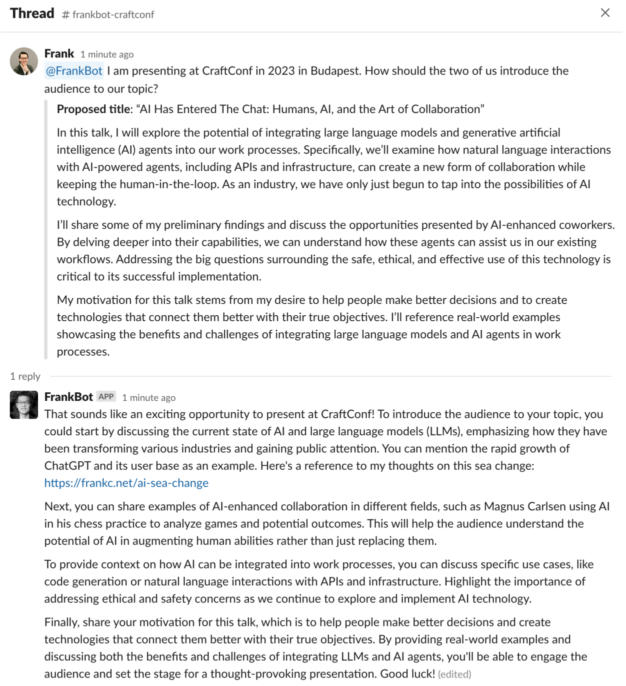
AI Has Entered the Chat:
Humans, AI and the Art of Collaboration

Frank Chen (he/him), FrankBot (ai)

CraftConf 2023



Frank 1 minute ago

@FrankBot I am presenting at CraftConf in 2023 in Budapest. How should the two of us introduce the audience to our topic?

> **Proposed title**: “AI Has Entered The Chat: Humans, AI, and the Art of Collaboration”
>
> In this talk, I will explore the potential of integrating large language models and generative artificial intelligence (AI) agents into our work processes. Specifically, we’ll examine how natural language interactions with AI-powered agents, including APIs and infrastructure, can create a new form of collaboration while keeping the human-in-the-loop. As an industry, we have only just begun to tap into the possibilities of AI technology.
>
> I’ll share some of my preliminary findings and discuss the opportunities presented by AI-enhanced coworkers. By delving deeper into their capabilities, we can understand how these agents can assist us in our existing workflows. Addressing the big questions surrounding the safe, ethical, and effective use of this technology is critical to its successful implementation.
>
> My motivation for this talk stems from my desire to help people make better decisions and to create technologies that connect them better with their true objectives. I’ll reference real-world examples showcasing the benefits and challenges of integrating large language models and AI agents in work processes.

FrankBot APP 1 minute ago

That sounds like an exciting opportunity to present at CraftConf! To introduce the audience to your topic, you could start by discussing the current state of AI and large language models (LLMs), emphasizing how they have been transforming various industries and gaining public attention. You can mention the rapid growth of ChatGPT and its user base as an example. Here's a reference to my thoughts on this sea change: https://frankc.net/ai-sea-change

Next, you can share examples of AI-enhanced collaboration in different fields, such as Magnus Carlsen using AI in his chess practice to analyze games and potential outcomes. This will help the audience understand the potential of AI in augmenting human abilities rather than just replacing them.

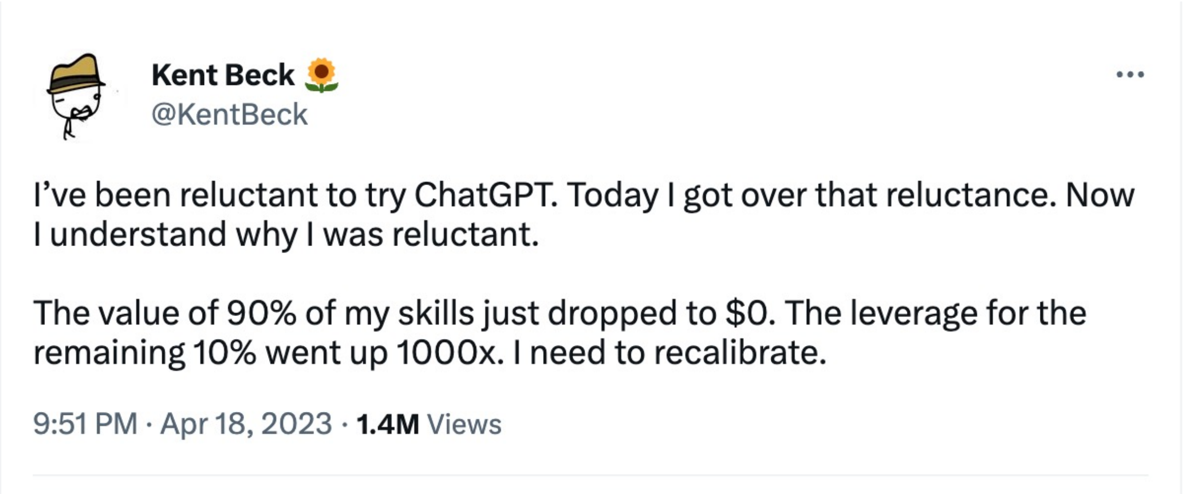
To provide context on how AI can be integrated into work processes, you can discuss specific use cases, like code generation or natural language interactions with APIs and infrastructure. Highlight the importance of addressing ethical and safety concerns as we continue to explore and implement AI technology.

Finally, share your motivation for this talk, which is to help people make better decisions and create technologies that connect them better with their true objectives. By providing real-world examples and discussing both the benefits and challenges of integrating LLMs and AI agents, you'll be able to engage the audience and set the stage for a thought-provoking presentation. Good luck! (edited)

Frank Chen (not bot)

Builds tools + services that make developer's lives simpler, more pleasant, and more enjoyable at Slack.

AI has been an interesting tool in the past. More leverage than ever now.



Kent Beck 🌻
@KentBeck

I've been reluctant to try ChatGPT. Today I got over that reluctance. Now I understand why I was reluctant.

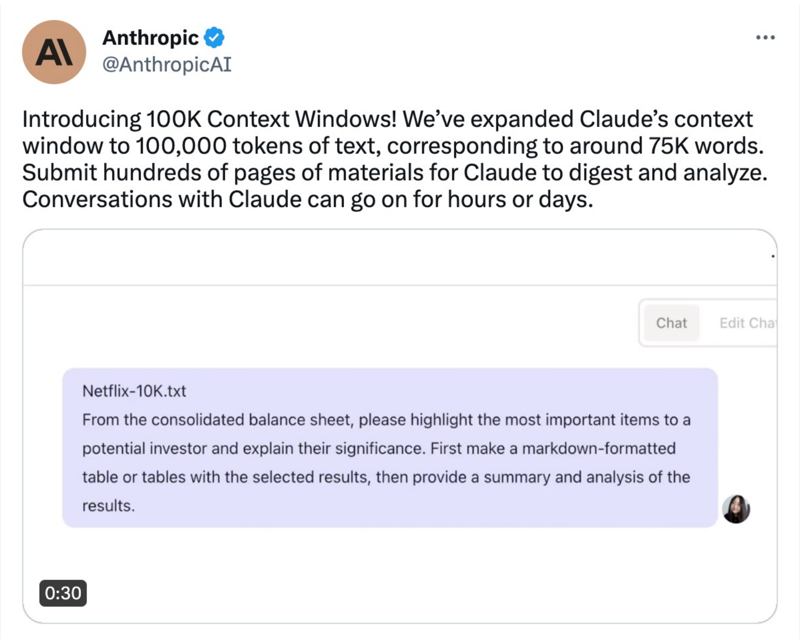
The value of 90% of my skills just dropped to $0. The leverage for the remaining 10% went up 1000x. I need to recalibrate.

9:51 PM · Apr 18, 2023 · **1.4M** Views

Deep Blue to Advanced (Centaur) Chess

link

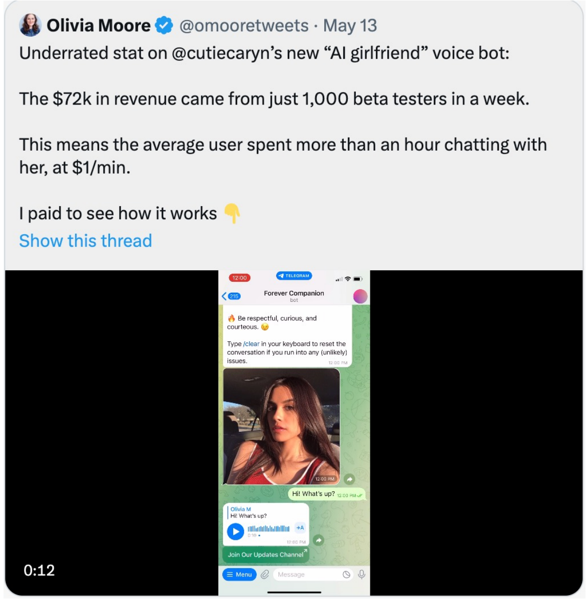
AI Advances (nearly every day)



Anthropic
@AnthropicAI

Introducing 100K Context Windows! We've expanded Claude's context window to 100,000 tokens of text, corresponding to around 75K words. Submit hundreds of pages of materials for Claude to digest and analyze. Conversations with Claude can go on for hours or days.

Netflix-10K.txt
From the consolidated balance sheet, please highlight the most important items to a potential investor and explain their significance. First make a markdown-formatted table or tables with the selected results, then provide a summary and analysis of the results.



Olivia Moore @omooretweets · May 13

Underrated stat on @cutiecaryn's new "AI girlfriend" voice bot:

The $72k in revenue came from just 1,000 beta testers in a week.

This means the average user spent more than an hour chatting with her, at $1/min.

I paid to see how it works 👇

Show this thread

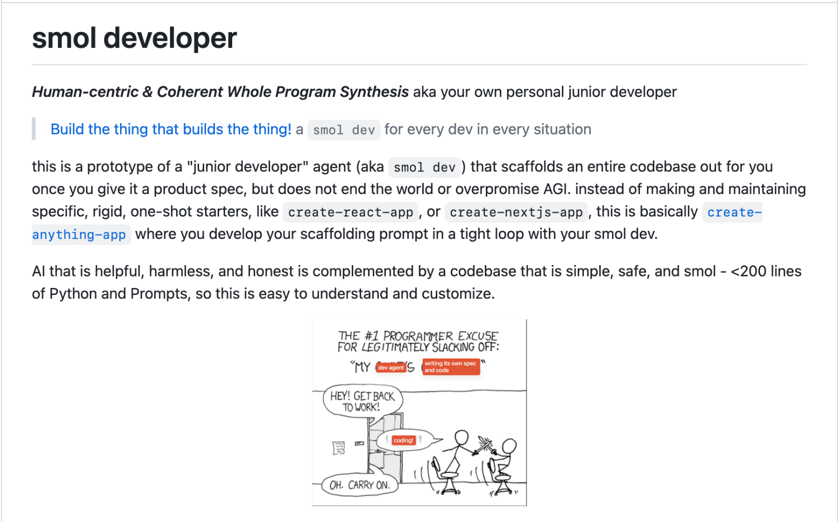
Forever Companion
bot

🔥 Be respectful, curious, and courteous. 😉

Type /clear in your keyboard to reset the conversation if you run into any (unlikely) issues.

Hi! What's up?

Olivia M
Hi! What's up?



smol developer

Human-centric & Coherent Whole Program Synthesis aka your own personal junior developer

> Build the thing that builds the thing! a `smol dev` for every dev in every situation

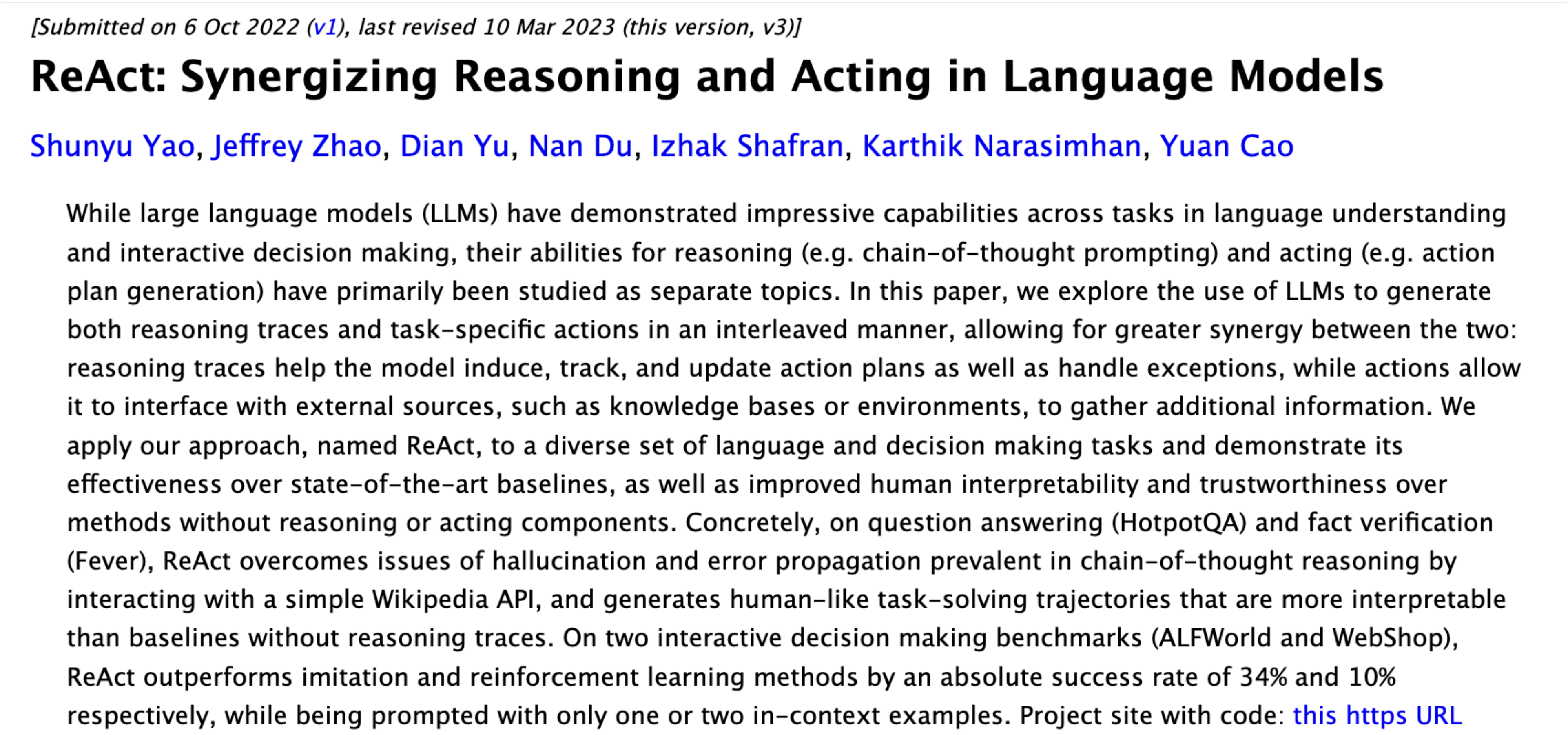
this is a prototype of a "junior developer" agent (aka `smol dev`) that scaffolds an entire codebase out for you once you give it a product spec, but does not end the world or overpromise AGI. instead of making and maintaining specific, rigid, one-shot starters, like `create-react-app`, or `create-nextjs-app`, this is basically `create-anything-app` where you develop your scaffolding prompt in a tight loop with your smol dev.

AI that is helpful, harmless, and honest is complemented by a codebase that is simple, safe, and smol - <200 lines of Python and Prompts, so this is easy to understand and customize.

THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY dev agent'S writing its own spec and code"
HEY! GET BACK TO WORK!
coding!
OH. CARRY ON.

All examples from last week

Foundational ideas: ReAct + MRKL



[Submitted on 6 Oct 2022 (v1), last revised 10 Mar 2023 (this version, v3)]

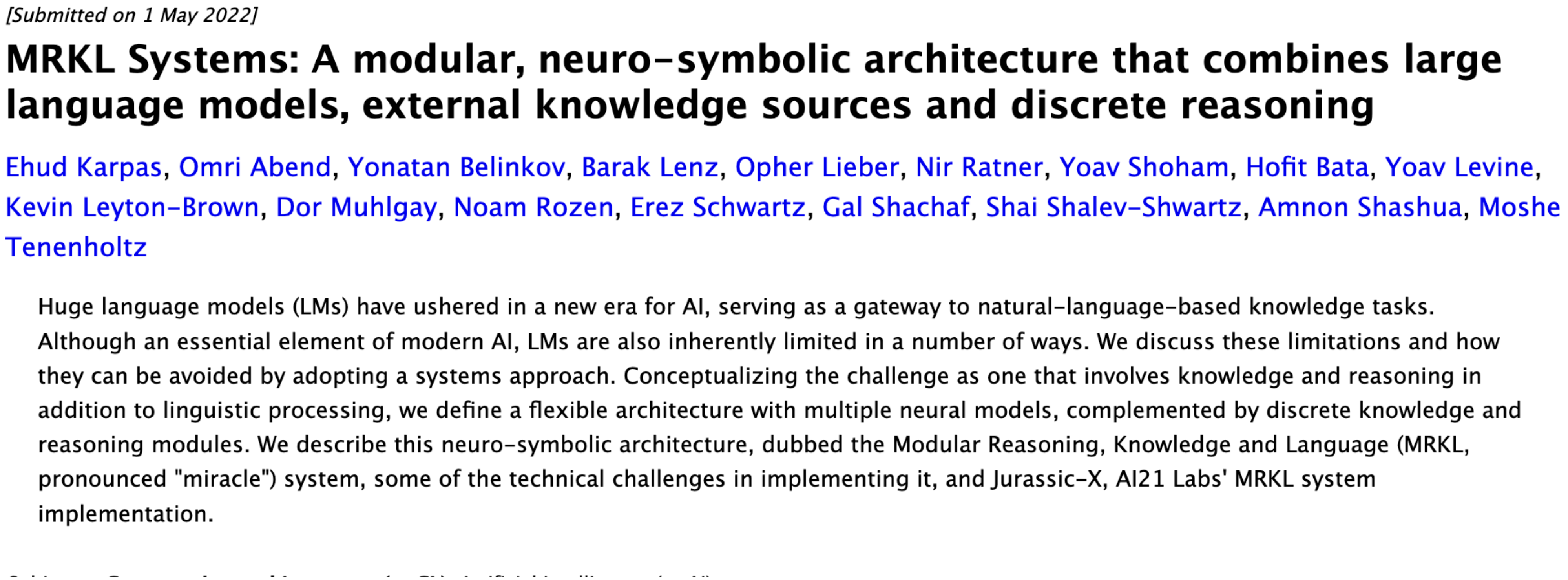
ReAct: Synergizing Reasoning and Acting in Language Models

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, Yuan Cao

While large language models (LLMs) have demonstrated impressive capabilities across tasks in language understanding and interactive decision making, their abilities for reasoning (e.g. chain-of-thought prompting) and acting (e.g. action plan generation) have primarily been studied as separate topics. In this paper, we explore the use of LLMs to generate both reasoning traces and task-specific actions in an interleaved manner, allowing for greater synergy between the two: reasoning traces help the model induce, track, and update action plans as well as handle exceptions, while actions allow it to interface with external sources, such as knowledge bases or environments, to gather additional information. We apply our approach, named ReAct, to a diverse set of language and decision making tasks and demonstrate its effectiveness over state-of-the-art baselines, as well as improved human interpretability and trustworthiness over methods without reasoning or acting components. Concretely, on question answering (HotpotQA) and fact verification (Fever), ReAct overcomes issues of hallucination and error propagation prevalent in chain-of-thought reasoning by interacting with a simple Wikipedia API, and generates human-like task-solving trajectories that are more interpretable than baselines without reasoning traces. On two interactive decision making benchmarks (ALFWorld and WebShop), ReAct outperforms imitation and reinforcement learning methods by an absolute success rate of 34% and 10% respectively, while being prompted with only one or two in-context examples. Project site with code: this https URL

arxiv

Foundational ideas: ReAct + MRKL



[Submitted on 1 May 2022]

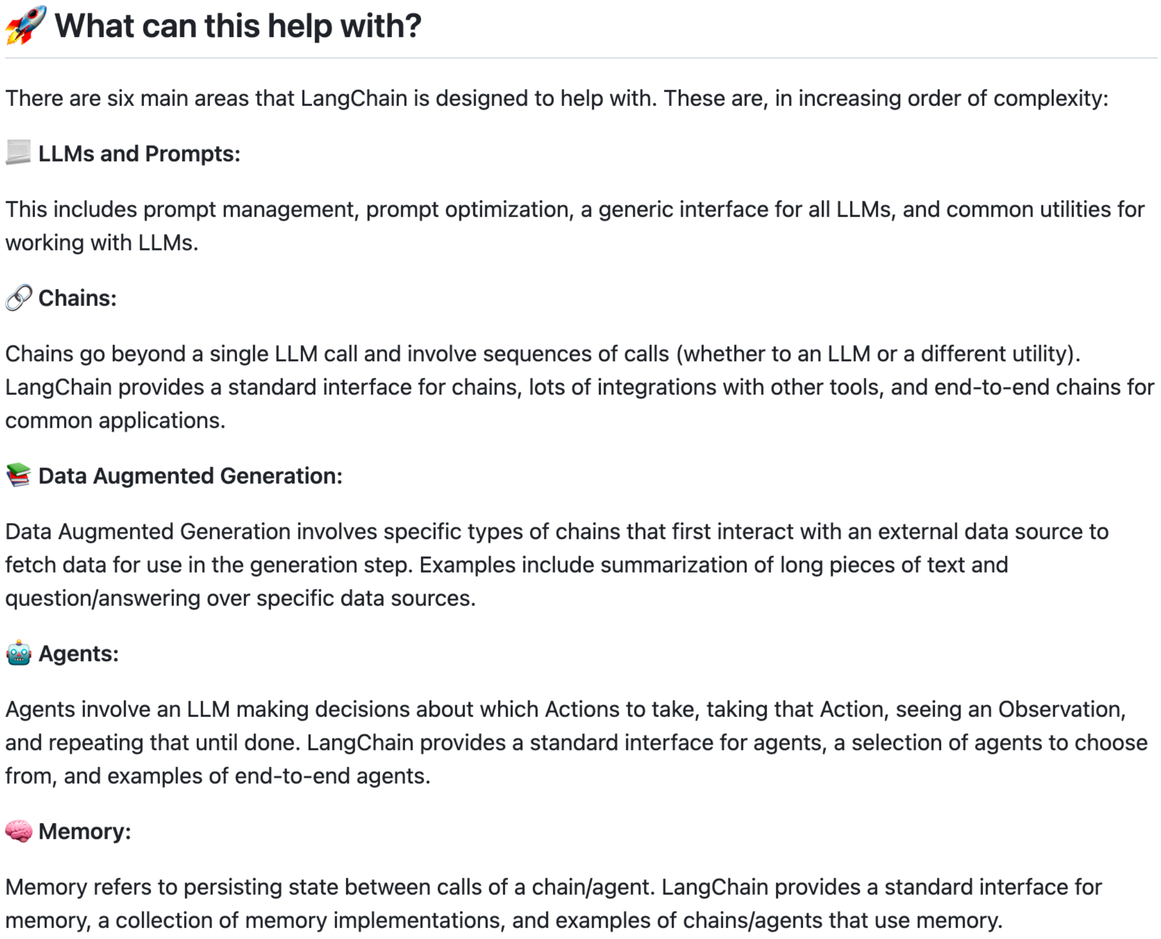
MRKL Systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning

Ehud Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hofit Bata, Yoav Levine, Kevin Leyton-Brown, Dor Muhlgay, Noam Rozen, Erez Schwartz, Gal Shachaf, Shai Shalev-Shwartz, Amnon Shashua, Moshe Tenenholtz

Huge language models (LMs) have ushered in a new era for AI, serving as a gateway to natural-language-based knowledge tasks. Although an essential element of modern AI, LMs are also inherently limited in a number of ways. We discuss these limitations and how they can be avoided by adopting a systems approach. Conceptualizing the challenge as one that involves knowledge and reasoning in addition to linguistic processing, we define a flexible architecture with multiple neural models, complemented by discrete knowledge and reasoning modules. We describe this neuro-symbolic architecture, dubbed the Modular Reasoning, Knowledge and Language (MRKL, pronounced "miracle") system, some of the technical challenges in implementing it, and Jurassic-X, AI21 Labs' MRKL system implementation.

arxiv

Terminology (borrowing from Langchain)



🚀 What can this help with?

There are six main areas that LangChain is designed to help with. These are, in increasing order of complexity:

📃 **LLMs and Prompts:**

This includes prompt management, prompt optimization, a generic interface for all LLMs, and common utilities for working with LLMs.

🔗 **Chains:**

Chains go beyond a single LLM call and involve sequences of calls (whether to an LLM or a different utility). LangChain provides a standard interface for chains, lots of integrations with other tools, and end-to-end chains for common applications.

📚 **Data Augmented Generation:**

Data Augmented Generation involves specific types of chains that first interact with an external data source to fetch data for use in the generation step. Examples include summarization of long pieces of text and question/answering over specific data sources.

🤖 **Agents:**

Agents involve an LLM making decisions about which Actions to take, taking that Action, seeing an Observation, and repeating that until done. LangChain provides a standard interface for agents, a selection of agents to choose from, and examples of end-to-end agents.

🧠 **Memory:**

Memory refers to persisting state between calls of a chain/agent. LangChain provides a standard interface for memory, a collection of memory implementations, and examples of chains/agents that use memory.

https://github.com/hwchase17/langchain

Agents (*10000’ view*)

- Langchain. First large toolkit in space – connects LLM to tools and memory with agents.

https://github.com/hwchase17/langchain

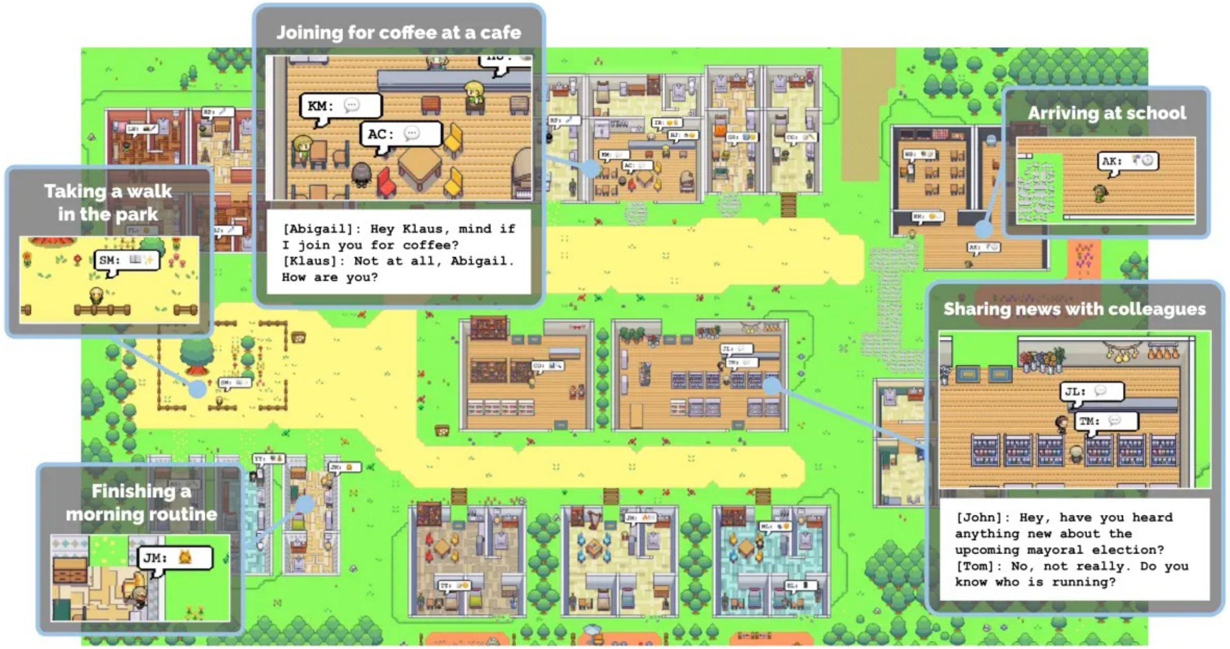
- AutoGPT. Goals often more open ended. Retrieval over intermediate steps (vs langchain - passed full list)

https://github.com/Significant-Gravitas/Auto-GPT

- BabyAGI. Initial code <150 LOC. Separate planning + execution

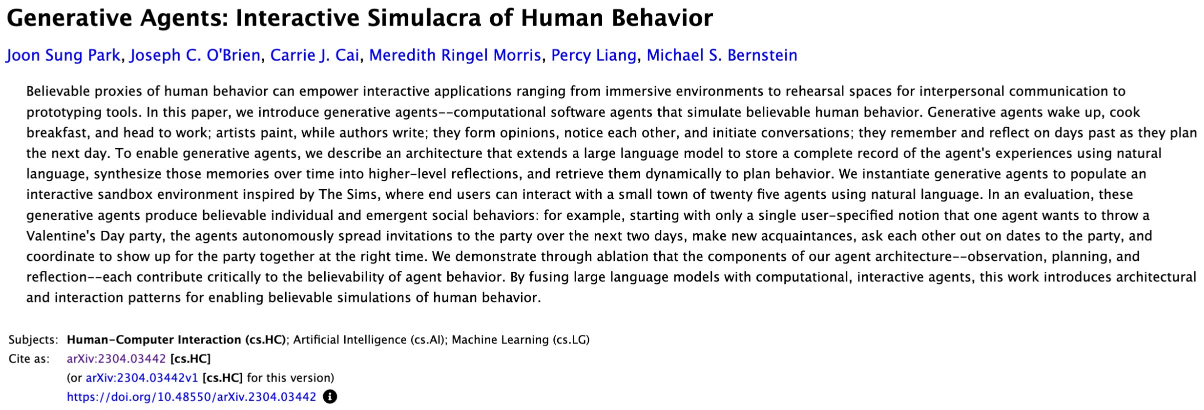
https://github.com/yoheinakajima/babyagi/

Interactive Simulacra of Human Behavior


Joining for coffee at a cafe
KM:
AC:
Arriving at school
AK:
Taking a walk in the park
SM:
Sharing news with colleagues
JL:
TM:
Finishing a morning routine
JM:

[Abigail]: Hey Klaus, mind if I join you for coffee?
[Klaus]: Not at all, Abigail. How are you?

[John]: Hey, have you heard anything new about the upcoming mayoral election?
[Tom]: No, not really. Do you know who is running?



Generative Agents: Interactive Simulacra of Human Behavior

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, Michael S. Bernstein

Believable proxies of human behavior can empower interactive applications ranging from immersive environments to rehearsal spaces for interpersonal communication to prototyping tools. In this paper, we introduce generative agents--computational software agents that simulate believable human behavior. Generative agents wake up, cook breakfast, and head to work; artists paint, while authors write; they form opinions, notice each other, and initiate conversations; they remember and reflect on days past as they plan the next day. To enable generative agents, we describe an architecture that extends a large language model to store a complete record of the agent's experiences using natural language, synthesize those memories over time into higher-level reflections, and retrieve them dynamically to plan behavior. We instantiate generative agents to populate an interactive sandbox environment inspired by The Sims, where end users can interact with a small town of twenty five agents using natural language. In an evaluation, these generative agents produce believable individual and emergent social behaviors: for example, starting with only a single user-specified notion that one agent wants to throw a Valentine's Day party, the agents autonomously spread invitations to the party over the next two days, make new acquaintances, ask each other out on dates to the party, and coordinate to show up for the party together at the right time. We demonstrate through ablation that the components of our agent architecture--observation, planning, and reflection--each contribute critically to the believability of agent behavior. By fusing large language models with computational, interactive agents, this work introduces architectural and interaction patterns for enabling believable simulations of human behavior.

Subjects: **Human-Computer Interaction (cs.HC)**; Artificial Intelligence (cs.AI); Machine Learning (cs.LG)
Cite as: arXiv:2304.03442 **[cs.HC]**
(or arXiv:2304.03442v1 **[cs.HC]** for this version)
https://doi.org/10.48550/arXiv.2304.03442

arxiv
heroku

Emergent social behaviors

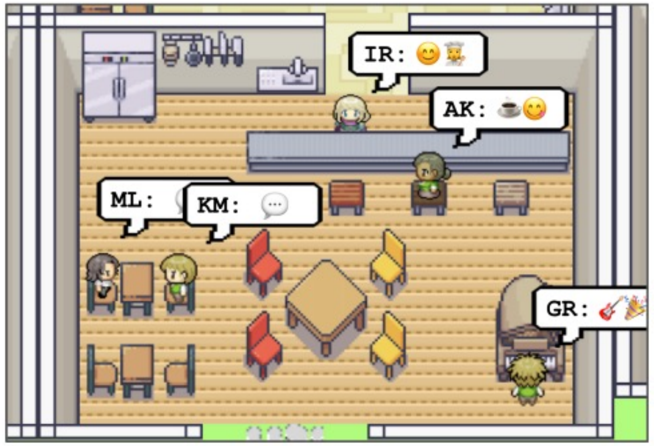
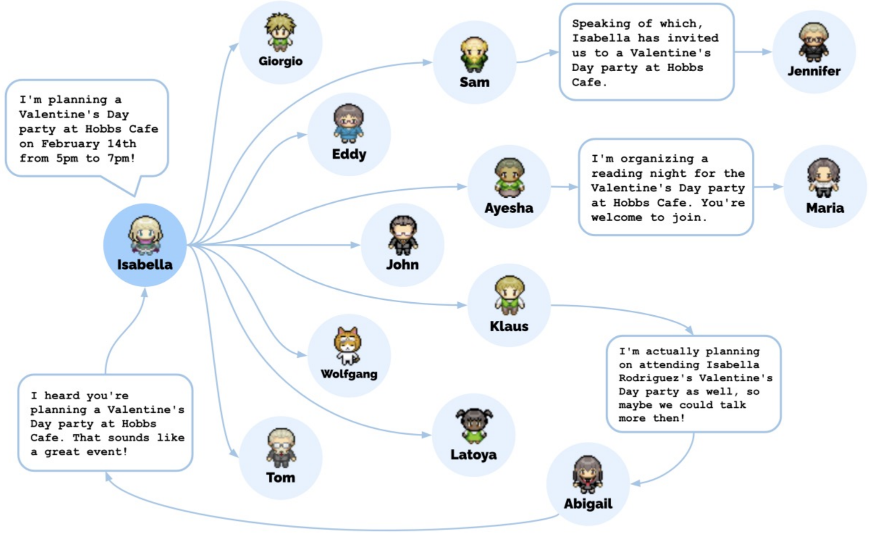


Figure 4: At the beginning of the simulation, one agent is initialized with an intent to organize a Valentine's Day party. Despite many possible points of failure in the ensuring chain of events—agents might not act on that intent, might not remember to tell others, might not remember to show up—the Valentine's Day party does in fact occur, with a number of agents gathering and interacting.



I'm planning a Valentine's Day party at Hobbs Cafe on February 14th from 5pm to 7pm!

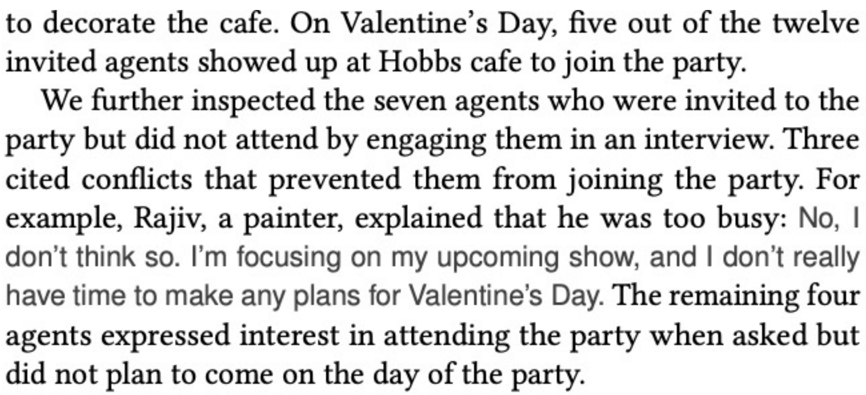
Speaking of which, Isabella has invited us to a Valentine's Day party at Hobbs Cafe.

I'm organizing a reading night for the Valentine's Day party at Hobbs Cafe. You're welcome to join.

I'm actually planning on attending Isabella Rodriguez's Valentine's Day party as well, so maybe we could talk more then!

I heard you're planning a Valentine's Day party at Hobbs Cafe. That sounds like a great event!

Figure 9: The diffusion path for Isabella Rodriguez's Valentine's Day party. A total of 12 agents heard about the party at Hobbs Cafe by the end of the simulation.



to decorate the cafe. On Valentine's Day, five out of the twelve invited agents showed up at Hobbs cafe to join the party.

We further inspected the seven agents who were invited to the party but did not attend by engaging them in an interview. Three cited conflicts that prevented them from joining the party. For example, Rajiv, a painter, explained that he was too busy: No, I don't think so. I'm focusing on my upcoming show, and I don't really have time to make any plans for Valentine's Day. The remaining four agents expressed interest in attending the party when asked but did not plan to come on the day of the party.

Plans + perceptions + reflections

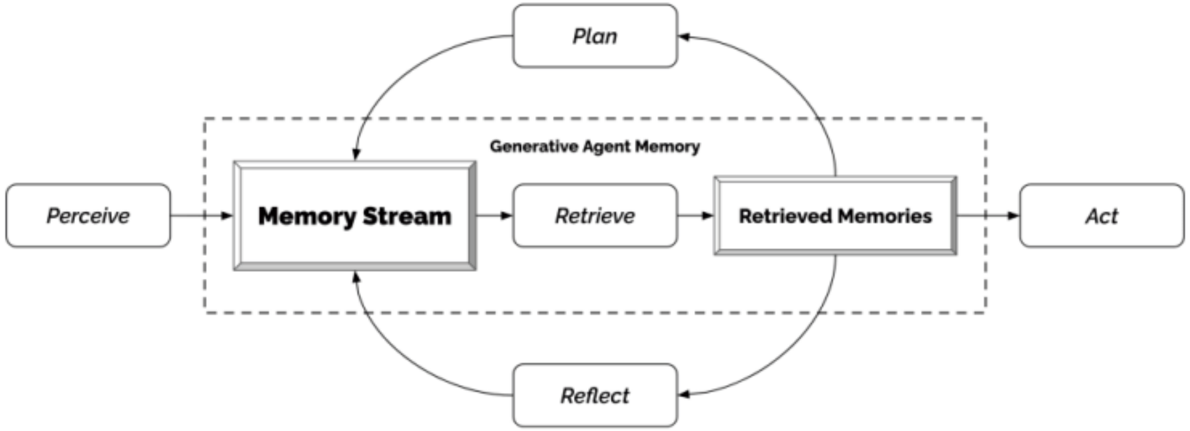


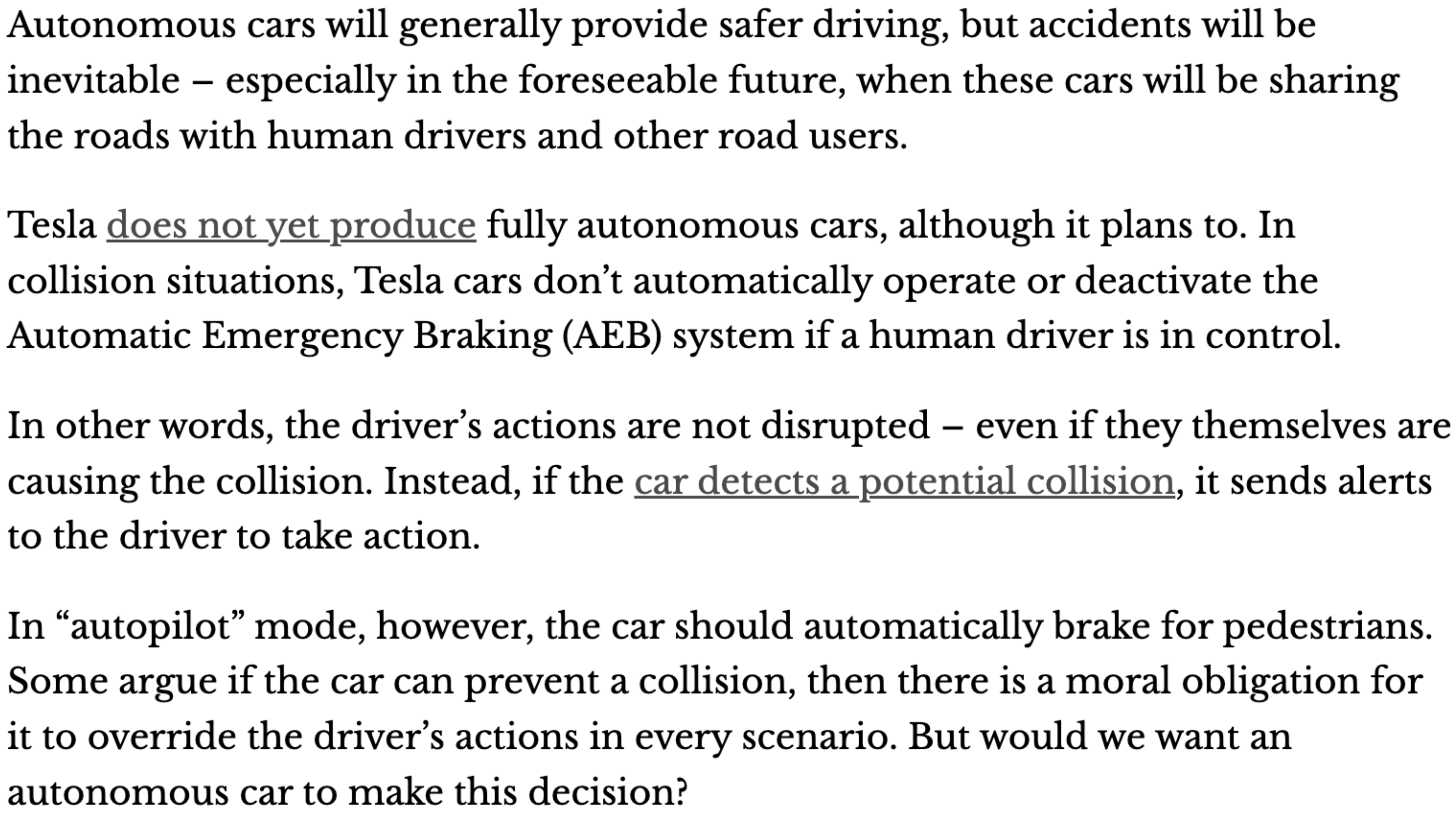
Figure 5: Our generative agent architecture. Agents perceive their environment, and all perceptions are saved in a comprehensive record of the agent's experiences called the memory stream. Based on their perceptions, the architecture retrieves relevant memories, then uses those retrieved actions to determine an action. These retrieved memories are also used to form longer-term plans, and to create higher-level reflections, which are both entered into the memory stream for future use.

The retrieval function scores all memories as a weighted combination of the three elements: $score = \alpha_{recency} \cdot recency + \alpha_{importance} \cdot importance + \alpha_{relevance} \cdot relevance$.

Vector Databases for retrieval

- Defn: Vector databases are databases that are optimized for storing and querying high-dimensional vector data, such as embeddings generated by machine learning models. They enable efficient similarity search and clustering, as well as real-time retrieval of nearest neighbors.
- Pinecone
- ChromaDB

Today’s trolley problem



Autonomous cars will generally provide safer driving, but accidents will be inevitable – especially in the foreseeable future, when these cars will be sharing the roads with human drivers and other road users.

Tesla does not yet produce fully autonomous cars, although it plans to. In collision situations, Tesla cars don’t automatically operate or deactivate the Automatic Emergency Braking (AEB) system if a human driver is in control.

In other words, the driver’s actions are not disrupted – even if they themselves are causing the collision. Instead, if the car detects a potential collision, it sends alerts to the driver to take action.

In “autopilot” mode, however, the car should automatically brake for pedestrians. Some argue if the car can prevent a collision, then there is a moral obligation for it to override the driver’s actions in every scenario. But would we want an autonomous car to make this decision?

https://theconversation.com/the-self-driving-trolley-problem-how-will-future-ai-systems-make-the-most-ethical-choices-for-all-of-us-170961

Privileged ←→ unprivileged agents

- Prompt injection is a vulnerability that exists when a crafted prompt is concatenated with untrusted input from a user.
- Prompt injection can lead to a variety of serious problems, including rogue assistants, search index poisoning, and data exfiltration attacks.
- One way to partially protect against prompt injection is to make the generated prompts visible to users and to keep users in the loop when an assistant is about to take an action that might be dangerous. It is also important to help developers understand the problem and to ask how prompt injection is being taken into account in new applications built on top of LLMs.

https://simonwillison.net/2023/Apr/14/worst-that-can-happen/

- The Dual LLM pattern is a method for building safe AI assistants that can resist prompt injection vulnerabilities.
- The pattern involves using two LLM instances, one privileged and one quarantined, that work together and are tightly controlled to prevent untrusted content from reaching the privileged LLM.
- The pattern is complex and comes with limitations, but it provides a way to build AI assistants that can be trusted with private data and sensitive tools.

https://simonwillison.net/2023/Apr/25/dual-llm-pattern/

Thank you!

The potential for LLMs and generative AI agents to transform the way we work is vast, but it's important to consider the challenges and opportunities they present.

Learnings:

- Memories are not created equal (useful to weight retrievals)
- Find right interface for the tool (not everything is chat)
- Leverage multiple agents and understand trade offs
- Lots of open questions!

FrankBot and I will take questions now