

# Project ideas

CSP

Last compiled: June 5, 2024

## 1 LMs for prior elicitation for probabilistic cognitive models

1. probabilistic cognitive models often rely on human data to estimate (statistical) world knowledge
2. running prior elicitation experiments is costly, and there are many degrees of freedom in eliciting responses and in interpreting raw data (how to scale / interpret slider ratings etc.)
3. this project could investigate if we can use LLMs, properly prompted, to elicit prior judgements instead of human participants
  - try out different personalities in system prompt
  - check if judgements agree with human judgements
  - check if model predictions are worse or better than with human priors

## 2 Relevance judgements and LLMs

1. probe the human-likeness of LLM's relevance judgements w/ and w/o prompting
2. compare to human data

## 3 How false beliefs spread through language

### 3.1 Argumentative RSA agents

- run a multi-agent simulation, similar to prior literature [1], on how false beliefs spread and amplify in a social network
  - agents observe data from experiments
  - in previous work: agents showed their data to other agents
  - here: agents describe their findings in language
  - use RSA model with argumentative speakers
  - listeners can be naive or argumentation-aware
  - check impact on false belief propagation

### 3.2 Argumentative RSA agents

- as before but with griceChain agents instead of RSA

## 4 LLMs and causal reasoning

### 4.1 Squeeze causal intuitions out of LLMs

- use LLMs to “construct” intuitive causal models of everyday events
- test whether there are biases (e.g., towards specific causal structures)

## 5 LLM-chain related

Below are some student project ideas which would help expand and substantiate the framework.

### 5.1 Testing i/o

Consists of testing griceChain on other datasets or evaluating the performance in general. Requires no or little change to the current griceChain codebase.

1. Applying the griceChain implementation of the contrastive reference game to more pre-existing datasets.
  - To extend the reference game or grounded language use setting to more naturalistic datasets, Google search images or MS COCO images could be used.
  - Integrating the image captioning endpoint for the utterance proposal based on actual images can be part of the project or be implemented by us.
2. Collecting/constructing a dataset for semantic parsing
  - Currently, the semantic parsing tests contain both examples from SuperGLUE and GLUE benchmarks as well as hand-created examples representative of the modeled communicative tasks. However, for the former, sometimes the annotated truth values need to be changed compared to the original benchmark depending on the prompt / question phrasing. For the latter, more examples could be created. Students could create larger versions of such parsing datasets.
3. Applying the pipeline with other LLM backbones to check for possible advantages for models weaker than GPT-X
  - LLaMA, OpenAssistant, GPT4All...
  - involves benchmarking their instruction-following / few-shot learning abilities to GPT
4. Accuracy and variance estimation of the pipeline on a given task or for given sub-components
  - for a module like the semantic parser, there is no available estimate of the variance of results across different calls of the API, as well as across different sampling temperature sampling or tested phenomena. Having this information based on rigorous testing would be very useful for estimating the performance of the overall system.
  - this requires sizable datasets of different phenomena as well as resources for running the evaluations.
  - one concern might be how generalizable / useful these results are across models, especially if we want to offer the pipeline as a cross-model solution and need to guarantee stable performance.

### 5.2 Extending the model to more tasks

Require more substantial code extensions/revisions, but is based mostly on previous literature. E.g., implement some other RSA model with SIFD.

1. Extend griceChain to non-contrastive reference games.
2. Integrating new (e.g., more natural) communicative tasks beyond reference. Extending the pipeline to more complex or natural tasks could allow us to show the advantages of the controlled pipeline more clearly. This would require a bigger refactoring/extension of the current codebase.
  - One idea that we discussed concerned the use of different utility functions.
    - One natural extension here is to add argumentative strength to the utility function; the reason that this is natural is that argstrength in a Bayes factor/likratio implementation grounds out in a combo of semantic parser and state proposal. To implement this we need to add a state-proposer that proposes new states compatible with the utterance. For more look at the paper on argstrength RSA.
    - Another idea is to use relevance (TODO)

- Another idea is to generate possible speaker goals/preferences based on the utterance. For instance, answering wh-questions (or questions in general) might require reasoning about the speaker’s background knowledge (e.g., sensible information or goals the person might want to pursue). [PT: This is essentially a transfer / extension of the QA model to SIFD] Furthermore, the iterative pipeline might allow to model QA based on conversation history which updates goals etc rather than single-shot QA.
- One other idea could be to zoom in on the process of reasoning about vague expressions like gradable adjectives. For instance, given descriptions of utterance context, the pipeline might trade off between contextual information and general world knowledge for inferring aspects like the comparison class. E.g. a griceChain implementation of Qing & Franke (2014).

### 5.3 Improving current griceChain implementation for discriminative reference games

This involves fairly substantial revisions to the code & possibly even conceptual work. Student would change griceChain in some way to improve the performance on current task. Possibly extend SIFD paradigm.

1. Clarification questions: cf. self-critical and revising agents
  - The pragmatic module could be augmented by reasoning about which action next to picking one of the available utterances to select. These alternative actions could be, e.g., asking clarification questions. This could be guided by reasoning about the uncertainty of the current information. [PT: uncertainty in LLMs is quite a big topic by itself; modeling the reasoning is also a non-trivial task, which could be solved most naively by recursively applying the SIFD pipeline to compare utilities of possible outcomes of different actions]
2. “Loopy agent” going back to the step of generating possible utterance proposals whenever current set of options isn’t useful
  - Involves the issue of putting a cap on the number of iterations the agent could loop for.
  - Otherwise, a relatively straightforward recurrence over our current approach.

### 5.4 Real world application

1. Finding some real world use case for griceChain, exploiting its advantages over bare LLMs in terms of explainability and transparency.

## 6 Probing LLMs

- Probing LLMs seems like one of the straightforward methods for investigating the representations they might be building up.
- Following up on the discussion of whether LLMs do develop some form of understanding or reasoning and how to find that out, it would be great to get into the common methods hands-on.
- [PT: concrete applications tbd.]

## References

- [1] Kevin J.S. Zollman. “Network Epistemology: Communication in Epistemic Communities”. In: *Philosophy Compass* 8.1 (2013), pp. 15–27.