Generative Semantic Transformation Process

A Case Study in Goal Prediction via Online Bayesian Language Inference

Lorenss Martinsons*, John Muchovej*, Ilker YIldirim

* denotes equal contribution

Yale Psychology

Project Page

Cognitive & Neural Computation Lab

Yale University

Question & Proposal

Question: humans have robust language comprehension – which is believed to be mediated by parallel [predictive] production. **Could a computational model formalizing this comprehension through predictive production (1) be developed and (2) lead to more robust language comprehension in a goal-prediction environment?**

Proposal: embed a language production system (Large Language Models – LLMs) within a causal generative framework of the world to guide language prediction and comprehension.

TL;DR (Summary)

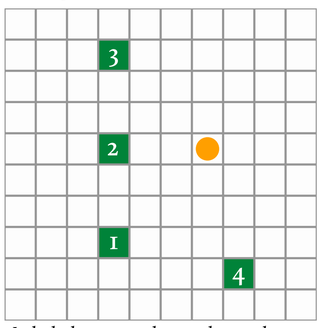
Human language comprehension is robust to noisy utterances. How this robustness occurs remains an open question in the cognitive sciences, neuroscience, and artifiicial intelligence. *We hypothesize that language comprehension is robust because it embeds, in an online fashion, a generative model of language production.*

To do this, we embed a language model (GPT-2) in a causal generative model of how world states arise and are then projected onto utterances in natural language. **We find that our model outperforms an unmodified GPT-3.5 and GPT-4 (tested in Jan 2024) when judged against human performance.**

Outstanding Questions

- Can this approach succeed in more ambiguous environments?
- Does this approach generalize to other domains than goal-inference? (e.g., physics)
- Could this approach handle more sophisticated language use?
- Does this approach support context-sensitive communication?
- Could disentangling communicative streams aid in comprehension?
- How might other inference procedures improve the computational efficiency of GSTP?

Experimental Paradigm

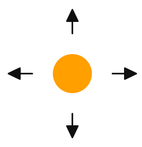


Labeled, monocolor goals are shown to participants.

Prompts are a sentence constructed **one utterance** at a time. Both *humans* and *model* must make **goal inferences** at each utterance.

Prompt 1: The agent went one right, — **Goal Inference** 4

Prompt 2: The agent went one right, drove 4 south. — **Goal Inference** 4



Agent may move in cardinal directions: N, E, S, W.

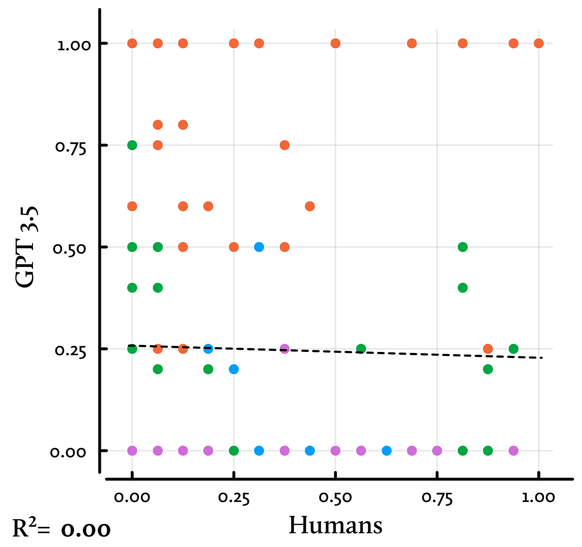
Sentence Stimuli

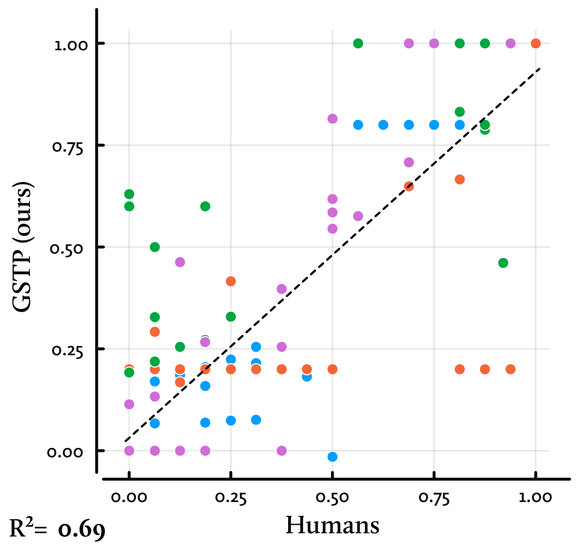
Sentence stimuli take the form of "The agent <**utterances**>"

#	Utterances (v + c + d)				Goal
1	went 3 left.				2
2	went one right,	drove 4 south.			4
3	went 1 up,	seemed three west,	drove 2 up.		3
4	moved 1 down,	went one more,	walked 3 south.		4
5	moved two wanted,	went 1 down,	submitted initial west,	seemed two down.	1
6	walked two there,	ran one up,	walked one left,	moved 2 north.	3
7	went 2 left,	walked one south,	moved one west,	went two down.	1
8	went 1 randomly,	walked 1 hoped,	walked two up,	seemed two left.	3
9	went undercover left.				2
10	moved to left,	jumped 2 up,	drove 1 west,	drove one up.	3
11	went 1 top,	moved three west,	did two south.		1
12	proceeded and sailed,	drove one east,	walked 3 south.		4
13	went opened down,	walked immediately follows,	jumped two went,	went 2 west.	1

Results (N=16)

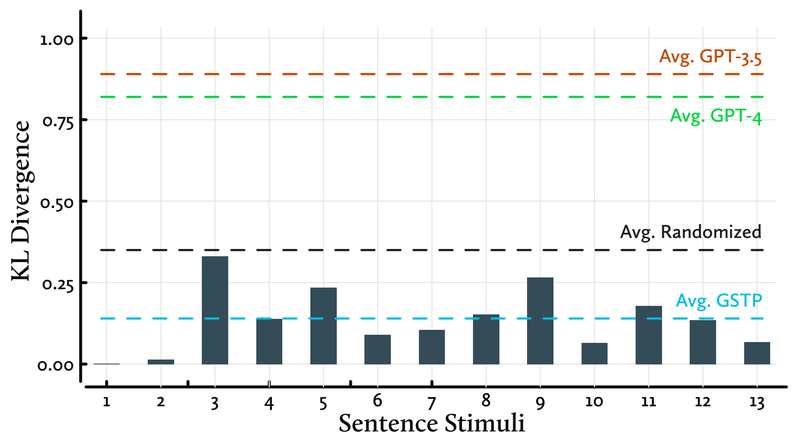
GSTP significantly predicts human judgements compared to state-of-the-art LLMs like GPT-3.5.





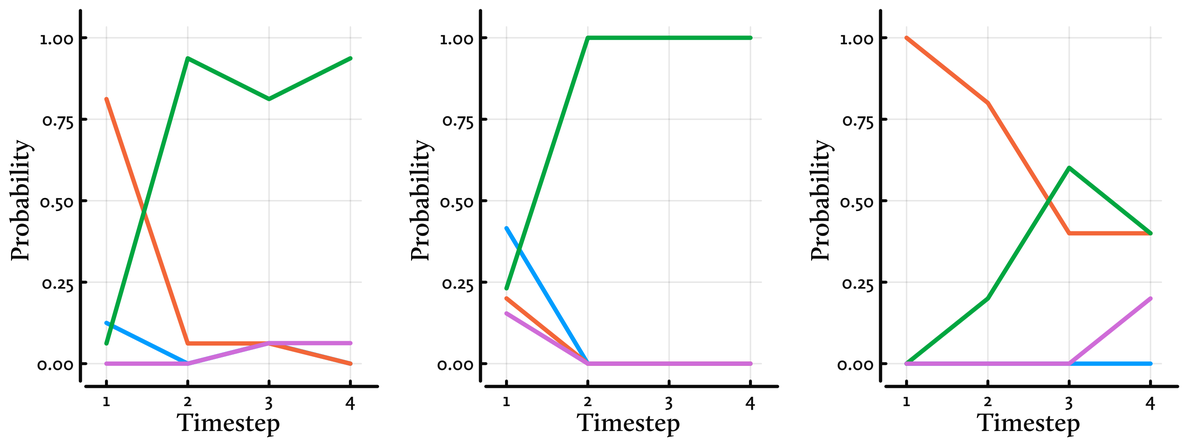
GSTP diverges from humans across many prompts, but substantially outperforms pure LLMs.

This deviation could, in part, be due to higher confidence in goal predictions by GSTP relative to humans.

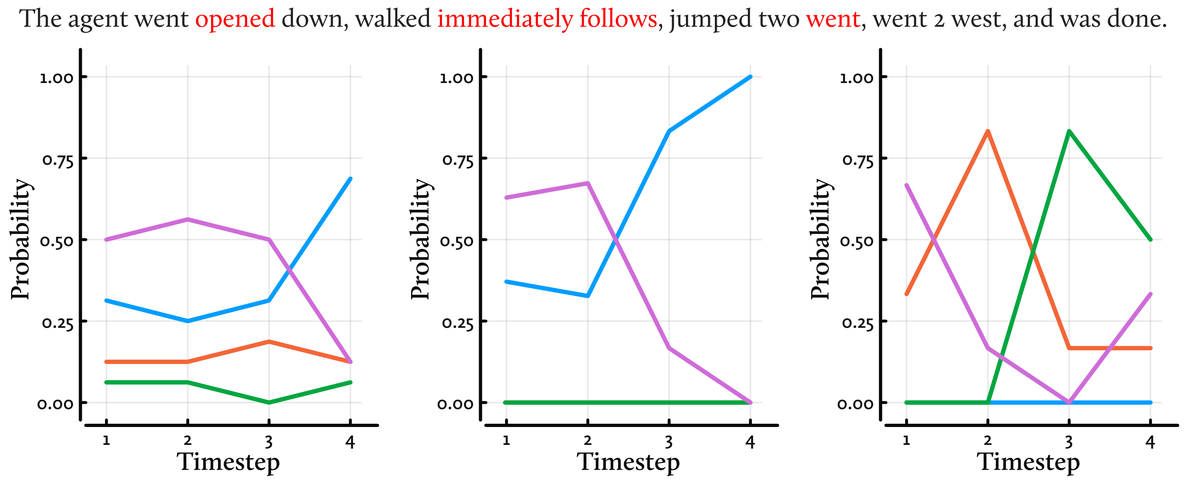


Humans **GSTP** **GPT-4**

The agent moved to left, jumped 2 up, drove 1 west, drove one up, and was done.



The agent went opened down, walked immediately follows, jumped two went, went 2 west, and was done.



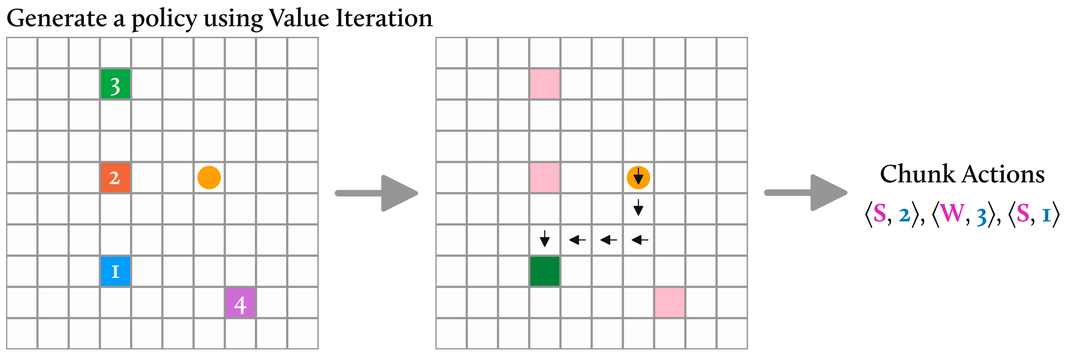
Model Architecture (GSTP)

Agent Model

Sample a target goal

$P(G = g) \sim U($ 1 , 2 , 3 , 4 $)$

Generate a policy using Value Iteration



Chunk Actions

⟨S, 2⟩, ⟨W, 3⟩, ⟨S, 1⟩

Semantic Transformation

"went 3 left"

v + c + d

movement identifier — count — direction

v ∈ **V** GPT-2's 20 nearest words for "action"

c ∈ **C** {1, 2, ..., 10} and spelled-out variants {one, two, ..., ten}

d ∈ **D** {"north", "east", ...} and {"up", "right", ...}

Language Generation

For each utterance component (v + c + d)

movement identifier | count | direction

Sample **Bern(0.8)** to produce perfectly

0.8 → **Production: Perfect**
Sample a d from D according to direction from current action chunk.

north	east	south	west	up	right	down	left

0.2 → **Production: Noisy**
Use the current context to produce the next-token, masking out the relevent V, C, or D set (using an LLM: GPT-2).

did	more	steps	wanted	...
0.653	0.221	0.12	0.05	...

Concatenate production with existing sentence to proceed to the next component

→ **Utterance** *"went 3 left"*

A similar procedure is followed for movement identifier and count, but is ommitted here for brevity.