

Towards Neuro-Symbolic Models of Language Cognition: LLMs as Proposers and Evaluators

Polina Tsvilodub (polina.tsvilodub@uni-tuebingen.de)

Department of Linguistics, University of Tübingen

Fausto Carcassi (f.carcassi@uva.nl)

Institute of Logic, Language and Computation, University of Amsterdam

Michael Franke (michael.franke@uni-tuebingen.de)

Department of Linguistics, University of Tübingen

Abstract



Building explanatory computational models of pragmatic language use has been a long-standing goal in cognitive science and pragmatics (e.g., Frank and Goodman (2012); Kramer and van Deemter (2012); Ferreira (2019)). However, extant models are often limited to explaining selection from pre-specified sets of utterances or interpretations. To address this limitation, we explore a neuro-symbolic approach to modeling production of more domain-independent, open-ended language based on a classical cognitive model by Dale and Reiter (1995). This approach combines LLM-powered modules which supply in-principle unrestricted inputs and processing for context-dependent components of the task, with structured reasoning modules within an architecture implementing cognitive structure. We show that this hybrid approach can model substantially open-ended language production for complex contexts from the A3DS dataset. Our neuro-symbolic model is at the same time cognitively plausible and outperforms a few-shot GPT-3.5-turbo baseline.

Keywords: cognitive modeling, language production, referential expressions, large language models, hybrid models

Introduction

The ability to flexibly and efficiently communicate across an infinity of different contexts, while using limited cognitive and linguistic resources is a distinctly human capacity. Identifying and explaining the cognitive mechanisms and reasoning processes underlying such language generation and interpretation has been a long-standing goal of cognitive science. *Computational cognitive modeling* is one of the main tools that has allowed to describe, study and explain many language-related phenomena, including human word learning (Xu & Tenenbaum, 2007), sentence processing (Levy, Reali, & Griffiths, 2008) and production (Levelt, 1999), as well as pragmatic language use (Frank & Goodman, 2012).

Despite substantial progress, some fundamental features of linguistic cognition are challenging to formal modeling. Models often need to include various information that the agent might recruit in their linguistic behavior, e.g., her own beliefs, desires, intentions, and expectations (Clark, 1996; McRae & Matsuki, 2009; Lupyan & Clark, 2015; Rohde, Futrell, & Lucas, 2021, among others). A first strategy to solve this problem is to attempt a direct specification of all information that could be relevant to perform a communicative task in its full generality. However, even comparatively simple tasks like establishing reference to an object in context is effectively unrestricted in the number of situations in which it can occur.

Additionally, general world knowledge might turn out to be relevant for a communicative task, and reasoners might invoke very different representations for different inputs, even while solving the same task. For example, suppose that you have the task to pick out the first of the two following symbols: , . You might say “The potato-like object” and a prototypical speaker of English will likely be able to identify the intended referent, despite the fact that nothing about the picture itself makes a connection to potatoes. Rather, the utterance choice is based on knowledge of the associations interlocutors could make upon seeing the images, which goes beyond the images themselves and involves world knowledge. Finding an appropriate utterance in this case requires information that is supplied “on the fly” by general cognition, but which cannot generally be anticipated a priori for any instance of the task, e.g., picking out one image out of two options. In these cases, manually specified spaces of representation will fall short of being applicable in the open-ended variety involved in human communication without the need to change the model specification, even for small input changes.

Given that the exhaustive specification strategy is not promising for modeling language cognition, a second common strategy is to restrict both the input that the model can deal with and the information that the model can recruit in solving the task, under the assumption that the included scenarios are representative of the task in general and, therefore, that the chosen restrictions allow for an unbiased assessment of the resulting model. For instance, when modeling the domain of pragmatic interpretation in a reference game, only a few available expressions and contexts might be included under the assumption that no qualitative shift in mental manipulation will happen for other contexts or expressions (Frank & Goodman, 2012, among others). The restriction is generally done either by manually specifying the domain for the relevant variables or by running experiments to elicit relevant judgments from human participants.

This second strategy might pose various problems. Practically, manually specifying a space of relevant items is laborious and running experiments to crowd-source these specifications is costly. Further, a manual specification increases the degrees of freedom for the modeler generally beyond what is informed by their theoretical commitments, leaving them to make more or less arbitrary choices about what might become

relevant for the modeled phenomenon. This might lead to biased results even in the absence of such intentions (Chambers, 2017).

In this paper, we consider a third strategy to get around manual specification in cognitive modeling. We explore a neuro-symbolic architecture which combines neural modules performing the open-ended subtasks within a manually specified task decomposition. We study the possibilities offered by this approach for modeling less restricted language use than under previous approaches. Specifically, we consider pragmatic language generation as a case study of an open-ended linguistic cognitive task, and we use a Large Language Model (LLM) as a black-box stand-in model for the components of the cognitive model which traditionally require manual restriction (e.g., the space of utterances that can be used).

The use of LLMs in this context is particularly natural, since recent LLMs generate impressively fluent natural language across contexts and tasks (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023). Further, they are trained on a huge amount of text and can serve as an approximation of the world knowledge that could become relevant in utterance production (Wray, III, Kirk, & Laird, 2021; Zhang, Lehman, Stanley, & Clune, 2023). Recent work has already employed LLMs as components in more complex computational systems. For instance, several advanced prompting strategies for LLMs use search procedures (Yao et al., 2023), condition the generation on additional information (J. Liu et al., 2022), or make use of additional components for, e.g., math problems (He-Yueya, Poesia, Wang, & Goodman, 2023), complex reasoning (Creswell, Shanahan, & Higgins, 2022; He-Yueya et al., 2023; Paranjape et al., 2023; Poesia, Gandhi, Zelikman, & Goodman, 2023) or programming tasks (Gao et al., 2022). However, the primary focus of these systems is often improving LLM performance on particular tasks where simple prompting achieves worse results. Other work has built LLM-powered *agents* for simulating human behavior (Park et al., 2022), e.g., by combining cognitively inspired LLM components (e.g., for planning and memory, Park et al. (2023)). Wong et al. (2023) leverage LLMs in combination with a probabilistic language of thought for exploring human-like commonsense reasoning.

In sum, we explore a concrete case study, similar to previous cognitively inspired work such as the *cognitive language agents* of Sumers, Yao, Narasimhan, and Griffiths (2023). Additionally, we focus on exploring the potential and usefulness of LLMs as a building block in computational cognitive modeling which translates theoretical insights from cognitive science to models which allow model testing and model comparison in light of empirical evidence (see Frank (2023) for related, more general discussion).

Our choice of case study—pragmatic language generation in the context of a reference game—offers several advantages for this exploration. First, context-dependent natural language use is a particularly natural application for our LLM-based strategy because it employs the LLM components on

a domain that they have been primarily trained for. Second, it allows direct comparison and careful evaluation in light of long-standing work in cognitive science and linguistics. Finally, it offers an avenue for direct assessment of the neuro-symbolic modeling strategy when applied to established cognitive models, as we describe in the next section. In the final discussion we give some pointers to how the insights gained from our model could generalize to language cognition more generally.

The rest of the paper is structured as follows. First, we present a proof-of-concept model of contrastive utterance generation building on the model by Dale and Reiter (1995) in order to explore whether and how LLMs might be used in order to extend the cognitive science toolbox. Then, we test this model in a concrete task and compare it to an LLM baseline and a simpler lesioned model.

Models of Contrastive Utterance Generation

We focus on the task of *contrastive utterance generation* (Kramer & van Deemter, 2012). In this task, the speaker aims to produce an utterance that uniquely identifies a *target* state among a set of possible alternative *distractor* states from which the target should be distinguished. This task is a particularly natural testing ground for our idea. First, it has been thoroughly investigated from different perspectives. Probabilistic modeling of utterance selection has served as a prime case study within computational pragmatics (Frank & Goodman, 2012; Golland, Liang, & Klein, 2010; Hawkins, Frank, & Goodman, 2020; Franke & Degen, 2016). Despite its simplicity, performance in this task is related in complex ways to human pragmatic skills. Typically, language users aim to avoid unnecessarily long utterances, and systematically exploit pragmatic reasoning to convey the intended content while keeping utterances brief (Grice, 1975). Further, various approaches to generating referential expressions have been considered in computational linguistics (Gatt & Krahmer, 2018; Krahmer & Van Deemter, 2012; Dale & Reiter, 1995, among others).

According to a popular algorithmic idea, utterances (including contrastive referential expressions) are constructed not in a single pass, but rather in an *iterative* fashion (Newell & Simon, 1972; Ferreira, 2019). A speaker would start by generating simple utterances, and then run a loop that alternates an evaluation step, where the current utterances are judged with respect to the task at hand, and a generation step, in which the currently best utterances are enriched. The loop continues until an utterance is found that solves the task.

However, a cognitive model of contrastive utterance generation based on the iterative picture that can manipulate open-ended language is still lacking. This is in part because such an algorithm requires specifying the utterance evaluation mechanism and the way they are enriched in the loop. These components depend on the context in an open-ended way, and in previous implementation of this as the Iterative Algorithm (IA) by Dale and Reiter (1995) they had to be hand-specified

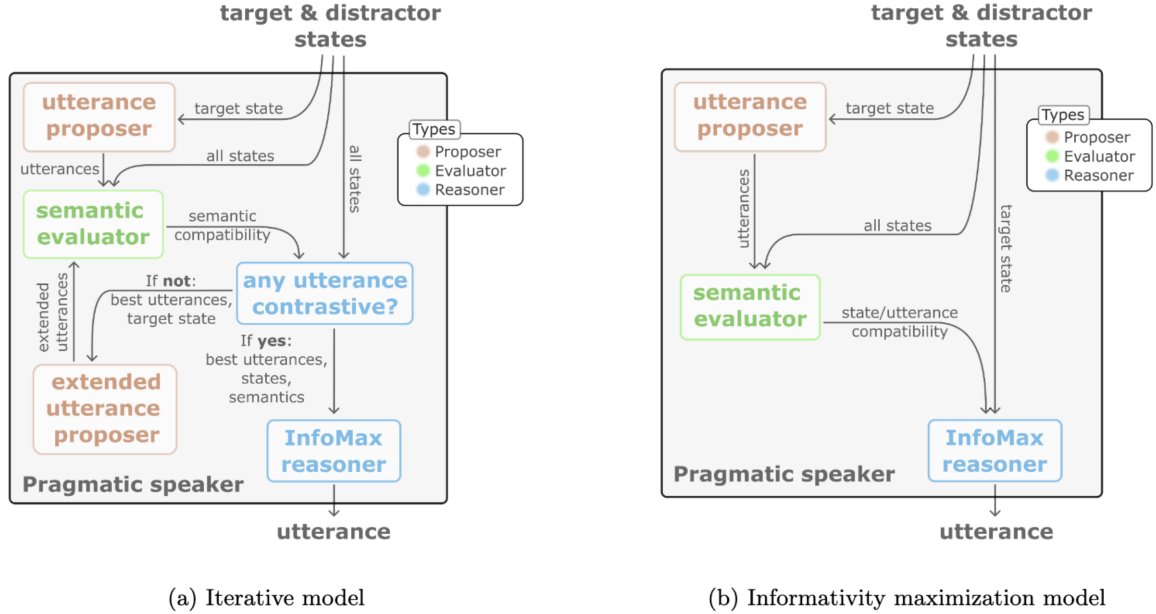


Figure 1: Side-by-side illustration of two models for contrastive utterance generation. In both models, an utterance proposer initially generates utterances for the target state, which are evaluated for truth by a semantic evaluator for all states in the context (target and distractors). At this point the two models diverge. In the IMM (b), the informativity maximizer module then directly selects an utterance given the semantic evaluator’s output. In the IM (a), the semantic evaluation is passed to a contrastivity evaluator. If any utterance is contrastive, the informativity maximizer selects an utterance, otherwise the most contrastive utterance is extended by an extended utterance proposer and passed back to the semantic evaluator. This cycle repeats until a contrastive utterance has been found. In sum, the IM can be thought of as an extension where the model dynamically evaluates the produced utterances and improves them in a loop until the task is solved (or a maximum number of five iterations is reached).

for each application. This makes this task a natural testing ground for the neuro-symbolic strategy we discussed above, by including LLMs as components for sampling information that would normally be hand-specified.¹ Furthermore, it highlights that neuro-symbolic cognitive models might consist of different generalizable types of modules. We categorize three kinds of modules. First, *evaluators* provide context-dependent assessment of alternatives, *proposers* supply these possible alternatives or contingencies (e.g., plausible utterances for a given context, plausible interpretations of an utterance; cf. Sumers et al. (2023)), and *reasoners* combine and process information supplied by the other two types of modules. Given their context-dependent nature, the first two types might often be neural, while the last type can be symbolic.

In the following, we first provide an *iterative model* within this framework which directly implements the IA. Then, we compare it to a simpler model which generates the contrastive expressions in a single pass.

Iterative Model

We implement the iterative algorithm (IA) for a reference game by combining both symbolic computations and LLMs

¹For all LLM components and the baseline, GPT-3.5 (gpt-3.5-turbo, checkpoints of summer 2023) with temperature $\tau = 0.1$ was used.

as described above, which we refer to as the *iterative model* (IM; Figure 1 (a)). The context consists of a target state and one or more distractor states. First, a set of utterances is proposed describing a single feature of the target state. On each iteration, the model evaluates the contrastivity of the candidate utterances generated so far. If none of the utterances is contrastive, the model selects the most informative utterances available, adds some detail of the target, and starts a new iteration.² This is repeated until an utterance is found that solves the task. The utterances are generated, extended, and evaluated by calls to an LLM with appropriate prompts, but the iterative structure is encoded in the model architecture as displayed in Figure 1 (a).

More technically, IM takes as input a list of full state descriptions, one of which is the target state and the remaining ones are distractors. First, the (LLM-based) *single detail utterance proposer* generates candidate utterances that describe a single detail of the target state based on the target state description. Second, the *semantic evaluator* determines the (literal) truth value of all candidate utterances for each state

²Note that distractors are not taken into account when extending the description of the target. Instead, we consider only the utterances produced so far along with the speaker’s background knowledge about the target (i.e., all attributes that are true of the target). This potentially allows the model to be applied to non-contrastive utterance generation tasks in future research.

(target and distractors). Third, based on the semantic evaluation in the previous step, the *contrastivity reasoner* evaluates the contrastivity of the generated utterances, and determines whether any utterance is fully contrastive (i.e., only true of the target). If so, one utterance is chosen among the contrastive utterances by the *infomax* (*informativity maximization*) reasoner and returned. Otherwise, a set of the most contrastive utterances is constructed, and the (LLM-based) *extended utterance proposer* module produces new alternatives for each utterance, each of which includes one more detail from the full description. The loop repeats from the semantic evaluation on until a fully contrastive utterance is produced or the maximal iteration steps have been reached, in which case the *infomax reasoner* greedily selects the most contrastive among the utterances produced in the most recent loop.

An exhaustive search over minimal contrastive expressions becomes computationally intractable with increasingly complex contexts (Krahmer & Van Deemter, 2012). To get around this problem, the IM implements a search over a (partial) tree of possible referential utterances. Only a single detail of the target is added on each iteration, in an order proposed by an LLM module rather than a manually-specified order as in Dale and Reiter (1995). Therefore, the tree depth roughly corresponds to the number of details included in the utterance, while the width corresponds to the number of sampled utterance proposals on each loop. If the number of proposals by the utterance proposers is less than the number of features in the scene, not all possible utterances will be considered. Moreover, the search is greedy, in the sense that only the most contrastive utterances are passed to the following iteration. The algorithm therefore ensures that utterances are considered in order of the amount of details they describe.

Informativity Maximization Model

The IM presented above can be naturally lesioned by removing the iterative part, resulting in what we call the *Informativity Maximization Model* (IMM; Figure 1 (b)). In this case, the first batch of produced and evaluated utterances is used by the *infomax* reasoner, rather than iteratively improved upon. This is a simple way of generating an utterance for solving the contrastive reference task, but it does not adapt to the complexity of the task in context. The context is identical to the IM, but in contrast to the first loop of the IM the utterances of the IMM are not constrained to contain a single detail. The compatibility of each utterance is evaluated on all the distractor states in the context and the model selects the utterance which is applicable to as few distractors as possible and therefore is most informative.

As for the IM, the input to the model is a list of state descriptions, including the target state and one or more distractors. The model proceeds in three steps (Figure 1 (b)). First, an *utterance proposer* module uses an LLM call to generate candidate utterances for the target state based on the target state description. Second, a *semantic evaluator* module determines, again via an LLM call, the truth value of each candidate utterance for each state (target and distractors). Lastly,

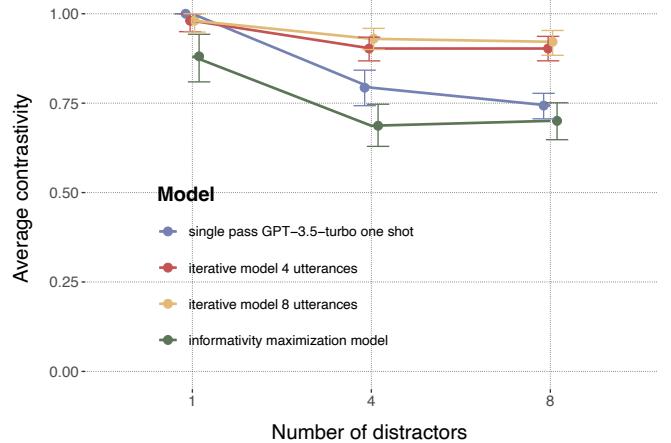


Figure 2: Reference game results: distribution over contrastivity values (y-axis) by number of distractors (x-axis) and number of utterances proposed (color). Error bars show bootstrapped 95%-CIs.

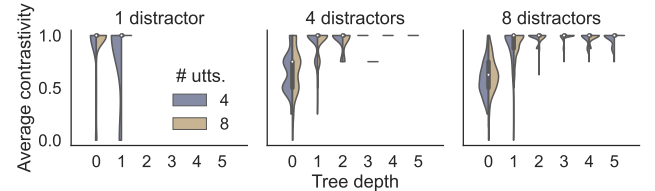


Figure 3: Development of task success over increasing tree depth in the IM: distribution over contrastivity values (y-axis) over increasing tree depth (extended utterance proposal and evaluation iterations; x-axis), by number of distractors (facets) and tree width (number of proposed utterances; color). Dots indicate means, thick bars indicate quartiles, thinner lines indicate minimal values.

an *infomax reasoner* module selects the most specific true utterance.

The model solves the task when an utterance is generated that uniquely identifies the target in the context of the distractors. This is made more difficult by the fact that utterances are generated in a single pass, with the only criterion of being true of the target. This restriction not only decreases the model performance, but is cognitively implausible. Experimental work has shown that humans adjust the level of granularity of produced referential expressions depending on the context (Graf, Degen, Hawkins, & Goodman, 2016; Degen, Hawkins, Graf, Kreiss, & Goodman, 2020).

Experiments

We test the models described above on a contrastive reference game (Lewis, 1969) with the 3Dshapes dataset (Burgess

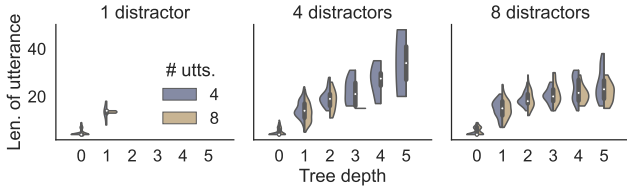


Figure 4: Length of optimal utterances over increasing tree depth in the IM: distribution over number of words (y-axis) over increasing tree depth (extended utterance proposal and evaluation iterations; x-axis), by number of distractors (facets) and number of proposed utterances (color). Dots indicate means, thick bars indicate quartiles, thinner lines span minimal and maximal values.

& Kim, 2018).³ We use a text-based derivative of the dataset, A3DS (Tsvilodub & Franke, 2023), which contains textual descriptions of scenes consisting of a 3D geometric object in an otherwise empty room. The states in the derived dataset are unique and consist of a combination of values for the following attributes:

- Wall color, floor color, object color (independent)
red | orange | yellow | green | blue | pink | purple
- Object position (relative to background)
middle | right corner | left corner
- Object size
small | medium-sized | large
- Object type
ball | cylinder | block | pill

for a total of 12348 possible states. The state descriptions used as context on which the models were tested were of the form “The floor is {floor color}, the wall is {wall color}, the {object color} {size} {object} is in the {position}”.⁴ While not open-ended, this large state-space constitutes a challenge in the contrastive utterance generation task, while enabling automatic evaluation of contrastivity. In particular, the structured nature of the dataset allows to automatically determine which features are contrastive, i.e., different for the target scene in any given context, so that the contrastivity of a generated expression can be verified by checking if it mentions contrastive features.

Baseline

We compare the results of the two models above with a baseline model. The baseline consists of a single call to an LLM asking for an utterance that solves the task. Following recent results showing that LLM performance is improved with examples as well as instructions about the reasoning which help

³All materials can be found at [will be added in de-anonymized version of paper].

⁴We also tested supplying state descriptions in the form of lists of “{feature-value}”. However, the unnatural formatting led to poor performance of the LLMs.

solving the task (Wei et al., 2022, among others), we use a one-shot chain-of-thought prompt for the baseline.

Simulation Procedure

Each reference game included a target state sampled at random and one or more distinct distractors. Each distractor differed from the target by maximally two features, which made the identification of contrastive features more difficult. We set the number of distractors to one, four, or eight. Moreover, for the IM we set the number of utterances sampled by the utterance proposer to either four or eight utterances. For the IMM, the utterance proposer always sampled ten utterances. We tested both models as well as the baseline on 100 reference games for each of the parameter configurations.

We evaluated the *contrastivity* of the final returned utterance, i.e., the accuracy of the model in each reference game. That is, we calculated the proportion of distractors against which the target was set apart by the utterance. This was done via the evaluation script from Tsvilodub and Franke (2023). For instance, assuming that the target appeared in the context of four distractors and that the generated utterance was true of the target and one distractor, the accuracy for this reference game would be 0.75. We compared the performance of the hybrid models to the performance of the baseline.

Results

Results are shown in Figure 2. We computed the average accuracy across references games in one configuration (i.e., for each number of distractors) for both models and the baseline. The performance of the IMM decreased as the number of distractors in the context increased. A bootstrapping analysis revealed that the performance of the IMM was credibly worse than all other models, across the number of distractors ($P = 1$). However, in contrast to the LLM baseline, Figure 2 suggests a trend towards a stabilization of the contrastivity with higher numbers of distractors for the IMM, while the LLM baseline performance decreased.

In contrast to the IMM, the iterative model (IM) generated highly contrastive utterances which successfully set apart the target from the distractors (Fig. 2). The contrastivity of utterances produced by the IM remained high even with a large number of distractors, outperforming the baseline and the IMM (see Fig. 2, eight distractors). Bootstrapping confirmed that the average contrastivity of the IM for four and eight distractors was above the baseline ($P = 1$), while for one distractor, no significant difference was observed due to a ceiling effect.

The number of iterations required in the IM until fully contrastive utterances were produced increased with the number of distractors and thus with the difficulty of identifying contrastive features (Figure 3). In particular, this shows that the IM increased the complexity of the computation and of the generated utterances in a context-dependent way. Furthermore, Figure 3 suggests a slight trade-off between the tree width and tree depth required for producing contrastive utterances (x-axis vs. color): when more utterances were

proposed at each step, it was more likely that at least one of them mentioned contrastive features, so that fewer iterations were required overall. Figure 4 suggests that the IM produced context-sensitive pragmatic contrastive expressions which were shorter and, therefore, as an approximation, mentioned less scene features in a simpler context (one distractor) than in more complex contexts (four or eight distractors). Manual inspection of the model outputs revealed that, when approaching the maximal number of iterations, utterances sometimes became more descriptive. In some cases the utterances repeated single features within the sentence, but mostly they contained additional information or reformulations of the partial description passed to the extended utterance proposer.

Discussion

In this paper, we focused on a promising strategy towards open-ending computational cognitive models of linguistic cognition via neuro-symbolic architectures, focusing on contrastive utterance generation as a case study. We used this architecture to implement an iterative model based on the IA (Dale & Reiter, 1995) for a reference game setting. The model adapts the generated utterances to the complexity of the task at hand, while producing open-ended language that is natural in context. We compared the iterative model (IM) to a simple model (IMM) that produces utterances in a single pass by describing the target scene and then evaluating the informativity of the proposed utterances in context. We found that the IM outperforms both the simple model and a pure LLM baseline in a case study with the A3DS dataset.

The case study we consider gives us some insights into the more general neuro-symbolic approach to cognitive modeling we consider. We observed that the LLM-based proposers worked well and provided plausible samples. However, despite improved performance, the IM was not perfect in that for some reference games, utterances were extended further even in the presence of necessary contrastive features (cf. ceiling effect / asymptotic shape in Figs. 3, 4). We hypothesize that this is due to the limitations of the LLM-based semantic evaluator which provided information for the stopping decision. Improvement of this component opens up an exciting avenue for future work towards integrating LLMs in full probabilistic cognitive models. This would allow for quantitative model comparison, e.g., with respect to experimental human data.

Contrastive utterance generation, though a natural first step to apply the strategy we explore in this paper, is only one case study. Nonetheless, it points to a general strategy for developing computational models of linguistic cognition and related domains that involve open-ended resources. First, a cognitive task is analyzed as a series of processing steps broken down as much as can be informed by theoretical considerations. This analysis will normally leave some steps not analyzed enough for a fully decomposed implementation. For instance, in the IM we presented above we did not have an

algorithmic account of how the truth of an utterance is determined in a context. Instead of manually specifying in advance their behavior as has traditionally been done, we can capture these steps as information processing modules described in natural language, given inputs from other steps in our analysis. Finally, we can offload these task descriptions to LLMs or other neural modules.

This strategy presents several challenges. First, the strategy inherits problems with LLMs in general, such as excessive sensitivity to apparently minor changes to the prompt, uninterpretability, so-called hallucinations and biases (Bender, Gebru, McMillan-Major, & Shmitchell, 2021; Ji et al., 2023; N. F. Liu et al., 2023; Shi et al., 2023; Zhao et al., 2023). Second, the specific strategy we propose leads to using LLMs for open-ended tasks, which often instantiate intractable problems. Previous literature has shown that such problems cannot be solved, even approximately, by LLMs (van Rooij et al., 2023). However, since we do not use the LLMs themselves as explanatory components in the model, we do not need them to model humans across all possible scenarios. Rather, it is sufficient for our strategy that the task is solved for the cases on which the model is tested—for typical inputs, this can be achieved by LLMs and give better empirical coverage than would be possible with hand-specified modules.

The modeling strategy we propose also has various advantages. First, it provides cognitive models which are as informed by the modeler’s theoretical commitments as possible, and no more. For instance, we showed above how we could build a model of utterance generation without an algorithmic picture of truth evaluation. While LLMs are not theory free, they are our best data-based approximation to general problem solvers, and offer the modeler fewer degrees of freedom in modeling. Second, this strategy allows the models to deal with more open-ended input than has been possible so far, and therefore the resulting models can be tested on a wider variety of inputs. For instance, while the A3DS dataset does have a constrained set of states, this state space is not encoded internally in the IM, and so it is virtually open-ended from the model’s perspective. More importantly, while we tested the model above on the A3DS dataset for ease of evaluation, IM itself accepts in principle any state description. Finally, the neural modules, insofar as they encode a specific task, can be reused across cognitive models, which could over time accumulate into a toolbox of well-tested modules for cognitive modeling.

In conclusion, we have provided a neuro-symbolic model of pragmatic language generation, showing that, despite limitations, integrating modern LLMs into the toolbox of cognitive scientists might offer a new research agenda on open-ending cognitive models.

References

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the*

- 2021 acm conference on fairness, accountability, and transparency (p. 610–623). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3442188.3445922> doi: 10.1145/3442188.3445922
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hassel, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 1877–1901). Curran Associates, Inc.
- Burgess, C., & Kim, H. (2018). *3d shapes dataset*.
- Chambers, C. (2017). *The seven deadly sins of psychology*. Princeton University Press.
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... others (2022). Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Clark, H. H. (1996). *Using language*. Cambridge university press.
- Creswell, A., Shanahan, M., & Higgins, I. (2022). Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*.
- Dale, R., & Reiter, E. (1995). Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive science*, 19(2), 233–263.
- Degen, J., Hawkins, R. D., Graf, C., Kreiss, E., & Goodman, N. D. (2020). When redundancy is useful: A bayesian approach to “overinformative” referring expressions. *Psychological Review*, 127(4), 591.
- Ferreira, V. S. (2019). A mechanistic framework for explaining audience design in language production. *Annual Review of Psychology*, 70(1), 29–51.
- Frank, M. C. (2023). *Large language models as models of human cognition*.
- Frank, M. C., & Goodman, N. D. (2012). Predicting pragmatic reasoning in language games. *Science*, 336(6084), 998–998.
- Franke, M., & Degen, J. (2016). Reasoning in reference games: Individual-vs. population-level probabilistic modeling. *PloS one*, 11(5), e0154854.
- Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., ... Neubig, G. (2022). Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*.
- Gatt, A., & Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61, 65–170.
- Golland, D., Liang, P., & Klein, D. (2010). A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 conference on Empirical Methods in Natural Language Processing* (pp. 410–419).
- Graf, C., Degen, J., Hawkins, R. X., & Goodman, N. D. (2016). Animal, dog, or dalmatian? level of abstraction in nominal referring expressions. In *Cogsci*.
- Grice, H. P. (1975). Logic and conversation. In *Speech acts* (pp. 41–58). Brill.
- Hawkins, R. D., Frank, M. C., & Goodman, N. D. (2020). Characterizing the dynamics of learning in repeated reference games. *Cognitive science*, 44(6), e12845.
- He-Yueya, J., Poesia, G., Wang, R. E., & Goodman, N. D. (2023). Solving math word problems by combining language models with symbolic solvers. *arXiv preprint arXiv:2304.09102*.
- Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., ... Fung, P. (2023, mar). Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12). Retrieved from <https://doi.org/10.1145/3571730> doi: 10.1145/3571730
- Krahmer, E., & Van Deemter, K. (2012). Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1), 173–218.
- Kramer, E., & van Deemter, K. (2012). Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1), 173–218.
- Levelt, W. J. (1999). Models of word production. *Trends in cognitive sciences*, 3(6), 223–232.
- Levy, R., Reali, F., & Griffiths, T. (2008). Modeling the effects of memory on human online sentence processing with particle filters. *Advances in neural information processing systems*, 21.
- Lewis, D. (1969). *Convention*. Cambridge, MA.
- Liu, J., Liu, A., Lu, X., Welleck, S., West, P., Le Bras, R., ... Hajishirzi, H. (2022). Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 3154–3169). Dublin, Ireland: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2022.acl-long.225> doi: 10.18653/v1/2022.acl-long.225
- Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2023). *Lost in the middle: How language models use long contexts*.
- Lupyan, G., & Clark, A. (2015). Words and the world: Predictive coding and the language-perception-cognition interface. *Current Directions in Psychological Science*, 24(4), 279–284. Retrieved from <https://doi.org/10.1177/0963721415570732> doi: 10.1177/0963721415570732
- McRae, K., & Matsuki, K. (2009). People use their knowledge of common events to understand language, and do so as quickly as possible. *Language and linguistics compass*, 3(6), 1417–1429.
- Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.
- Paranjape, B., Lundberg, S., Singh, S., Hajishirzi, H., Zettlemoyer, L., & Ribeiro, M. T. (2023). Art: Automatic multi-step reasoning and tool-use for large language mod-

- els. *arXiv preprint arXiv:2303.09014*.
- Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology* (pp. 1–22).
- Park, J. S., Popowski, L., Cai, C., Morris, M. R., Liang, P., & Bernstein, M. S. (2022). Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th annual acm symposium on user interface software and technology* (pp. 1–18).
- Poesia, G., Gandhi, K., Zelikman, E., & Goodman, N. D. (2023). *Certified deductive reasoning with language models*.
- Rohde, H., Futrell, R., & Lucas, C. G. (2021). What's new? a comprehension bias in favor of informativity. *Cognition*, 209, 104491.
- Shi, F., Chen, X., Misra, K., Scales, N., Dohan, D., Chi, E., ... Zhou, D. (2023). *Large language models can be easily distracted by irrelevant context*.
- Sumers, T., Yao, S., Narasimhan, K., & Griffiths, T. L. (2023). Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., ... others (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Tsvilodub, P., & Franke, M. (2023). Evaluating pragmatic abilities of image captioners on A3DS. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 1277–1285). Toronto, Canada: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2023.acl-short.110>
doi: 10.18653/v1/2023.acl-short.110
- van Rooij, I., Guest, O., Adolfi, F. G., de Haan, R., Kolokolova, A., & Rich, P. (2023). Reclaiming ai as a theoretical tool for cognitive science.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., & Zhou, D. (2022). Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.
- Wong, L., Grand, G., Lew, A. K., Goodman, N. D., Mansinghka, V. K., Andreas, J., & Tenenbaum, J. B. (2023). *From word models to world models: Translating from natural language to the probabilistic language of thought*.
- Wray, R. E., III, Kirk, J. R., & Laird, J. E. (2021). *Language models as a knowledge source for cognitive agents*.
- Xu, F., & Tenenbaum, J. B. (2007). Word learning as bayesian inference. *Psychological review*, 114(2), 245.
- Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.
- Zhang, J., Lehman, J., Stanley, K., & Clune, J. (2023). Omni: Open-endedness via models of human notions of interestingness. *arXiv preprint arXiv:2306.01711*.
- Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., ... Du, M. (2023). Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*.