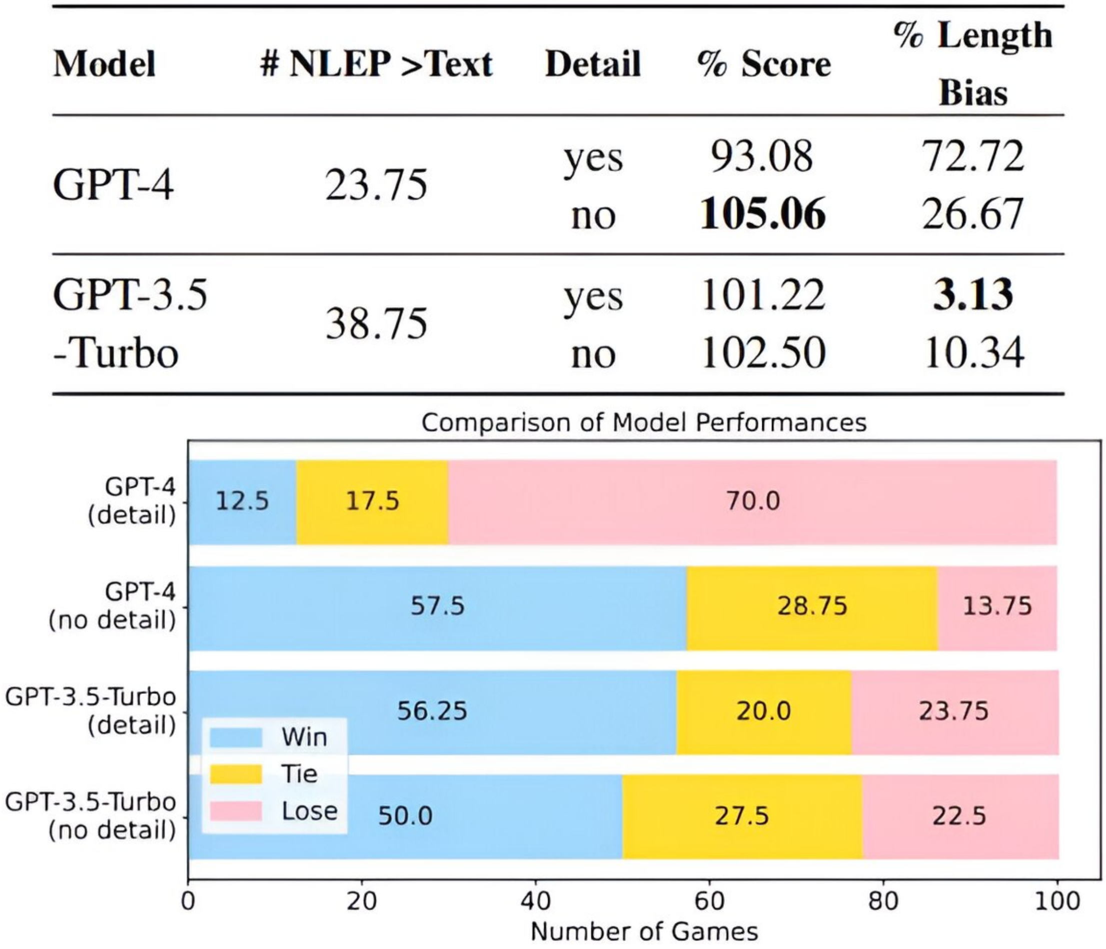
TechXplore

New technique improves the reasoning capabilities of large language models

June 14 2024, by Adam Zewe



Model	# NLEP >Text	Detail	% Score	% Length Bias
GPT-4	23.75	yes	93.08	72.72
		no	**105.06**	26.67
GPT-3.5 -Turbo	38.75	yes	101.22	**3.13**
		no	102.50	10.34

Automatic evaluations of NLEP against standard LLM-based generation with different models. Credit: *arXiv* (2023). DOI: 10.48550/arxiv.2309.10814

Large language models like those that power ChatGPT have shown impressive performance on tasks like drafting legal briefs, analyzing the sentiment of customer reviews, or translating documents into different languages.

These machine-learning models typically use only natural language to process information and answer queries, which can make it difficult for them to perform tasks that require numerical or symbolic reasoning.

For instance, a large language model might be able to memorize and recite a list of recent U.S. presidents and their birthdays, but that same model could fail if asked the question: "Which U.S. presidents elected after 1950 were born on a Wednesday?" (The answer is Jimmy Carter.)

Researchers from MIT and elsewhere have proposed a new technique that enables large language models to solve natural language, math and data analysis, and symbolic reasoning tasks by generating programs. The research is published on the *arXiv* preprint server.

Their approach, called natural language embedded programs (NLEPs), involves prompting a language model to create and execute a Python program to solve a user's query, and then output the solution as natural language.

They found that NLEPs enabled large language models to achieve higher accuracy on a wide range of reasoning tasks. The approach is also generalizable, which means one NLEP prompt can be reused for multiple tasks.

NLEPs also improve transparency, since a user could check the program to see exactly how the model reasoned about the query and fix the program if the model gave a wrong answer.

"We want AI to perform complex reasoning in a way that is transparent and trustworthy. There is still a long way to go, but we have shown that combining the capabilities of programming and natural language in large language models is a very good potential first step toward a future where people can fully understand and trust what is going on inside their AI model," says Hongyin Luo Ph.D. an MIT postdoc and co-lead author of a paper on NLEPs.

The research will be presented at the Annual Conference of the North American Chapter of the Association for Computational Linguistics.

Problem-solving with programs

Many popular large language models work by predicting the next word, or token, given some natural language input. While models like GPT-4 can be used to write programs, they embed those programs within natural language, which can lead to errors in the program reasoning or results.

With NLEPs, the MIT researchers took the opposite approach. They prompt the model to generate a step-by-step program entirely in Python code, and then embed the necessary natural language inside the program.

An NLEP is a problem-solving template with four steps. First, the model calls the necessary packages, or functions, it will need to solve the task. Step two involves importing natural language representations of the knowledge the task requires (like a list of U.S. presidents' birthdays). For step three, the model implements a function that calculates the answer. And for the final step, the model outputs the result as a line of natural language with an automatic data visualization, if needed.

"It is like a digital calculator that always gives you the correct computation result as long as the program is correct," Luo says.

The user can easily investigate the program and fix any errors in the code directly rather than needing to rerun the entire model to troubleshoot.

The approach also offers greater efficiency than some other methods. If a user has many similar questions, they can generate one core program and then replace certain variables without needing to run the model repeatedly.

To prompt the model to generate an NLEP, the researchers give it an overall instruction to write a Python program, provide two NLEP examples (one with math and one with natural language), and one test question.

"Usually, when people do this kind of few-shot prompting, they still have to design prompts for every task. We found that we can have one prompt for many tasks because it is not a prompt that teaches LLMs to solve one problem, but a prompt that teaches LLMs to solve many problems by writing a program," says Luo.

"Having language models reason with code unlocks many opportunities for tool use, output validation, more structured understanding into model's capabilities and way of thinking, and more," says Leonid Karlinsky, principal scientist at the MIT-IBM Watson AI Lab.

'No magic here'

NLEPs achieved greater than 90% accuracy when prompting GPT-4 to solve a range of symbolic reasoning tasks, like tracking shuffled objects or playing a game of 24, as well as instruction-following and text classification tasks. The researchers found that NLEPs even exhibited 30% greater accuracy than task-specific prompting methods. The method also showed improvements over open-source LLMs.

Along with boosting the accuracy of large language models, NLEPs could also improve data privacy. Since NLEP programs are run locally, sensitive user data do not need to be sent to a company like OpenAI or Google to be processed by a model.

In addition, NLEPs can enable small language models to perform better without the need to retrain a model for a certain task, which can be a costly process.

"There is no magic here. We do not have a more expensive or fancy language model. All we do is use program generation instead of natural language generation, and we can make it perform significantly better," Luo says.

However, an NLEP relies on the program generation capability of the model, so the technique does not work as well for smaller models which have been trained on limited datasets.

In the future, the researchers plan to study methods that could make smaller language models generate more effective NLEPs. In addition, they want to investigate the impact of prompt variations on NLEPs to enhance the robustness of the model's reasoning processes.

More information: Tianhua Zhang et al, Natural Language Embedded Programs for Hybrid Language Symbolic Reasoning, *arXiv* (2023). DOI: 10.48550/arxiv.2309.10814

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology

Citation: New technique improves the reasoning capabilities of large language models (2024, June 14) retrieved 21 June 2024 from https://techxplore.com/news/2024-06-technique-capabilities-large-language.html