TechXplore

# AI can now attend a meeting and write code for you. Here's why you should be cautious

January 3 2024, by Simon Thorne

Credit: Unsplash/CC0 Public Domain

Microsoft recently launched a new version of all of its software with the addition of an artificial intelligence (AI) assistant that can do a variety of tasks for you. Copilot can summarize verbal conversations on Teams online meetings, present arguments for or against a particular point based on verbal discussions and answer a portion of your emails. It can even write computer code.

This quickly developing technology appears to take us even closer to a future where AI makes our lives easier and takes away all of the boring and repetitive things we have to do as humans.

But while these advancements are all very impressive and useful, we must be cautious in our use of such large language models (LLMs). Despite their intuitive nature, they still require skill to use them effectively, reliably and safely.

## Large language models

LLMs, a type of "deep learning" neural network, are designed to understand the user's intent by analyzing the probability of different responses based on the prompt provided. So, when a person inputs a prompt, the LLM examines the text and determines the most likely response.

ChatGPT, a prominent example of an LLM, can provide answers to prompts on a wide range of subjects. However, despite its seemingly knowledgeable responses, ChatGPT does not possess actual knowledge. Its responses are simply the most probable outcomes based on the given prompt.

When people provide ChatGPT, Copilot and other LLMs with detailed descriptions of the tasks they want to accomplish, these models can excel at providing high-quality responses. This could include generating text, images or computer code.

But, as humans, we often push the boundaries of what technology can do and what it was originally designed for. Consequently, we start using these systems to do the legwork that we should have done ourselves.

## Why over-reliance on AI could be a problem

Despite their seemingly intelligent responses, we cannot blindly trust LLMs to be accurate or reliable. We must carefully evaluate and verify their outputs, ensuring that our initial prompts are reflected in the answers provided.

To effectively verify and validate LLM outputs, we need to have a strong understanding of the subject matter. Without expertise, we cannot provide the necessary quality assurance.

This becomes particularly critical in situations where we are using LLMs to bridge gaps in our own knowledge. Here our lack of knowledge may lead us to a situation where we are simply unable to determine whether the output is correct or not. This situation can arise in generation of text and coding.

Using AI to attend meetings and summarize the discussion presents obvious risks around reliability. While the record of the meeting is based on a transcript, the meeting notes are still generated in the same fashion as other text from LLMs. They are still based on language patterns and probabilities of what was said, so they require verification before they can be acted upon.

They also suffer from interpretation problems due to homophones, words that are pronounced the same but have different meanings. People are good at understanding what is meant in such circumstances due to the context of the conversation.

But AI is not good at deducing context nor does it understand nuance.

So, expecting it to formulate arguments based upon a potentially erroneous transcript poses further problems still.

Verification is even harder if we are using AI to generate computer code . Testing computer code with test data is the only reliable method for validating its functionality. While this demonstrates that the code operates as intended, it doesn't guarantee that its behavior aligns with real-world expectations.

Suppose we use generative AI to create code for a sentiment analysis tool. The goal is to analyze product reviews and categorize sentiments as positive, neutral or negative. We can test the functionality of the system and validate the code functions correctly—that it is sound from a technical programming point of view.

However, imagine that we deploy such software in the real world and it starts to classify sarcastic product reviews as positive. The sentiment analysis system lacks the contextual knowledge necessary to understand that sarcasm is not used as positive feedback, and quite the opposite.

Verifying that a code's output matches the desired outcomes in nuanced situations such as this requires expertise.

Non programmers will have no knowledge of software engineering principles that are used to ensure code is correct, such as planning, methodology, testing and documentation. Programming is a complex discipline, and software engineering emerged as a field to manage software quality.

There is a significant risk, as my own research has shown, that non-experts will overlook or skip critical steps in the software design process, leading to code of unknown quality.

## Validation and verification

LLMs such as ChatGPT and Copilot are powerful tools that we can all benefit from. But we must be careful to not blindly trust the outputs given to us.

We are right at the start of a great revolution based on this technology. AI has infinite possibilities but it needs to be shaped, checked and verified. And at present, humans beings are the only ones who can do this.

Provided by The Conversation

Citation: AI can now attend a meeting and write code for you. Here's why you should be cautious (2024, January 3) retrieved 28 April 2024 from https://techxplore.com/news/2024-01-ai-code-cautious.html