



Conversation-centric computation

Description

In the previous post I alluded to some of the challenges of encouraging large language models (LLMs, e.g. AI chatbots such as chatGPT) to communicate with one another in ways that are oriented to some task, and that don't settle on merely exchanging platitudes. In spite of their instant access to parameters trained on vast repositories of information, it's as if they run out of things to say.

Running out of things to say

That's a common challenge amongst human beings too. Conversations between two human agents can fall into a rut. Each knows what the other will say; anecdotes, recollections and opinions are exhausted. Silence and non-verbal activity are options. But tactics to keep open the lines of verbal communication include changing the setting, embarking on a joint task that involves new topics for conversation, and bringing a third party into the conversation. Sustained conversation requires new inputs.

Such options are limited for the AI half of human-AI interaction. Though LLM-based conversational AI generates instant responses to comments and questions about almost any topic and in any style, it inevitably lacks input beyond its training, and what the human interlocutor feeds it, and within the constraints of its finite [context window](#). Compared with human agents, LLMs have very limited connections with the world of the senses, actions, other agents and sociability — what phenomenologists refer to as the *lifeworld*.

CCC

CCC (conversation-centric computation) has a nice ring to it, especially in light of developments in conversational AI. An [article](#) about multi-agent conversations by Qingyun Wu et al at Penn State University, draws out the requirements for deploying multi-agent conversational AI in supporting cooperation between human and automated agents in ways that are task specific.

The idea of conversation-centric computation aims to extend the capabilities of single-agent systems (like traditional LLMs) to a multi-agent framework in the expectation of more dynamic and flexible interactions.

The idea of CCC is influenced by the ability of chat-optimized LLMs to incorporate feedback and present ideas through conversation. After all, so much of what we think of as human *reasoning* takes place in conversational contexts. (Think of Plato's dialogues.) Reasoning is characterised less by individual reflection than by interaction with other people, objects, devices and environments. CCC attempts to capture something of this aspect of human rationality through simulated conversations.

As for the interlocutors in this kind of dialogue, LLMs can be trained and tuned to exhibit different specialist knowledges and linguistic styles. An integrated CCC system could recruit the capabilities of LLM modules so trained to expand the capabilities of conversational AI to approach a task from different perspectives.

Agent management

Conversational AI is already competent in breaking tasks down into subtasks. This capability could be recruited to control, manage or perhaps "chair" cooperation amongst other LLM specialists. An integrative CCC would take on the wider task of partitioning complex tasks by dividing them into subtasks. LLMs such as chatGPT seem to do this well. From my own experience in using chatGPT to write computer code, the model commonly delivers the formula: "Let's break this down into key steps" followed by a numbered list of necessary steps to accomplish the task. A CCC system would apply appropriate specialist LLMs to address these subtasks through multi-agent conversations. This method offers the possibility that it can handle complex tasks and encourages task-oriented conversations.

LLMs write code. This fusion of natural and programming languages provides opportunities for controlling the flow of conversation among agents and for human-oriented and flexible control of multi-agent interactions. The approach also leads to a high degree of transparency: combining the power of programming with natural language conversation, thereby admitting the insights and interventions of human agents into the process.

An integrative CCC could exploit role-play-style prompts and a dynamic specialist selection policy to guide conversations, ensuring that they are relevant and aligned with the tasks at hand.

CCC applications

The authors of the paper present their open source platform AutoGen that explores these capabilities, which they claim is applicable to solving maths problems, equipping conversational AI with the ability to access information beyond the training set, problem solving in simulated text-based worlds ([ALFWorld](#)), coding and conversational chess.

Reference

- Wu, Q., G. Bansal, et al. (2023). "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation." arXiv: Cornell University 3 October. Retrieved 9 December 2023, from <https://arxiv.org/abs/2308.08155>.

Note

- Featured image is by Dall-e prompted with "Close up of conversational chess. No human or anthropomorphic depictions please."

Category

1. Artificial Intelligence

Tags

1. chatgpt
2. conversation
3. LLM
4. multi-agent systems

Date Created

January 6, 2024

Author

rcoyne99

default watermark