatic interpretation, the listener reasons about the speaker (Equation 10), who in turn reasons about the listener being one order of pragmatic reasoning below (Equation 11). This bottoms out to reasoning about a literal (zero-order) listener (Equation 12), where the normalised lexicon comes into play (Equation 13). Finally, the conditional entropy of the probability distribution over referents given the signal is shown in Equation 14.

3.2 Complexity theory

Computational-level models such as those above have very specific computational resource demands. These demands can be analysed using mathematical proof techniques from computational complexity theory (Garey and Johnson, 1979). A model's resource demands (also referred to as computational complexity) are defined by the worst-case running time of the fastest possible algorithm that computes the specified input-output mapping. Worst-case complexity is most appropriate assuming that all instances from the model's input domain may possibly occur.[3] The computational complexity of a model can be proven by reduction or by proposing an algorithm, and is given in terms of the input size of the model (e.g., the size of the lexicon).

In the first method (reduction), one constructs a mathematical relationship, i.e., a polynomial-time reduction, between the model of interest (say M_I) and a model who's complexity is known (say M_K). A reduction proves that either M_I is a special case of M_K or the other way around.[4] Depending on the complexity of M_K, the reduction may inform us about the complexity of M_I. If M_I reduces to M_K and M_K is easy, then M_I must be easy too, because we can use the 'fast' algorithm that exists for M_K to compute M_I. If M_K reduces to M_I and M_K is hard, then M_I must be hard too, otherwise if M_I would be easy, we could compute M_K easily too. A reduction is denoted as $A \leq B$, where A reduces to B.

$$\begin{aligned} M_I \text{ is easy} &\iff M_I \leq M_K \text{ and } M_K \text{ is easy} \\ M_I \text{ is hard} &\iff M_I \geq M_K \text{ and } M_K \text{ is hard} \end{aligned}$$

Polynomial time reductions can be used to prove that models are easy or hard. Easy models belong to the complexity class P and for these models there exist polynomial-time (or faster) algorithms. Hard models belong to class NP-hard; these models are as hard as all other models in NP and require exponential time or worse, assuming that P $\neq$ NP. See Table 2 for example resource requirements.

In the second method (proposing an algorithm), one creates an algorithm that computes the model *exactly* and then analyses the algorithm's complexity profile. Unless one can prove the algorithm is the fastest, this method gives an upperbound on the model's computational complexity. This method affords comparison between models of similar complexity class. This is the method we use to deter-

[3] If one finds this assumption to generic, one can propose a restricted special case model. Such a model may have a different computational complexity. Parameterized complexity analysis (Downey and Fellows, 1999; van Rooij et al., 2019) is a sophisticated approach for investigating various special case models.

[4] A polynomial-time reduction from A to B does not strictly prove a special case relationship. Formally it proves that at polynomial cost any input of A can be transformed into an equivalent input for B such that the output of B is consistent with the output of A.

mine the complexity of the interactional and pragmatic models, because they are both polynomial-time computable. We illustrate this method using matrix row normalization. Given a definition of basic computation step (e.g., multiplication), input size (e.g., $\max(|\text{rows}|, |\text{columns}|)$) and an algorithm (see Algorithm 1), one expresses the number of required computation steps. Here, n^2 computations steps are required.

Algorithm 1: Matrix row normalization taking $2kl = n^2$ steps, where $n = \max(k, l)$.

```
Data: M is a k × l matrix
for i ← 1 to k do
    for j ← 1 to l do
        S_i ← S_i + M_ij ;      // k x l steps
    end
end
for i ← 1 to k do
    for j ← 1 to l do
        M_ij ← M_ij / S_i ;     // k x l steps
    end
end
```

		Easy		Hard
n	$\log n$	n	n^3	2^n
5	.0069ms	.5ms	12.5ms	3.2ms
20	.013ms	2ms	.8s	105s
50	.017ms	5mss	1.3s	31,274,997h
100	.020ms	10ms	100s	9.6×10^{19}y
250	.026ms	25mss	26min	1.4×10^{65}y
500	.027ms	50mss	3.5h	9.6×10^{140}y

Table 2: Illustration of time required to compute models of varying complexity with input size n.

Using the second method, we derived upper bounds on the computational complexity of each model (see Appendix B for the full proofs). Table 3 shows the computational complexity for the different agent types.

Interactional	**Frugally pragmatic**	**Fully pragmatic**
$2m(t-1) + 2mt + 2m$	**1:** $16m^2 + 4m$ **2:** $20m^2 + 4m$	$20m^2 + 2m$

Table 3: Computational complexity comparison across agent types. m denotes the maximum of $|S|$ and $|R|$ (number of signals and referents, respectively), and t denotes the number of turns. Frugally pragmatic agents may end up in one of two scenarios: either (1) they are sufficiently certain about their 1st-order inference or (2) they will make an additional 2nd-order inference.

3.3 Simulation details

For the purposes of this paper, we assume that there is no disparity between the agent types within a given speaker-listener pair, meaning that interactional speakers always converse with an interactional listener, and pragmatic speakers always converse with a pragmatic listener. This provides a clear-cut contrast to compare the effect of OIR versus pragmatic reasoning on efficiency in communication.

We ran simulations to see which agent type performs best at communicating efficiently (which we break down into communicative success and resource costs). These simulations consist of a set of interactions between two agents. An interaction starts with the speaker being assigned a randomly chosen intended referent, and ends when the listener reaches an interpretation based on the signal(s) sent by the speaker. If the agents are of the interactional type, they can use multiple turns; if the agents are pragmatic, the speaker can only send one signal. We cap the number of turns at $2 \times |S| - 1$, to make sure agents do not get stuck in an infinite loop of other-initiated repair. In addition to interacting agents being of the same type, we also assume that there is no asymmetry between interacting agents: they always share the same lexicon.

In the simulations described below, we looked at three different lexicon sizes ($|S| \times |R|$=6x4, 15x10, and 30x20) in order to investigate how the efficiency of the different strategies scales with lexicon size. We kept the ambiguity of the lexicons constant at a moderate level of 0.5 (given that we are interested in disambiguation), and the entropy threshold constant at $H_t = 1.0$ bits (which corresponds approximately to a probability distribution where most of the probability mass is distributed equally over two referents).Following Blokpoel et al. (2020), we define lexicon ambiguity as mean signal ambiguity, and signal ambiguity as the relative number of referents a signal is associated with. Appendix A shows additional simulation results that explore the effects of varying the ambiguity level and entropy threshold parameters.

For each combination of parameter settings, we randomly generate 1,000 lexicons of the corresponding size and ambiguity level, and have the corresponding pair of agents interact for $2 \times |R|$ times (about randomly selected referential intentions). We constrain the set of possible lexicons

such that (i) each referent has at least one signal associated with it, and (ii) each signal has an equal level of ambiguity. The latter constraint is to avoid potential effects of skewed ambiguity (e.g. when half of the signals refer to all referents and the other signals to none, in the case of a mean ambiguity of 0.5) (Blokpoel et al., 2020).

3.4 Measures: Communicative success and resource costs

For each simulation, we measured (i) the communicative success, (ii) the interactional cost, and (iii) the computational cost. We define communicative success as 1.0 if the listener's interpretation matches the speaker's intended referent, and 0.0 otherwise. We define interactional cost as the number of turns (i.e. the total number of signals and repair initiators that are sent back and forth between speaker and listener). Computational resource requirements are based on the complexity upper bound derived for each model (see Table 3 and Appendix B).

4 Results

Figure 2a shows the mean communicative success for the different agent types and lexicon sizes. The frugally pragmatic listeners were always sufficiently certain about the intended referent of the speaker when using a lexicon of size 6x4, resulting in the agents staying with their first order inference for that lexicon size. For the lexicon sizes of 15x10 and 30x20, the frugally pragmatic listeners were always too *un*certain about the speaker's intended referent, and therefore always went up to order $n = 2$.[5]

As Figure 2a shows, the pragmatic agents have an advantage in terms of communicative success for the smallest lexicon size (6x4), while for bigger lexicon sizes (15x10 and 30x20) the interactional agents have an advantage. This can be accounted for by the fact that the interactional agents do not use OIR for a lexicon with only 4 referents and an ambiguity level of 0.5 (see Figure 2b), as they are already certain enough[6], and therefore choose randomly between two referents straight away (resulting in ∼50% communicative success). For bigger lexicons, however, they do use OIR, which explains the increased communicative success: through multiple turns they can reduce referential uncertainty.

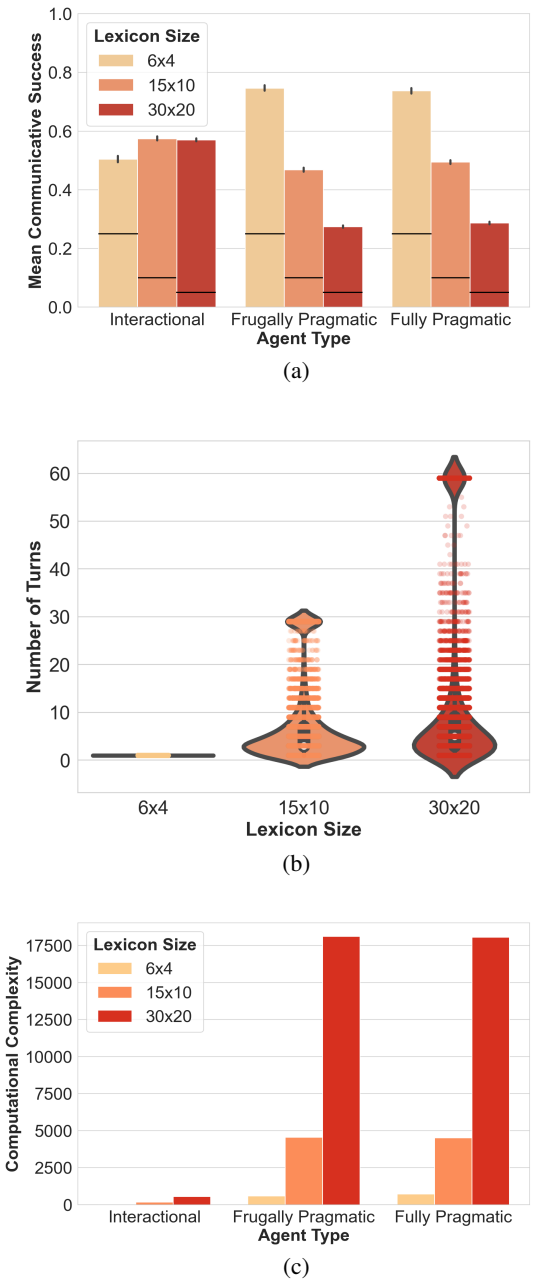

Figure 2: (a) Communicative success by agent type and lexicon size (horizontal lines indicate chance level, error bars 95% CIs). (b) Number of turns by lexicon size (interactional agents only); turns >1 increment by 2 since repair sequences are paired turns. (c) Computational complexity (in basic computation steps) by agent type and lexicon size. For interactional agents with a 6×4 lexicon no data is visible as the computation cost is very small (48) relative to the range of the y-axis.

[5]This model behaviour depends on the entropy threshold (lower values mean agents tolerate less uncertainty), ambiguity level (more ambiguous lexicons lead to more uncertainty), and lexicon size (larger lexicons result in more dispersed probability distributions, which causes higher uncertainty). See Appendix A for results with different parameter settings.

[6]Recall that the entropy threshold of 1.0 bits corresponds approximately to an equal distribution of probability mass over two referents.

Interactional agents perform approximately equally well with the bigger lexicons of 15x10 and 30x20, while the pragmatic agents show a steady decline in communicative success for bigger lexicon sizes. This decline can be explained by bigger lexicons resulting in more dispersed probability distributions, which causes less certainty for both speakers and listeners when choosing their productions and interpretations. This is more of a problem for pragmatic agents as they cannot do anything other than go one level up in pragmatic reasoning, while interactional agents can take as many turns as needed to reduce referential uncertainty (for as far as their lexicon allows). For pragmatic agents, we see no difference in communicative success between the Frugally Pragmatic and Fully Pragmatic strategies. This is as expected since they have access to the same pragmatic reasoning mechanisms and differ only in the successive deployment of orders of reasoning.

Figure 2b shows the distribution of the number of turns for the interactional agents. Here, a clear effect of lexicon size is visible: the bigger the lexicon, the more turns are used. This is unsurprising given that larger lexicons (given a constant ambiguity level) contain more referent associations per signal. Therefore, a larger lexicon causes more uncertainty, which results in more turns. (Note that we allowed agents to take more turns for bigger lexicons: we set a cap at $2 \times |S| - 1$ turns.) For the smallest lexicon size of 6x4, only one turn (i.e. one speaker production) is needed for the listener to be certain enough to end the interaction, meaning that listeners do not make use of OIR for this lexicon size. Most interactional sequences take less than 10 turns in total regardless of lexicon size. This means that interactional listeners need on average less than 5 repair attempts to reach a sufficiently certain interpretation.

Figure 2c shows the computation cost (as means of the computational complexity) by agent type and lexicon size. For the interactional agents, the average number of turns per lexicon size (6x4: 1.0, 15x10: 3.0, and 30x20: 4.7 turns) is entered into the computation cost, since the worst case is defined by an artificial limit on interaction length. As mentioned above, the frugally pragmatic agents always went up to order $n = 2$ for lexicon sizes 15x10 and 30x20, resulting in almost the same computation cost as for the fully pragmatic agents (see also Table 3). Only for a lexicon of size 6x4 the frugally pragmatic agents were certain enough to stay with their first-order inference, ending up with a slightly lower computation cost than the fully pragmatic agents.

There is a substantial difference in computation cost between the interactional and pragmatic agent types. Especially for larger lexicons the computation cost is considerably lower for interactional than for pragmatic agents. Compared to this difference, the degree to which computation cost is reduced for Frugally Pragmatic compared to Fully Pragmatic agents is a lot smaller. The effect of lexicon size is smaller for the interactional compared to the pragmatic agents, as the computation cost increases linearly with lexicon size for interactional agents, while it increases quadratically with lexicon size for pragmatic agents (see Table 3).

5 Discussion

Can communicators reduce their computational burden through interaction? We showed using a theoretical analysis that the use of other-initiated repair can be more efficient than pragmatic reasoning in communication, by reducing the computational demands of pragmatic reasoning through interaction. The chief computational advantage of repair in our model derives from the fact that it trades recursive pragmatic inferences (which scale quadratically with lexicon size) for computationally simpler conjunctions (which scale linearly). This advantage seems to scale to bigger lexicon sizes as well, with the communicative success of the interactional agents not being affected by lexicon size, whereas pragmatic agents' communicative success decreases. This supports the hypothesis that communicating agents can leverage interactive repair to reduce their computational burden, essentially outsourcing individual computation to interaction.

A number of design choices may affect the generalisability of these results. First, we have modelled only a simple form of interactive repair, albeit one corresponding to a widely used repair format (the open request). Other forms of repair may have different computational complexity profiles. For instance, restricted offers hold up a candidate understanding for confirmation, and their formulation likely requires some degree of pragmatic reasoning, adding to the computational complexity (Schlöder and Fernández, 2015). Also, dealing with some forms of repair may involve belief revision (Wilkes-Gibbs and Clark, 1992), which requires context-

sensitive abductive inferences known to be computationally intractable (Abdelbar and Hedetniemi, 1998; Bylander et al., 1991; Thagard and Verbeurgt, 1998). In sum, other-initiated repair is not a monolithic phenomenon, and the analytical tools we supply here can be used to systematically investigate the computational tractability of a range of possible interactional strategies (see e.g. Ginzburg and Fernández, 2010; van Rooij et al., 2011).

Another limitation is that the conditions under which the agents communicate are unrealistic in that all agent pairs share the exact same lexicon. Any potential misunderstanding thus stems solely from ambiguity, and not from one agent associating a given signal with a slightly different set of referents than their interlocutor. Relaxing this assumption is likely to cause problems for the simple repair strategy presented here, because it is based on conjunction. If interactional agents would base their (literal) productions and interpretations on conjunctions of more asymmetrical lexicons, divergences between intended referent and interpretation would soon arise, in which case we predict a decrease in communicative success, and therefore in efficiency. Pragmatic agents, on the other hand, have been shown to be able to leverage a moderate level of ambiguity in their lexicons to overcome asymmetry (Blokpoel et al., 2020).

We now consider two possible extensions to the current modelling work. Note, however, that these both come with additional computational demands and require careful theoretical re-analysis to investigate where efficiency trade-offs may play a role.

First, a hybrid model of pragmatic inference and other-initiated repair might combine the best of both worlds. The question then is if agents can achieve communicative success while keeping resource demands low by having their choice of strategy depend on an assessment of the situation. For example, in a speaker role, such a hybrid agent could 'level up' to a higher order of pragmatic reasoning in response to a repair initiator. Such hybrid agents will of course need a meta-cognitive capacity to decide which strategy to use (for instance by reasoning about the level of asymmetry between themselves and their interlocutor). This meta-level reasoning would bring additional computational resource demands that would affect the agents' efficiency. While a hybrid strategy may be able to preserve some of the efficiency trade-offs we have documented here, it is an open question whether they would not be dwarfed by the added computational cost of meta-cognition (see e.g. Otworowska et al., 2018).

Second, agents may revise their beliefs about the way their interlocutors use signals on the basis of conversation history, in order to overcome asymmetry (Hawkins et al., 2017). This form of updating might be able to explain why people are successful communicators while spending minimal interactional resources, but it comes at a computational cost too. Consider that agents would have to entertain the possibility that their interlocutor has any in principally possible lexicon, and from those infer the ones that are most likely given their conversation history. There exist, however, exponentially many possible lexicons (viz. 2^n for a lexicon of binary mappings, where n is the lexicon size, Blokpoel et al., 2020)[7].

Which of these (or other) models best explains the relation between interactive repair and pragmatic reasoning is an empirical question. Here we have shown that formal models informed by research on human interaction (Albert and Ruiter, 2018) can bring us closer to an understanding of the cognitive and communicative capacities of interacting people. The question of communicative efficiency is inherently one of computational plausibility. This question is best addressed through careful theoretical analysis as we have shown here. Further modelling can be used to refine our computational understanding of the phenomenon prior to empirical testing (cf. van Rooij and Baggio, 2020).

6 Conclusion

Using theoretical analysis, we showed that a simple form of other-initiated repair can ease the computational burden of pragmatic reasoning and thereby contribute to communicative efficiency. Our models make several simplifying assumptions, so scaling them to other interactional strategies will increase computational demands and perhaps alter the division of labour. Besides offering a proof of concept of how repair can ease the computational demands of communication, our methods pave the way for principled theory-driven analyses of how people balance cognitive and interactional resources in human interaction.

[7]For 30 signals and 20 referents there exist $2^{30\times20} = 1152921504606846976$ possible alternatives lexicons to consider. Even when agents can consider a million alternatives per second, it would take them about 3.6 years to update each time they hear their interlocutor speak.

Acknowledgments

This work is funded by the Netherlands Organisation for Scientific Research (NWO): MB is funded by Gravitation grant 024.001.006 of the Language in Interaction consortium, and MD and MW are supported by Vidi grant *Elementary particles of conversation* (016.Vidi.185.205). We would like to thank the reviewers for their valuable comments.

References

Aashraf M. Abdelbar and Sandra M. Hedetniemi. 1998. Approximating MAPS for belief networks is NP-hard and other theorems. *Artificial Intelligence*, 102(1):21–38.

Saul Albert and Jan-Peter de Ruiter. 2018. Improving Human Interaction Research through Ecological Grounding. *Collabra: Psychology*, 4(1).

Mark Blokpoel, Mark Dingemanse, Marieke Woensdregt, George Kachergis, Sara Bögels, Ivan Toni, and Iris van Rooij. 2020. Pragmatic communicators can overcome asymmetry by exploiting ambiguity. Preprint, Open Science Framework.

Tom Bylander, Dean Allemang, Michael C Tanner, and John R Josephson. 1991. The computational complexity of abduction. *Artificial Intelligence*, 49(1–3):25–60.

Andy Clark. 2006. Material Symbols. *Philosophical Psychology*, 19(3):291–307.

Herbert H. Clark and Edward Schaefer. 1987. Collaborating on contributions to conversations. *Language and Cognitive Processes*, 2(1):19–41.

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

Mark Dingemanse. 2020. Resource-rationality beyond individual minds: the case of interactive language use. *Behavioral and Brain Sciences*, 43:e9.

Mark Dingemanse, Seán G. Roberts, Julija Baranova, Joe Blythe, Paul Drew, Simeon Floyd, Rosa S. Gisladottir, Kobin H. Kendrick, Stephen C. Levinson, Elizabeth Manrique, Giovanni Rossi, and Nick Enfield, J. 2015. Universal principles in the repair of communication problems. *PLoS ONE*, 10(9):1–15.

Robert Downey and Mike Fellows. 1999. *Parameterized complexity*. Springer, Berlin.

Jerry A. Fodor. 2000. *The Mind Doesn't Work That Way: The Scope and Limits of Computational Psychology*. MIT press, Cambridge, MA.

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

Micheal R Garey and David S. Johnson. 1979. *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman, San Francisco, CA.

Edward Gibson, Richard Futrell, Steven T. Piandadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. 2019. How Efficiency Shapes Human Language. *Trends in Cognitive Sciences*, 23(5):389–407.

Jonathan Ginzburg and Raquel Fernández. 2010. Computational models of dialogue. In Alexander Clark, Chris Fox, and Shalom Lappin, editors, *The handbook of computational linguistics and natural language processing*, Blackwell handbooks in linguistics. Wiley-Blackwell, Chichester, West Sussex ; Malden, MA.

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

Herbert P. Grice. 1975. Logic and Conversation. In Herbert P. Grice, editor, *Studies in the Way of Words*, pages 305–315. Harvard University Press.

Pim F. Haselager. 1997. *Cognitive Science and Folk Psychology: The Right Frame of Mind.* Sage, London.

Robert X. D. Hawkins, Michael C. Frank, and Noah D. Goodman. 2017. Convention-formation in iterated reference games. *Proceedings of the 39th Annual Meeting of the Cognitive Science Society*.

Maryellen C. MacDonald, Neal J. Pearlmutter, and Mark S. Seidenberg. 1994. Lexical nature of syntactic ambiguity resolution. *Psychological Review*, 101(4):676–703.

David J. C. MacKay. 2003. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

Maria Otworowska, Mark Blokpoel, Marieke Sweers, Todd Wareham, and Iris van Rooij. 2018. Demons of ecological rationality. *Cognitive Science*, 42(3):1057–1066.

Steven T. Piantadosi, Harry Tily, and Edward Gibson. 2012. The communicative function of ambiguity in language. *Cognition*, 122(3):280–291.

Matthew Purver, Julian Hough, and Christine Howes. 2018. Computational Models of Miscommunication Phenomena. *Topics in Cognitive Science*.

Evan F. Risko and Sam J. Gilbert. 2016. Cognitive Offloading. *Trends in Cognitive Sciences*, 20(9):676–688.

Jan-Peter de Ruiter and Chris Cummins. 2012. A model of intentional communication: AIRBUS (Asymmetric Intention Recognition with Bayesian Updating of Signals). *Proceedings of SemDial 2012*, pages 149–50.

Emanuel A. Schegloff. 1992. Repair After Next Turn: The Last Structurally Provided Defense of Intersubjectivity in Conversation. *American Journal of Sociology*, 97(5):1295–1345.

Emanuel A. Schegloff, Gail Jefferson, and Harvey Sacks. 1977. The Preference for Self-Correction in the Organization of Repair in Conversation. *Language*, 53(2):361–382. ArticleType: primary_article / Full publication date: Jun., 1977 / Copyright © 1977 Linguistic Society of America.

Julian J. Schlöder and Raquel Fernández. 2015. Clarifying Intentions in Dialogue: A Corpus Study. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS-2015)*, London.

Dan Sperber and Deirdre Wilson. 1986. *Relevance: Communication and Cognition*, first edition. Blackwell Publishing.

Paul Thagard and Karsten Verbeurgt. 1998. Coherence as constraint satisfaction. *Cognitive Science*, 22(1):1–24.

Iris van Rooij. 2008. The Tractable Cognition Thesis. *Cognitive Science*, 32(6):939–984.

Iris van Rooij and Giosuè Baggio. 2020. Theory before the test: How to build high-verisimilitude explanatory theories in psychological science. Preprint, PsyArXiv.

Iris van Rooij, Mark Blokpoel, Johan Kwisthout, and Todd Wareham. 2019. *Cognition and Intractability: A Guide to Classical and Parameterized Complexity Analysis*. Cambridge University Press.

Iris van Rooij, Johan Kwisthout, Mark Blokpoel, Jakub Szymanik, Todd Wareham, and Ivan Toni. 2011. Intentional Communication: Computationally Easy or Difficult? *Frontiers in Human Neuroscience*, 5.

Thomas Wasow, Andrew Perfors, and David Beaver. 2005. The puzzle of ambiguity. In O Orgun and P Sells, editors, *Morphology and The Web of Grammar: Essays in Memory of Steven G. Lapointe*, pages 265–282. CSLI Publications.

Deanna Wilkes-Gibbs and Herbert H Clark. 1992. Coordinating beliefs in conversation. *Journal of Memory and Language*, 31(2):183–194.

A Additional Results

This appendix shows the simulation results for all different parameter settings that were run, by way of a robustness check. The parameters that were manipulated are (i) the ambiguity level, (ii) the entropy threshold (i.e. the level of uncertainty that the listener is willing to tolerate) and (iii) the lexicon size.

Figure 3 shows the mean communicative success for the different parameter settings for which simulations were run. For lexicon size 6x4, no data is shown for the agents of type Frugally Pragmatic 1 for the entropy thresholds of 0.8 and 1.0 bits combined with an ambiguity level of 0.8, as in these conditions all frugally pragmatic listeners levelled up to a higher order of pragmatic reasoning (for which the data can be found under Frugally Pragmatic Agents 2). For agents of type Frugally Pragmatic 2 with lexicon size 6x4, no data is available for any of the entropy thresholds combined with an ambiguity level of 0.2, and for the entropy thresholds of 1.0 and 1.5 bits combined with an ambiguity level of 0.5, because the frugally pragmatic listeners never levelled up to second-order reasoning in these conditions. For the Frugally Pragmatic Agents 1 the same happened for the larger lexicons of 15x10 and 30x20 with an ambiguity level of either 0.5 or 0.8 (and for the combination of an ambiguity level of 0.2 with an entropy threshold of either 0.8 or 1.0 bits for the lexicon size of 30x20), as all agents went an order up here as well (i.e., data for these parameter settings is shown under Frugally Pragmatic Agents 2). For the Frugally Pragmatic Agents 2, there is no data for the lexicon size of 15x10, an ambiguity level of 0.2 and entropy thresholds of 1.0 and 1.5 bits, as no agents went up from order 1 to order 2 in these conditions. Finally, the fully pragmatic agents did not have the possibility to move an order up, therefore no entropy threshold was set.

First of all, the expected effect of ambiguity level is visible: the higher the ambiguity level, the lower the communicative success. This holds for almost all conditions, except for the Frugally Pragmatic Agents 1 with a lexicon size of 6x4 and an entropy threshold of 1.5 bits. Here we can see a slight improvement in communicative success when the ambiguity goes up from 0.5 to 0.8, which can be explained by the fact that for a high ambiguity level, these agents decide to go up to order 2 of pragmatic reasoning most of the time, and only stay with order 1 when they are sufficiently certain about the speaker's intended referent. Another exception when it comes to the effect of ambiguity level on the communicative success can be detected for the interactional agents with a lexicon size of 15x10, for the entropy thresholds of 1.0 and 1.5 bits and an ambiguity level of 0.2 and 0.5: here, the interactional agents perform better with an ambiguity level of 0.5 than 0.2. This is due to the fact that an ambiguity level of 0.2 for a lexicon size of 15x10

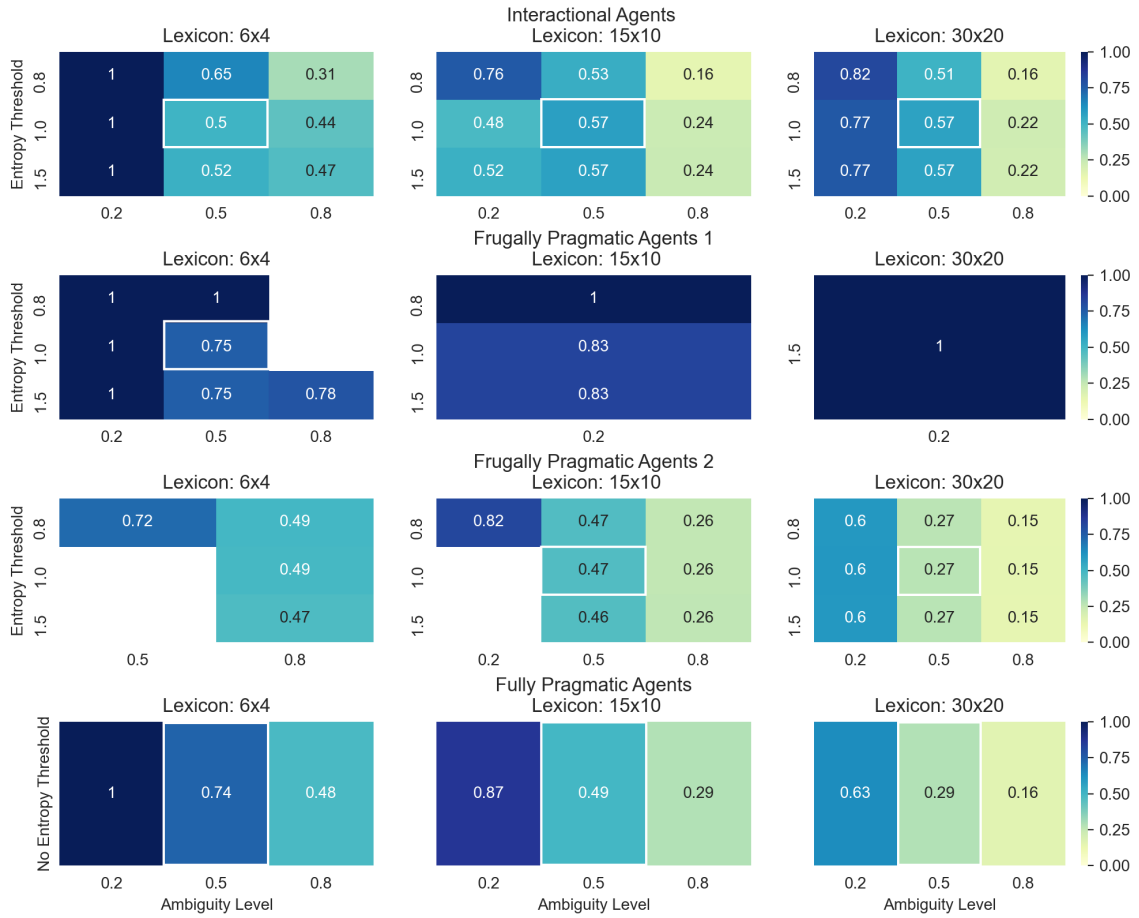

Figure 3: The mean communicative success for the different parameter settings: different agent types, entropy thresholds, ambiguity levels and lexicon sizes. For the frugally pragmatic agents 1, a lexicon size of 6x4, entropy thresholds of 1.0 and 0.8 bits, and an ambiguity level of 0.8 there is no data available as all agents went up on the order of pragmatic reasoning, for which the data is represented below at the frugally pragmatic agents 2 for a lexicon size of 6x4 (because the agents went up from an order of 1 to 2). Notice though that for a lexicon size of 6x4 no data is shown for an ambiguity level of 0.2 or an ambiguity level of 0.5 combined with an entropy threshold of either 1.0 or 1.5 bits, as no agents decided to go up from an order of 1 to 2. Again, for the frugally pragmatic agents 1 for the larger lexicons of 15x10 and 30x20 with an ambiguity level of either 0.5 or 0.8 (and for the combination of an ambiguity level of 0.2 and an entropy threshold of either 0.8 or 1.0 bits for the lexicon size of 30x20), all agents went an order up as well, explaining why no data is shown here. For the frugally pragmatic agents 2, there is no data for the lexicon size of 15x10, an ambiguity level of 0.2 and entropy thresholds of 1.0 and 1.5 bits, as no agents went up from an order of 1 to 2. Finally, the fully pragmatic agents did not have the possibility to move an order up, therefore no entropy threshold was set. The white outlines indicate the simulation results reported in the main body of the paper.

means that every signal refers to 2 referents. Therefore, agents do not use OIR for this ambiguity level as they have already reached the entropy threshold from the start; when the entropy threshold is set to 1.0 bits (or higher), agents are satisfied with having their set of possible interpretations narrowed down to two approximately equiprobable candidates. With a higher ambiguity level the agents do need to use OIR for these entropy thresholds, therefore they can reach an entropy level under the entropy threshold and only have one referent left to choose from in some cases.

Secondly, the entropy threshold is used to manipulate how much uncertainty a listener allows for in a conversation; we ran simulations with three different entropy thresholds: 0.8, 1.0 and 1.5. With an

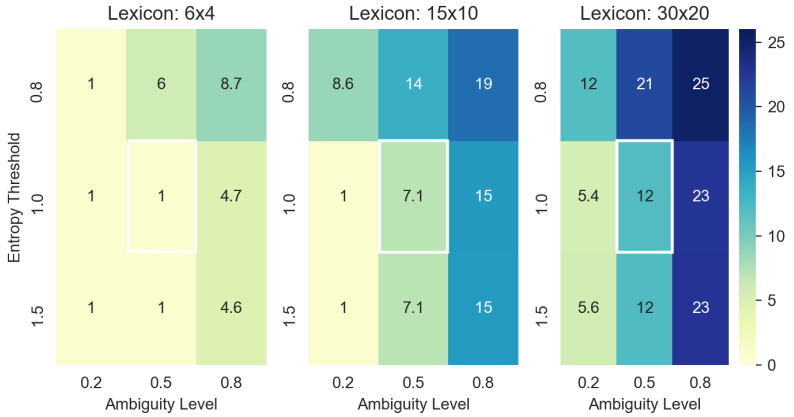

Figure 4: The mean number of turns for the interactional agents for the different parameter settings: different entropy thresholds, ambiguity levels and lexicon sizes. The white outlines indicate the simulation results reported in the main body of the paper.

entropy threshold of 0.8 bits, listeners are quite certain about which referent to choose, as one referent has a higher probability than the others. An entropy threshold of 1.0 bits means that the listener still has to choose between two more or less equally probable referents given a signal. Finally, with an entropy threshold of 1.5 bits, listeners have to choose between three more or less equally probable referents given a signal.[8] For the fully pragmatic agents the entropy threshold does not play a role, as these agents start at the maximum order of pragmatic reasoning ($n = 2$) from the beginning, regardless of their level of (un)certainty. When looking at the results in Figure 3, a clear effect of entropy threshold is not detectable. Overall, we can spot a small effect of the lowest entropy threshold of 0.8 bits leading to a higher communicative success, but this effect is not consistent across conditions and not very visible between the entropy thresholds of 1.0 and 1.5 bits.

Finally, an effect of lexicon size can be seen as well: for bigger lexicons the communicative success tends to be lower than for smaller ones. As discussed in the main body of the paper, this is due to bigger lexicons resulting in more dispersed probability distributions over signals and referents (for speakers and listeners respectively). Furthermore, we can observe that frugally pragmatic listeners go an order up in pragmatic reasoning (thereby entering the Frugally Pragmatic 2 scenario) when the ambiguity level is higher and when the lexicon size is larger, which happens more often for the agents who tolerate less uncertainty (i.e. have a lower entropy threshold). This is in line with our expectations, as bigger lexicons with higher ambiguity levels cause more dispersed probabilities over the referents given a signal. A listener who is uncertain about the speaker's intended referent is more likely to go up on the order of reasoning, and this effect will be stronger if the listener has a lower entropy threshold.

Figure 4 shows the mean number of turns for the interactional agents for the different ambiguity levels and entropy thresholds. These parameters have a clear effect on the number of turns. The higher the ambiguity level, the more turns are used to be certain enough about the speaker's intended referent. Next, the lower the entropy threshold, the more turns are needed to be certain enough (as a lower entropy threshold means that the agent tolerates *less* uncertainty). And finally, regarding the lexicon size: the bigger the lexicon, the more turns are needed to be certain enough, as bigger lexicons lead to more dispersed probability distributions over the referents given the signal(s).

As mentioned above, for lower entropy thresholds, agents want to eliminate more uncertainty (i.e. gain a lower conditional entropy), which they try to achieve by taking more turns. However, we can observe in Figure 4 that there is not a (big) difference in the number of turns that the agents take between an entropy threshold of 1.0 and 1.5 bits, which means that after some turns agents are equally certain for both entropy thresholds (probably both fall under 1.0, regardless of the threshold).

[8] We can make these generalisations based on the definition of conditional entropy, but note that a given conditional entropy value can in principle correspond to a number of different probability distributions.

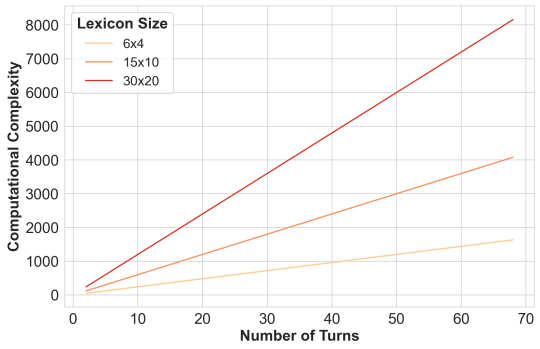

Figure 5: The computational complexity for different numbers of turns for different lexicon sizes, for the interactional agents.

Next, Figure 5 shows the computational complexity for the interactional agents for the different lexicon sizes and numbers of turns. Here, we can see that the computational complexity goes up for bigger lexicons and more turns, as expected.

Finally, Figure 6 shows the difference in conditional entropy over turns for different lexicon sizes for the interactional agents (as the pragmatic agents only perform one turn), meaning that the entropy difference of turn 2 for instance is given by:

$$\Delta_H = H(t_2) - H(t_1)$$

Here we can see that for the smaller lexicon sizes, the entropy difference between turns is smaller. Moreover, we can observe that around 10 turns the entropy does not differ much anymore when taking more turns, meaning that taking more than 10 turns in total (i.e. 5 per agent; including 5 counts of OIR) is not very effective when it comes to the listener's certainty about their interpretation. This is in line with the result discussed in the body of the paper that the interactional agents, regardless of lexicon size, take less than 5 turns most of the time.

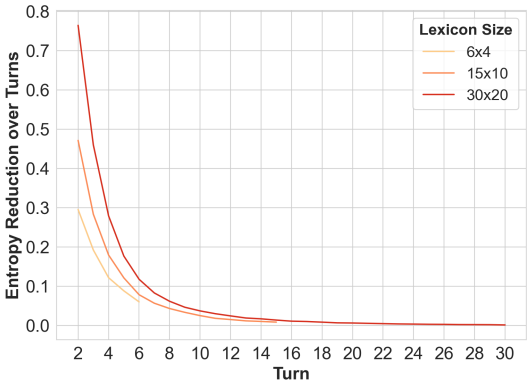

Figure 6: The difference in entropy between the number of turns for different lexicon sizes for the interactional agents.

B Computational Complexity Analysis of Interactional, Frugally Pragmatic and Fully Pragmatic Communication Strategies

B.1 Agent types and computational operations

We distinguish three agent types:

1. **Interactional agents:** Use other-initiated repair and conjunction to get to mutual understanding

2. **Frugally pragmatic agents:** Speaker is always order 2, but listener starts at order 1, and only levels up to order 2 when uncertainty is too high

3. **Fully pragmatic agents:** Both speaker and listener are order 2. Listener's strategy doesn't depend on their uncertainty

Together, these three agent types make use of three different operations of which we can analyse the computational complexity:

1. **Conjunction** (only used by interactional agents)

2. **Entropy** (only used by listeners of the interactional and frugally pragmatic agent types)

3. **Inference** (used by all agent types)

Below, we will analyse the computational complexity of the relevant operations for the three different agent types listed above in turn.

B.2 Computational complexity analysis

Let's start with distinguishing between $|S|$ (number of signals) and $|R|$ (number of referents) in our computational complexity analysis. We can later simplify by subsuming these two variables under a single variable m, which simply takes on whichever value is the maximum out of $|S|$ and $|R|$.

B.2.1 Interactional agents

Interactional speaker The interactional speaker has to do one conjunction step and one inference step per turn.

The conjunction step updates the lexicon L' (which exists only within a particular interaction and is 'reset' to the speaker's original lexicon at the start of each new interaction) by multiplying each value in the column corresponding to $r_{intended}$ in L' with the corresponding value in the signal row corresponding to the signal that was last sent s_{t-1}. This operation is specified below in algorithm 2.

Algorithm 2: Conjunction operation for interactional speaker

Data: L' is a lexicon matrix with $|S|$ rows and $|R|$ columns. $r_{intended}$ is the speaker's intended referent. s_{t-1} is the latest signal in the dialogue history D_r. Note that we assume that at each turn t, L' contains the outcome of the conjunction operation that was performed at the previous turn $t-1$ (if such a previous turn exists). If $t = 1$, L' is identical to the speaker's lexicon L.

Result: An updated lexicon L' on which the conjunction operation has been performed given $r_{intended}$ and s_{t-1}. Note that the only values that are updated in L' are the cells in the column corresponding to $r_{intended}$.

1 **for** $i \leftarrow 1$ **to** $|S|$ **do**

2 $\quad L'_{i,r_{intended}} = L'_{i,r_{intended}} * L_{s_{t-1},r_{intended}}$;

3 **end**

The computational complexity of step 2 in algorithm 2 is $|S|$ (i.e. the multiplication operation has to be done exactly once for each cell in the column corresponding to $r_{intended}$).

Algorithm 2 has to be performed exactly once for each turn $t > 1$ after the first turn. Therefore, this conjunction operation has to be performed exactly $t-1$ times in a given interaction. Thus, the overall computational complexity of conjunction for the interactional speaker is $|S|(t-1)$.

To determine which signal to send next, the speaker has to go along the column of their intended referent, and select the signal that has the highest value. Given that every agent type has to do this inference step, let's assume that lookup is free. In that case the speaker has to make $|S|$ comparisons to check which signal has the highest value.

Taken together, this means that the computational complexity for the interactional speaker strategy as a whole (per interaction) is $|S|(t-1) + |S|$.

Interactional listener In addition to the conjunction and inference steps, the interactional *listener* has to do an entropy step in between, to decide whether to move on to the inference step (if entropy is low), or whether to respond with a repair initiator (if entropy is high).

Let's again go through the steps in order:

The conjunction step updates the lexicon L' (which exists only within a particular interaction and is 'reset' to the listener's original lexicon at the start of each new interaction) by multiplying the signal row corresponding to the first signal that was received in the interaction $s_{t=1}$ with the signal row corresponding to the signal that was last received s_{t-1}. This operation is specified below in algorithm 3.

The computational complexity of step 2 in algorithm 3 is $|R|$ (i.e. the multiplication operation has to be done exactly once for each cell in the row corresponding to signal $s_{t=1}$).

Just like for the interactional speaker, algorithm 3 has to be performed exactly once for each turn $t > 1$ after the first turn. Therefore, this conjunction operation has to be performed exactly $t-1$ times in a given interaction. Thus, the overall computational complexity of conjunction for the interactional listener is $|R|(t-1)$.

At each turn of the interaction (including the very first turn), the listener does an entropy step to check how certain they are about their inference over possible intended referents. The entropy of the probability distribution over referents given the received signal s and the lexicon updated according to the dialogue history L_{D_r} (what is called L'

Algorithm 3: Conjunction operation for interactional listener

Data: L' is a lexicon matrix with $|S|$ rows and $|R|$ columns. s_{t-1} is the latest signal in the dialogue history D_r. Note that we assume that at each turn t, L' contains the outcome of the conjunction operation that was performed at the previous turn $t-1$ (if such a previous turn exists). If $t = 1$, L' is identical to the listener's lexicon L.

Result: An updated lexicon L' on which the conjunction operation has been performed given $s_{t=1}$ (the first signal in the interaction) and s_{t-1} (the latest signal in the interaction). Note that the only values that are updated in L' are the cells in the row corresponding to $s_{t=1}$ (i.e. the signal that was received in the very first turn of the interaction).

for $i \leftarrow 1$ **to** $|R|$ **do**
  $L'_{s_{t=1},i} = L'_{s_{t=1},i} * L'_{s_{t-1},i}$;
end

above) is given by equation 15.

$$H(R|s, L_{D_r}) = \sum_{r \in R} Pr(r|s, L_{D_r}) log_2 \frac{1}{Pr(r|s, L_{D_r})} \quad (15)$$

Thus, the listener has to first do a multiplication operation for each $r \in |R|$, and then sum each of the $|R|$ resulting values together. This means that the computational complexity of the entropy calculation is $2|R|$. Because the entropy calculation happens at every single turn, the overall computational complexity of the entropy operation for a given interaction is $2|R|t$.

Once the entropy falls below the entropy threshold, or once the cap on the number of turns has been reached, the listener will decide to move to the inference step to actually interpret the signal(s). In order to do that, the listener has to go along the row corresponding to the first signal that was sent $s_{t=1}$ and select the referent that has the highest value. To do this, the listener has to make $|R|$ comparisons.

Taken together, this means that the computational complexity for the interactional listener strategy as a whole (per interaction) is $|R|(t-1) + 2|R|t + |R|$.

B.2.2 Pragmatic agents

We can consider the RSA operation of updating the matrix of production/reception probabilities separately from the inference step of choosing an actual utterance or interpretation. The computational complexity of that RSA step by itself (for speakers and listeners alike) is $(2 + 4n)|S||R|$, where n is the order of pragmatic reasoning.

The idea behind this is that a pragmatic agent has to normalise $2n$ times (first along the rows and then along the columns for production, or first along the columns and then along the rows for interpretation; and that n times). Each normalisation step itself takes $2|S||R|$ (taking the sum over rows or columns takes $|S||R|$ steps, then dividing each cell by the relevant sum also takes $|S||R|$ steps). Taken together, this makes $2n \cdot 2|S||R| = (4n)|S||R|$ steps. However, we haven't yet incorporated the first normalisation step which turns the lexicon of binary mappings into a level-0 speaker (in the case of pragmatic production) or a level-0 listener (in the case of pragmatic interpretation). As explained above, this initial normalisation operation consists of $2|S||R|$ steps, so if we add it in, we get: $(2 + 4n)|S||R|$.

If we combine this with the inference step (which, as we saw above, is $|S|$ for speakers, and $|R|$ for listeners), we get $(2 + 4n)|S||R| + max(|S|, |R|)$. Which is a generic computational complexity analysis for pragmatic agents in general. But we can make this more specific by considering speakers and listeners separately, as we do below.

Frugally pragmatic speaker The frugally pragmatic speaker strategy is exactly the same as the fully pragmatic speaker strategy; see the corresponding complexity analysis below under 'Fully pragmatic speaker'.

Frugally pragmatic listener For the frugally pragmatic listener, the computational complexity of this strategy depends on whether the listener decides to level up to order 2 or not.

- **Scenario 1:** In this scenario, the listener *doesn't* level up, which means that $n = 1$. This yields: $(2 + 4n)|S||R| = (2 + (4 * 1))|S||R| = 6|S||R|$ for the RSA operation. This is then combined with the entropy calculation, which, as shown above, takes $2|R|$

	Interactional	Frugally pragmatic	Fully pragmatic
Speaker	$\|S\|(t-1)+\|S\|$	$10\|S\|\|R\|+\|S\|$	$10\|S\|\|R\|+\|S\|$
Listener	$\|R\|(t-1)+2\|R\|t+\|R\|$	**1:**$6\|S\|\|R\|+2\|R\|+\|R\|$ **2:**$10\|S\|\|R\|+2\|R\|+\|R\|$	$10\|S\|\|R\|+\|R\|$

Table 4: Computational complexity comparison across agent types (with speaker and listener strategy summed together).

	Interactional	Frugally pragmatic	Fully pragmatic
Speaker	$m(t-1)+m$	$10m^2+m$	$10m^2+m$
Listener	$m(t-1)+2mt+m$	**1:**$6m^2+2m+m$ **2:**$10m^2+2m+m$	$10m^2+m$

Table 5: Computational complexity comparison across agent types (with speaker and listener strategy summed together).

steps.
And finally, the listener has to do an inference step to come to an actual interpretation. As shown above, this takes $|R|$ steps.
Taken together, this means that the computational complexity for the frugally pragmatic listener who *doesn't* level up is $6|S||R| + 2|R| + |R|$.

- **Scenario 2:** In this scenario, the listener *does* level up, which means that $n = 2$. This yields: $(2 + 4n)|S||R|$ = $(2 + (4 * 2))|S||R|$ = $10|S||R|$ for the RSA operation. (Note that this subsumes the initial RSA operation at order $n = 1$; we assume that the listener can hold on to the outcome of that first $n = 1$ operation to use it as the basis for their subsequent $n = 2$ inference.)
  This is then combined with the entropy calculation, which, as shown above, takes $2|R|$ steps.
  And finally, the listener has to do an inference step to come to an actual interpretation. As shown above, this takes $|R|$ steps.
  Taken together, this means that the computational complexity for the frugally pragmatic listener who *does* level up is $10|S||R|+2|R|+|R|$.

Fully pragmatic speaker The frugally pragmatic speaker and fully pragmatic speaker strategies are exactly the same: they both start at order $n = 2$, and don't do anything other than regular RSA production. The RSA operation part for order $n = 2$ is $(2 + 4n)|S||R|$ = $(2 + (4 * 2)|S||R|$ = $10|S||R|$.

As shown above, the inference step for production takes $|S|$ steps.

Taken together, this means that the computational complexity for the frugally *or* fully pragmatic speaker is $10|S||R| + |S|$.

Fully pragmatic listener The fully pragmatic listener starts at order $n = 2$ straight away, and doesn't do any entropy calculation.

Taken together, this means that the computational complexity for the fully pragmatic listener is $10|S||R| + |R|$.

B.2.3 Comparison across agent types

Table 4 shows a comparison of the computational complexity of each strategy.

We can make the computational complexity of the speaker and listener roles more comparable by subsuming $|S|$ and $|R|$ under one combined variable $m = max(|S|, |R|)$, which simply takes on whichever is the highest value out of $|S|$ and $|R|$. (Given the parameter settings used in the simulations, this will always be $|S|$, given that $|S|$ was fixed at $1.5 \times |R|$.)

Division of labour between speaker and listener From Table 5 we can read off how the division of labour between speaker and listener differs between the different strategies. Only in the fully pragmatic agent types do speaker and listener do an exactly equal amount of work. In the interactional strategy,

Interactional	**Frugally pragmatic**			**Fully pragmatic**
$2m(t-1)+2mt+2$	**1:**$16m^2$ **2:**$20m^2+4m$	$+$	$4m$	$20m^2+2m$

Table 6: Computational complexity comparison across agent types (with speaker and listener strategy summed together).

the listener always does a bit more work than the speaker (because the listener thinks about how uncertain they are about their inference; the speaker in contrast only has to react when they get a repair request). In the frugally pragmatic strategy, the listener does *less* work than the speaker when they can stay at order $n = 1$ (scenario 1), but *more* work than the speaker when they have to level up to order $n = 2$ because their initial inference was too uncertain (scenario 2).

Comparison across agent types (collapsing across speaker and listener role) Ultimately however, we are interested in comparing across agent types. In order to do this, we can sum the complexity of the speaker and listener within each agent type together, as shown in Table 6.

As described in Section 4, we used the formulas in Table 6 in combination with the mean number of turns derived from the simulations for each separate strategy to yield the computational cost results shown in Figure 2c.