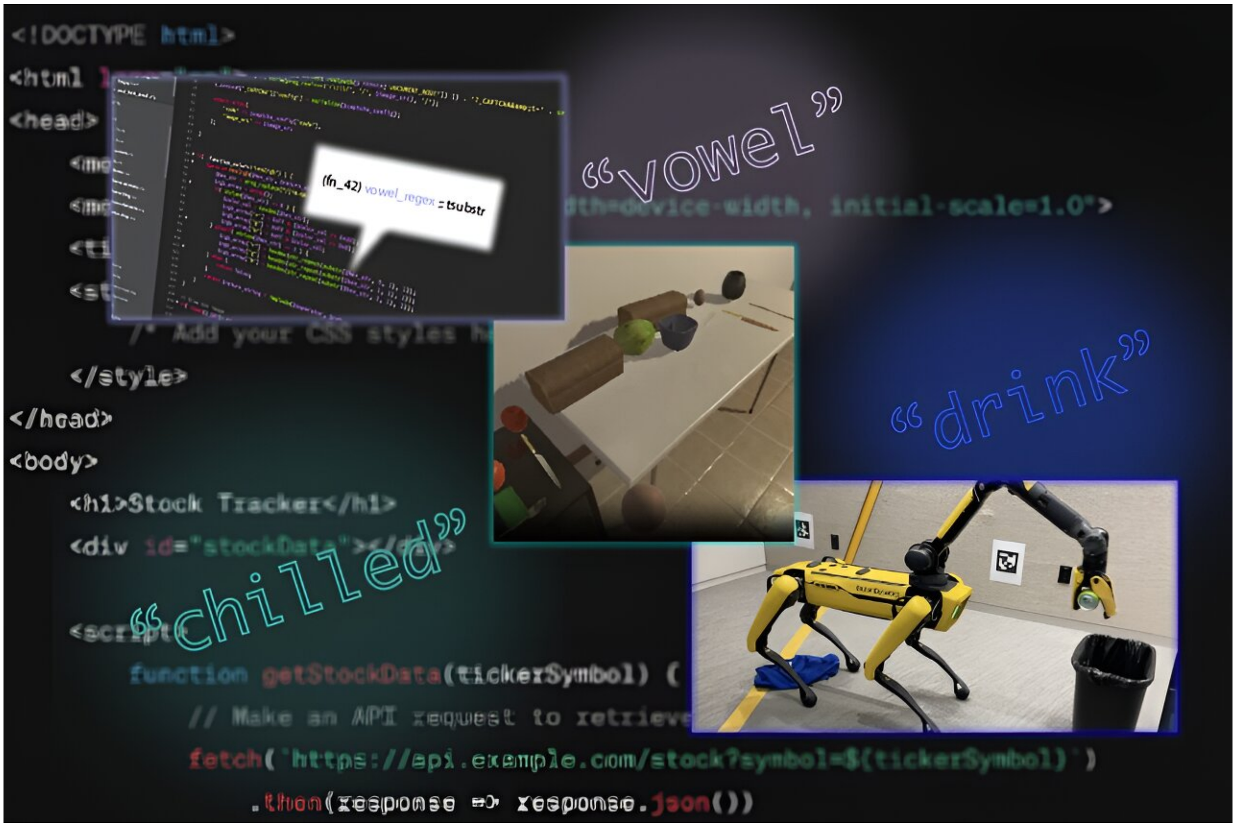
TechXplore

Natural language boosts LLM performance in coding, planning and robotics

May 1 2024, by Alex Shipps



Three new frameworks from MIT CSAIL reveal how natural language can provide important context for language models that perform coding, AI planning, and robotics tasks. Credit: Alex Shipps/MIT CSAIL, with components from the researchers and Pixabay

Large language models (LLMs) are becoming increasingly useful for programming and robotics tasks, but for more complicated reasoning problems, the gap between these systems and humans looms large. Without the ability to learn new concepts like humans do, these systems fail to form good abstractions—essentially, high-level representations of complex concepts that skip less-important details—and thus sputter when asked to do more sophisticated tasks.

Luckily, MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) researchers have found a treasure trove of abstractions within natural language. In three papers to be presented at the International Conference on Learning Representations this month, the group shows how our everyday words are a rich source of context for language models, helping them build better overarching representations for code synthesis, AI planning, and robotic navigation and manipulation. All three papers are also available on the *arXiv* preprint server.

The three separate frameworks build libraries of abstractions for their given task: LILO (library induction from language observations) can synthesize, compress, and document code; Ada (action domain acquisition) explores sequential decision-making for artificial intelligence agents; and LGA (language-guided abstraction) helps robots better understand their environments to develop more feasible plans. Each system is a neurosymbolic method, a type of AI that blends human-like neural networks and program-like logical components.

LILO: A neurosymbolic framework that codes

Large language models can be used to quickly write solutions to small-scale coding tasks, but cannot yet architect entire software libraries like the ones written by human software engineers. To take their software development capabilities further, AI models need to refactor (cut down and combine) code into libraries of succinct, readable, and reusable programs.

Refactoring tools like the previously developed MIT-led Stitch algorithm can automatically identify abstractions, so, in a nod to the Disney movie "Lilo & Stitch," CSAIL researchers combined these algorithmic refactoring approaches with LLMs. Their neurosymbolic method LILO uses a standard LLM to write code, then pairs it with Stitch to find abstractions that are comprehensively documented in a library.

LILO's unique emphasis on natural language allows the system to do tasks that require human-like common sense knowledge, such as identifying and removing all vowels from a string of code and drawing a snowflake. In both cases, the CSAIL system outperformed standalone LLMs, as well as a previous library learning algorithm from MIT called DreamCoder, indicating its ability to build a deeper understanding of the words within prompts.

These encouraging results point to how LILO could assist with things like writing programs to manipulate documents like Excel spreadsheets, helping AI answer questions about visuals, and drawing 2D graphics.

"Language models prefer to work with functions that are named in natural language," says Gabe Grand, an MIT Ph.D. student in electrical engineering and computer science, CSAIL affiliate, and lead author on the research. "Our work creates more straightforward abstractions for language models and assigns natural language names and documentation to each one, leading to more interpretable code for programmers and improved system performance."

When prompted on a programming task, LILO first uses an LLM to quickly propose solutions based on data it was trained on, and then the system slowly searches more exhaustively for outside solutions. Next, Stitch efficiently identifies common structures within the code and pulls out useful abstractions. These are then automatically named and documented by LILO, resulting in simplified programs that can be used by the system to solve more complex tasks.

The MIT framework writes programs in domain-specific programming languages, like Logo, a language developed at MIT in the 1970s to teach children about programming. Scaling up automated refactoring algorithms to handle more general programming languages like Python will be a focus for future research. Still, their work represents a step forward for how language models can facilitate increasingly elaborate coding activities.

Ada: Natural language guides AI task planning

Just like in programming, AI models that automate multi-step tasks in households and command-based video games lack abstractions. Imagine you're cooking breakfast and ask your roommate to bring a hot egg to the table—they'll intuitively abstract their background knowledge about cooking in your kitchen into a sequence of actions. In contrast, an LLM trained on similar information will still struggle to reason about what they need to build a flexible plan.

Named after the famed mathematician Ada Lovelace, who many consider the world's first programmer, the CSAIL-led "Ada" framework makes headway on this issue by developing libraries of useful plans for virtual kitchen chores and gaming. The method trains on potential tasks and their natural language descriptions, then a language model proposes action abstractions from this dataset. A human operator scores and filters the best plans into a library, so that the best possible actions can be implemented into hierarchical plans for different tasks.

"Traditionally, large language models have struggled with more complex tasks because of problems like reasoning about abstractions," says Ada lead researcher Lio Wong, an MIT graduate student in brain and cognitive sciences, CSAIL affiliate, and LILO co-author. "But we can combine the tools that software engineers and roboticists use with LLMs to solve hard problems, such as decision-making in virtual environments."

When the researchers incorporated the widely-used large language model GPT-4 into Ada, the system completed more tasks in a kitchen simulator and Mini Minecraft than the AI decision-making baseline "Code as Policies." Ada used the background information hidden within natural language to understand how to place chilled wine in a cabinet and craft a bed. The results indicated a staggering 59% and 89% task accuracy improvement, respectively.

With this success, the researchers hope to generalize their work to real-world homes, with the hopes that Ada could assist with other household tasks and aid multiple robots in a kitchen. For now, its key limitation is that it uses a generic LLM, so the CSAIL team wants to apply a more powerful, fine-tuned language model that could assist with more extensive planning. Wong and her colleagues are also considering combining Ada with a robotic manipulation framework fresh out of CSAIL: LGA (language-guided abstraction).

Language-guided abstraction: Representations for robotic tasks

Andi Peng, an MIT graduate student in electrical engineering and computer science and CSAIL affiliate, and her co-authors designed a method to help machines interpret their surroundings more like humans, cutting out unnecessary details in a complex environment like a factory or kitchen. Just like LILO and Ada, LGA has a novel focus on how natural language leads us to those better abstractions.

In these more unstructured environments, a robot will need some common sense about what it's tasked with, even with basic training beforehand. Ask a robot to hand you a bowl, for instance, and the machine will need a general understanding of which features are important within its surroundings. From there, it can reason about how to give you the item you want.

In LGA's case, humans first provide a pre-trained language model with a general task description using natural language, like "Bring me my hat." Then, the model translates this information into abstractions about the essential elements needed to perform this task. Finally, an imitation policy trained on a few demonstrations can implement these abstractions to guide a robot to grab the desired item.

Previous work required a person to take extensive notes on different manipulation tasks to pre-train a robot, which can be expensive. Remarkably, LGA guides language models to produce abstractions similar to those of a human annotator, but in less time.

To illustrate this, LGA developed robotic policies to help Boston Dynamics' Spot quadruped pick up fruits and throw drinks in a recycling bin. These experiments show how the MIT-developed method can scan the world and develop effective plans in unstructured environments, potentially guiding autonomous vehicles on the road and robots working in factories and kitchens.

"In robotics, a truth we often disregard is how much we need to refine our data to make a robot useful in the real world," says Peng. "Beyond simply memorizing what's in an image for training robots to perform tasks, we wanted to leverage computer vision and captioning models in conjunction with language. By producing text captions from what a robot sees, we show that language models can essentially build important world knowledge for a robot."

The challenge for LGA is that some behaviors can't be explained in language, making certain tasks underspecified. To expand how they represent features in an environment, Peng and her colleagues are considering incorporating multimodal visualization interfaces into their work. In the meantime, LGA provides a way for robots to gain a better feel for their surroundings when giving humans a helping hand.

An 'exciting frontier' in AI

"Library learning represents one of the most exciting frontiers in artificial intelligence, offering a path towards discovering and reasoning over compositional abstractions," says assistant professor at the University of Wisconsin-Madison Robert Hawkins, who was not involved with the papers. Hawkins notes that previous techniques exploring this subject have been "too computationally expensive to use at scale" and have an issue with the lambdas, or keywords used to describe new functions in many languages, that they generate.

"They tend to produce opaque 'lambda salads,' big piles of hard-to-interpret functions. These recent papers demonstrate a compelling way forward by placing large language models in an interactive loop with symbolic search, compression, and planning algorithms. This work enables the rapid acquisition of more interpretable and adaptive libraries for the task at hand."

By building libraries of high-quality code abstractions using natural language, the three neurosymbolic methods make it easier for language models to tackle more elaborate problems and environments in the future. This deeper understanding of the precise keywords within a prompt presents a path forward in developing more human-like AI models.

More information: Gabriel Grand et al, LILO: Learning Interpretable

Libraries by Compressing and Documenting Code, *arXiv* (2023). DOI: 10.48550/arxiv.2310.19791

Lionel Wong et al, Learning adaptive planning representations with natural language guidance, *arXiv* (2023). DOI: 10.48550/arxiv.2312.08566

Andi Peng et al, Learning with Language-Guided State Abstractions, *arXiv* (2024). DOI: 10.48550/arxiv.2402.18759

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: Natural language boosts LLM performance in coding, planning and robotics (2024, May 1) retrieved 25 June 2024 from https://techxplore.com/news/2024-05-natural-language-boosts-llm-coding.html