Robot Behavior-Tree-Based Task Generation with Large Language Models

Yue Cao, C.S. George Lee
Purdue University
AAAI-MAKE 2023 Symposium

The Upsurge of Large Language Models

IEEE Spectrum Type to search

NEWS | ARTIFICIAL INTELLIGENCE

GPT-4 Ups the Ante in the AI Arms Race ›

OpenAI's latest LLM is wildly more capable—and still sometimes a loose cannon

BY EDD GENT | 18 MAR 2023 | 5 MIN READ

AP AP NEWS

U.S. News World News Politics Sports Entertainment Business Technology Health Science Oddities Lifestyle Photography

Microsoft bakes ChatGPT-like tech into search engine Bing

By MATT O'BRIEN February 7, 2023

CNBC MARKETS BUSINESS INVESTING TECH POLITICS CNBC TV INVESTING CLUB

Mark Zuckerberg announces Meta's new large language model as A.I. race heats up

PUBLISHED FRI, FEB 24 2023·12:44 PM EST | UPDATED FRI, FEB 24 2023·1:01 PM EST

REUTERS® World Business Legal Markets More

Disrupted

3 minute read · March 21, 2023 5:30 PM EDT · Last Updated 2 days ago

Google begins opening access to its ChatGPT competitor Bard

By Jeffrey Dastin

Aa

AAAI-MAKE

PURDUE UNIVERSITY

The Upsurge of Large Language Models

- Large Language Models (LLMs):
  Neural language models with lots of parameters
  and trained on huge amount of data.
- Model timeline and parameter count (B for billion):

2018	2019	2020	2021	2022	2023
Jun: GPT 0.12 B Oct: BERT 0.34 B	Feb: GPT-2 1.5 B	May: GPT-3 175 B	May: LaMDA 137 B	Jan: InstructGPT 175 B Nov: ChatGPT unknown param[1]	Mar: GPT-4 unknown param

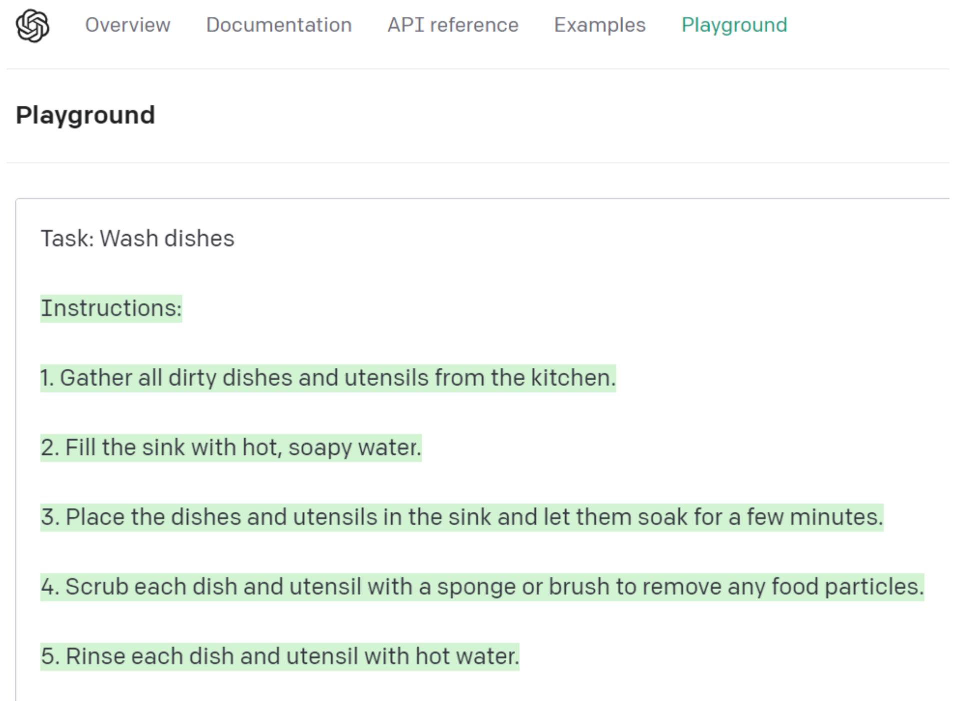
AAAI-MAKE

[1] Oren Leung, Is ChatGPT 175 Billion Parameters? Technical Analysis

PURDUE UNIVERSITY

Large Language Models Meet Robotics

GPT-3 can generate human activities.



An example tested in GPT-3 text-davinci-003 (released in Nov 2022). Text in green is generated.

How about using Large Language models in robotics?

Human activity planner

↓

Robot task planner[1]

AAAI-MAKE

[1] Wenlong, et al. "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents."

PURDUE UNIVERSITY

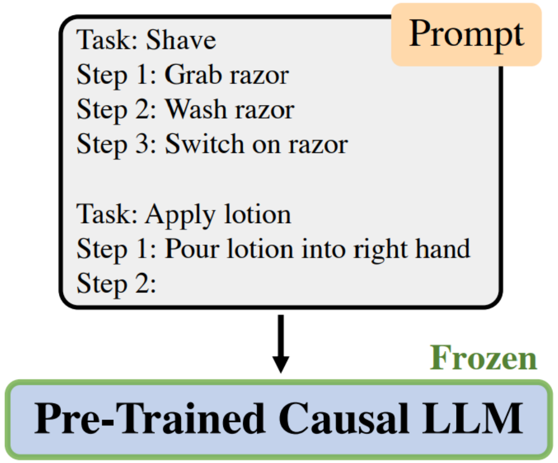
Large Language Models Meet Robotics

Previous works: Large language models as robot task planners

Work 1. LLM Planner[1]

Published in: Jan 2022

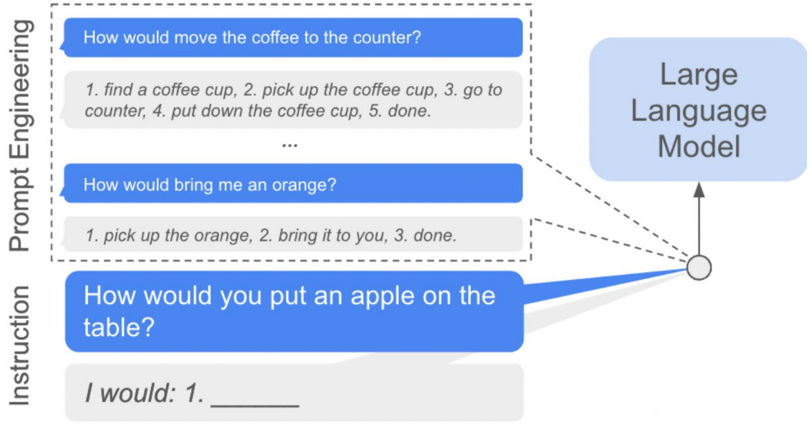
LLM: GPT-3



Work 2. SayCan[2]

Published in: Apr 2022

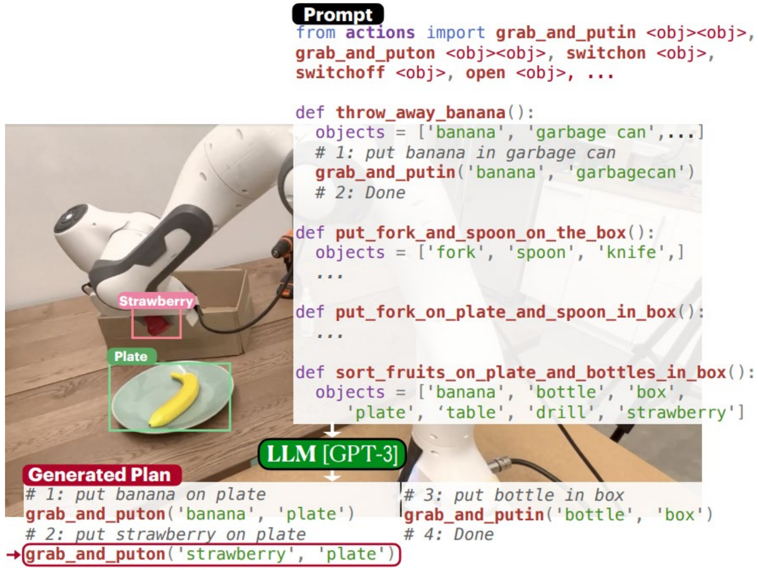
LLM: Google PaLM



Work 3. ProgPrompt[3]

Published in: Sep 2022

LLM: GPT-3



These 3 pictures are captured from the referred papers:

[1] Wenlong, et al. "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents."

[2] Ahn, Michael, et al. "Do as i can, not as i say: Grounding language in robotic affordances."

[3] Singh, Ishika, et al. "Progprompt: Generating situated robot task plans using large language models."

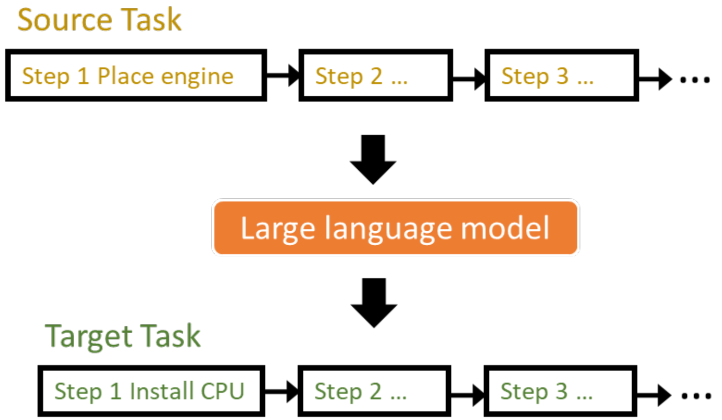
AAAI-MAKE

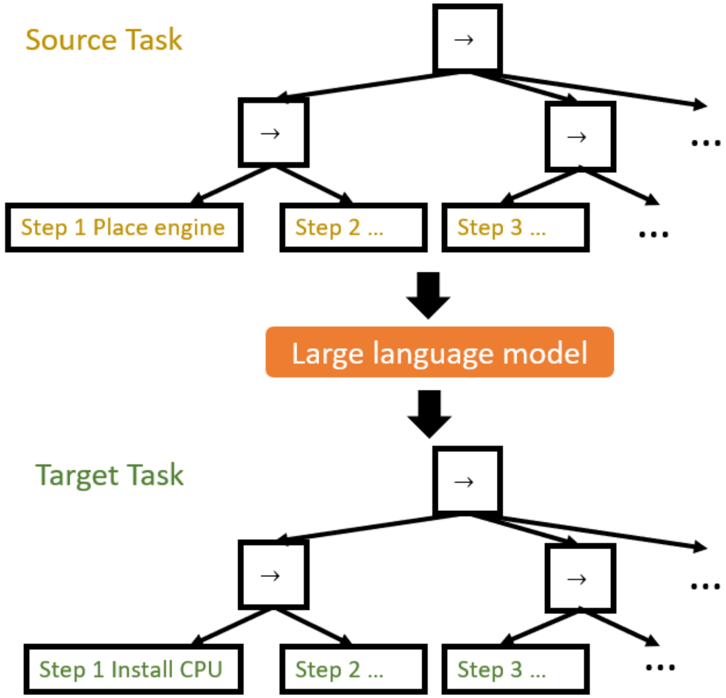
PURDUE UNIVERSITY

Large Language Models Meet Robotics

Previous works:
Generated task are sequential.

Can we generate modular tasks?
Better in terms of reusability and readability



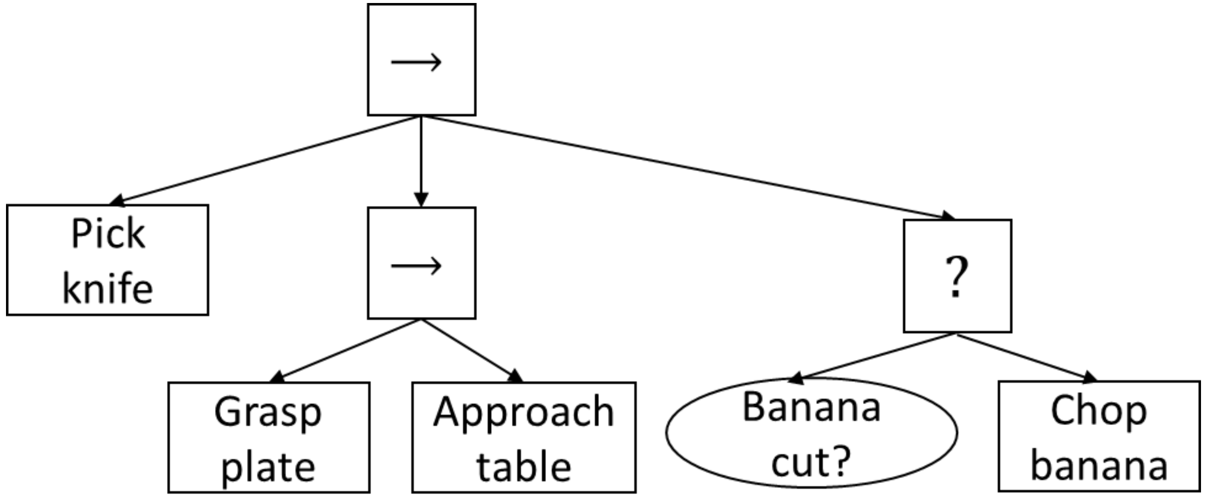


AAAI-MAKE

PURDUE UNIVERSITY

Our Idea: Generating Behavior-Tree Tasks

- A Modular task representation for robots: Behavior Tree
- Non-leaf node: control flow
  Leaf node: action (primitive task) or condition



A behavior tree example

AAAI-MAKE

PURDUE UNIVERSITY

Prompt Design in Task Generation

- In large language models, the prompt guides the text generation.
- Prompt: text that converts the original input into a template string, leaving two types of slots unfilled: input $[\quad]_X$ and output $[\quad]_Y$

Prior Work	Prompt Design
LLM Planner	Task: Task A; Step 1: [Sub-task 1] $_X$; Step 2: [Sub-task 2] $_X$; ... Task: Task B; [Step 1: Sub-task ?; Step 2: Sub-task ?; ...] $_Y$
SayCan	How would do Task A? 1. [Sub-task 1] $_X$, 2. [Sub-task 2] $_X$, ... How would do Task B? I would: [1. Sub-task ?, 2. Sub-task ?, ...] $_Y$
ProgPrompt	def Task A( ): # 1: [Sub-task 1] $_X$; # 2: [Sub-task 2] $_X$; ... def Task B( ): [# 1: Sub-task ?; # 2: Sub-task ?; ...] $_Y$

Then, how to design a prompt for behavior trees?

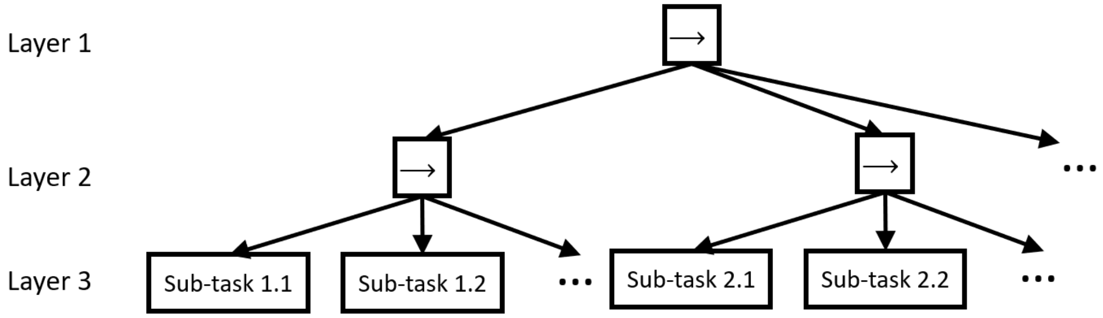
AAAI-MAKE

PURDUE UNIVERSITY

Prompt Design for Behavior Trees

- Our approach: Phase-Step Prompt

  It can generate a 3-layer behavior tree consisting of *Sequence* nodes (⟶) and *Action* nodes.



Source Task
Procedures:
Phase 1.
Step 1. [Sub-task 1.1] $_X$; Step 2. [Sub-task 1.2] $_X$; …
Phase 2.
Step 1. [Sub-task 2.1] $_X$; Step 2. [Sub-task 2.2] $_X$; …
…
Target Task: Task Description
Procedures:
[Phase 1.
Step 1. Sub-task ?; Step 2. Sub-task ?; …
Phase 2.
Step 1. Sub-task ?; Step 2. Sub-task ?; …
…] $_Y$

Layer 2 node: Phase; Layer 3 node: Step

AAAI-MAKE

PURDUE UNIVERSITY

Phase-Step Prompt Example

Source Task	Target Task: Desktop assembly
Procedures:	Procedures:
Phase 1.	Phase 1.
Step 1. Put car at a conveyor;	Step 1. Place desktop case on table;
Step 2. Lift the car.	Step 2. Insert motherboard into the case.
Phase 2.	Phase 2.
Step 1. Pick the wheel;	Step 1. Install CPU;
Step 2. Approach conveyor;	Step 2. Install RAM;
Step 3. Align wheel with wheel hub.	Step 3. Install power supply.
Phase 3.	Phase 3.
Step 1. Insert screws;	Step 1. Connect all cables and peripherals;
Step 2. Fasten screws;	Step 2. Power on the device;
Step 3. Leave the conveyor.	Step 3. Test for proper functionality.

Source task: car wheel assembly

Target task: desktop assembly

AAAI-MAKE

PURDUE UNIVERSITY

Phase-Step Prompt Example

Source Task	Target Task: Desktop assembly
Procedures:	Procedures:
Phase 1.	Phase 1.
Step 1. Put car at a conveyor;	Step 1. Place desktop case on table;
Step 2. Lift the car.	Step 2. Insert motherboard into the case.
Phase 2.	Phase 2.
Step 1. Pick the wheel;	Step 1. Install CPU;
Step 2. Approach conveyor;	Step 2. Install RAM;
Step 3. Align wheel with wheel hub.	Step 3. Install power supply.
Phase 3.	Phase 3.
Step 1. Insert screws;	Step 1. Connect all cables and peripherals;
Step 2. Fasten screws;	Step 2. Power on the device;
Step 3. Leave the conveyor.	Step 3. Test for proper functionality.

Issue: some generated sub-tasks are too abstract for robots to execute.

AAAI-MAKE

PURDUE UNIVERSITY

Decompose into Primitive Tasks

- We specify a verb list based on the robot capabilities, such as L = {pick, drop, push, pull, rotate, move, place}.
- Decide which one is a primitive task: semantic similarity test

$$\mathrm{Sim}(v, L_i) = 1 - 2\arccos\left(\frac{Enc_1(v)\cdot Enc_1(L_i)}{\|Enc_1(v)\|\|Enc_1(L_i)\|}\right)/\pi$$

For example, we apply the *Universal Sentence Encoder*[1] *as* $Enc_1(\,)$. All similarities between "install" and any verb in L is below a threshold 0.5, meaning "install …" is non-primitive.

AAAI-MAKE

[1] Cer, Daniel, et al. "Universal sentence encoder."

PURDUE UNIVERSITY

Decompose into Primitive Tasks

- Decomposition strategy 1:
  use the same source target, only change the target task description. Keep performing tree decomposition until all the sub-tasks are primitive.

  Source Task
  Procedures:
  Phase 1.
  Step 1. Same as before
  …

  ~~Target Task: Desktop assembly~~
  Target Task: Install CPU in desktop in 1 phase
  Procedures:
  To be generated …

- Decomposition strategy 2:
  add the verb requirement into prompt
  Will limit the expanded depth to 1.

  Source Task
  Procedures:
  Phase 1.
  Step 1. Same as before
  …

  ~~Target Task: Desktop assembly~~
  Target Task: Install CPU in desktop in 1 phase, only use the following verb: pick, drop, push, pull, rotate, move, place
  Procedures:
  To be generated …

Example: “Install CPU” is a generated sub-task of “Desktop assembly” but is non-primitive based on our verb list.

AAAI-MAKE

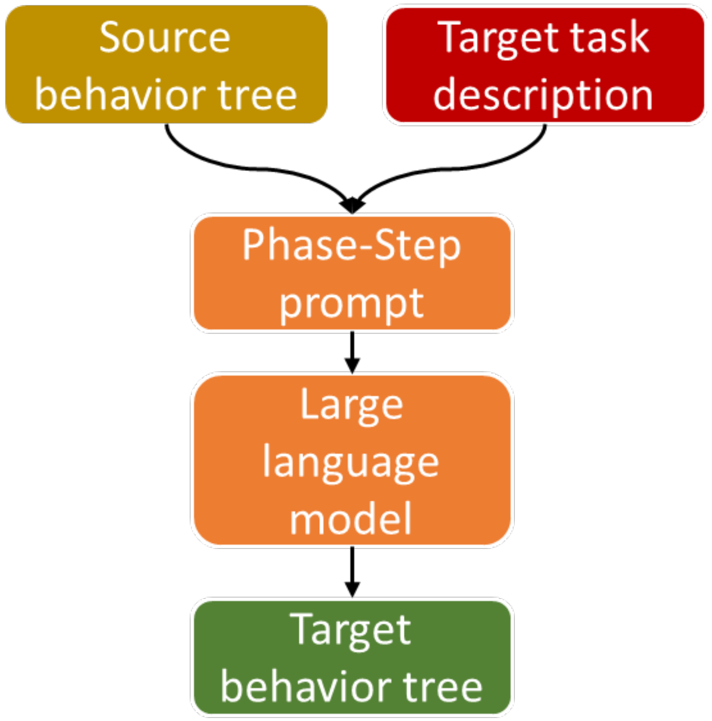
PURDUE UNIVERSITY

Automatic Source-Task Selection

Source Task
Procedures:
Phase 1.
Step 1. Put car at a conveyor;
Step 2. Lift the car.
Phase 2.
Step 1. Pick the wheel;
Step 2. Approach conveyor;
Step 3. Align wheel with wheel hub.
Phase 3.
Step 1. Insert screws;
Step 2. Fasten screws;
Step 3. Leave the conveyor.

Target Task: Desktop assembly
Procedures:
To be generated …

- The source-task behavior tree still requires an end-user to design.
- Any way to make this process automatic?



AAAI-MAKE

PURDUE UNIVERSITY

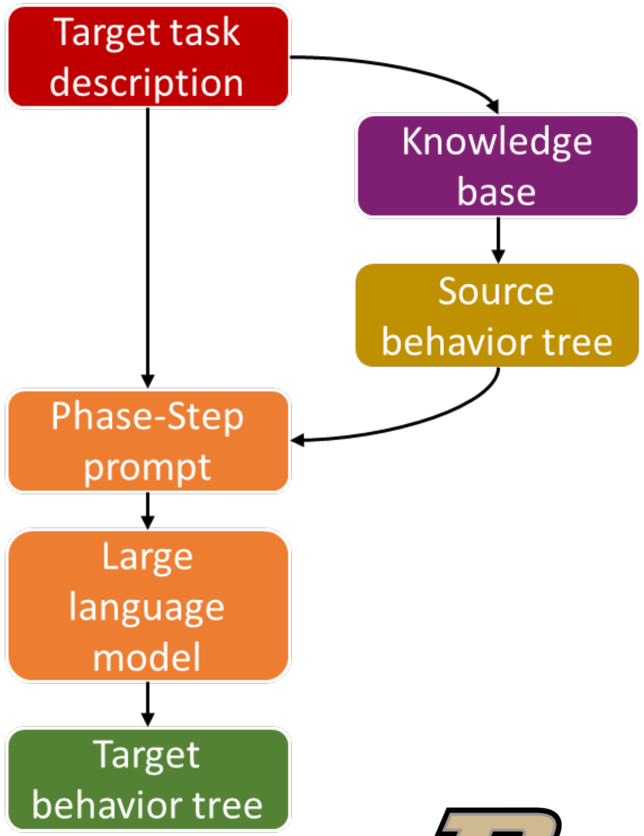
Automatic Source-Task Selection

Source Task
Procedures:
Phase 1.
Step 1. Put car at a conveyor;
Step 2. Lift the car.
Phase 2.
Step 1. Pick the wheel;
Step 2. Approach conveyor;
Step 3. Align wheel with wheel hub.
Phase 3.
Step 1. Insert screws;
Step 2. Fasten screws;
Step 3. Leave the conveyor.

Target Task: Desktop assembly
Procedures:
To be generated …

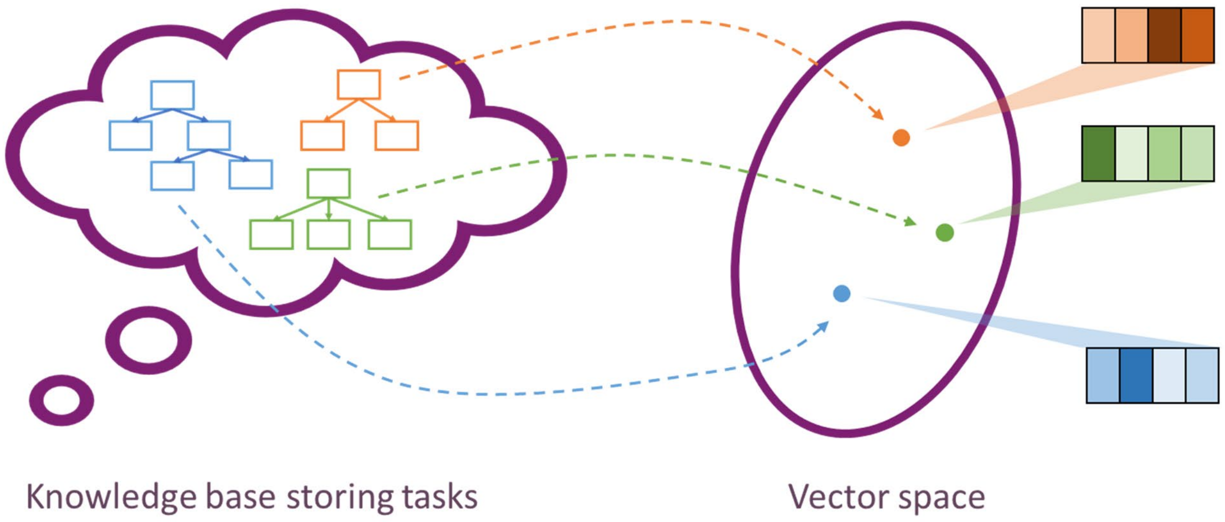
Solution:
Retrieve relevant behavior tree from a knowledge base.

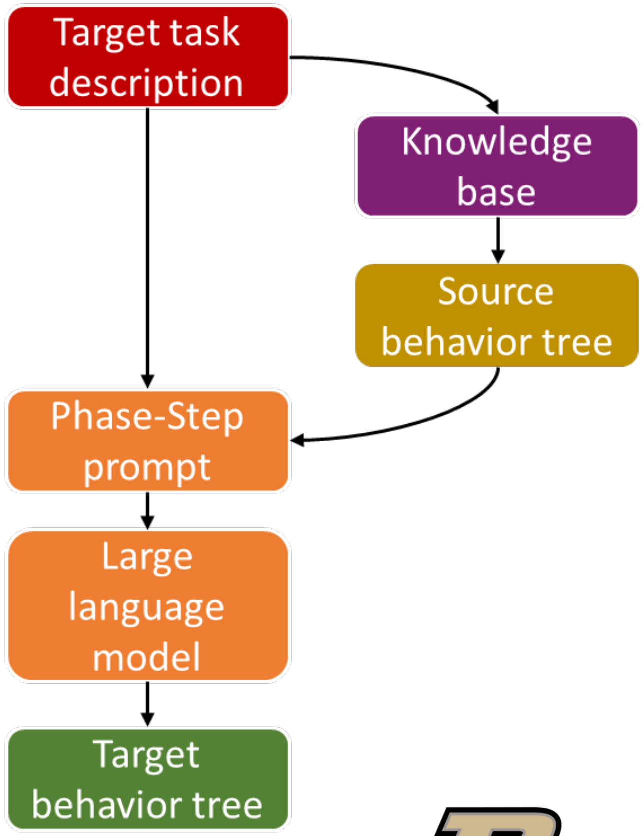
Now, the end-user just needs to input a short target task description.



Automatic Source-Task Selection

- Embedding-based[1] behavior tree retrieval. Find the most semantically similar behavior tree from the knowledge base.





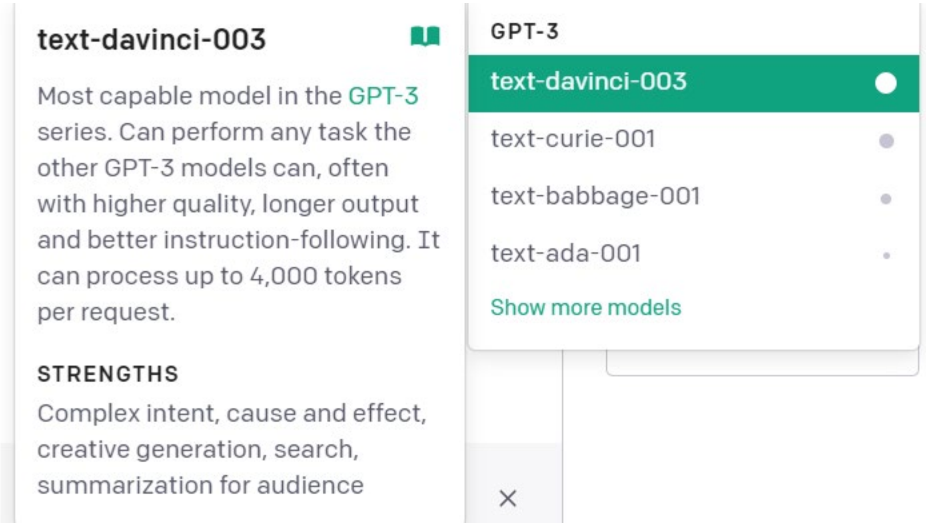
AAAI-MAKE

[1] Cao, Yue, and CS George Lee. "Behavior-Tree Embeddings for Robot Task-Level Knowledge."

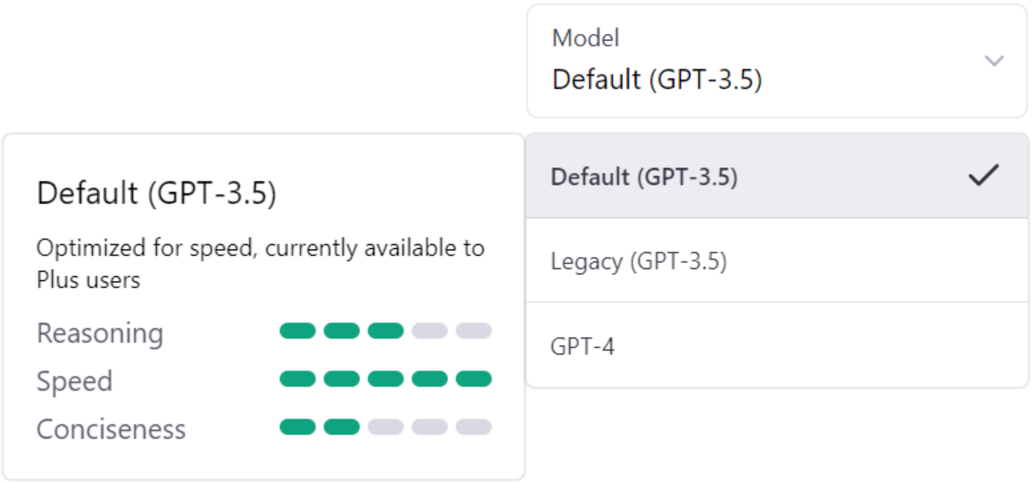
PURDUE UNIVERSITY

Evaluations: Models Used

1. GPT-3 text-davinci-003 released in Nov 2022



2. ChatGPT Feb/13/2023 version first version released in Nov 2022



ChatGPT PLUS

AAAI-MAKE

PURDUE UNIVERSITY

Evaluation 1: Ablation Study

- Can LLMs generate modular tasks without our Phase-Step prompt?
- Rarely.

	GPT-3 text-davinci-003		ChatGPT	
	Avg. R	Avg. N_{total}	Avg. R	Avg. N_{total}
PS-none prompt	0.12	5.67	0.22	6.90
PS-wheel prompt	0.65	7.80	0.60	9.77
PS-desktop prompt	0.93	8.80	0.66	9.13

Each test was conduct using 30 different target task descriptions.
PS-none prompt: No phase-step prompt, example: "Generate a desk assembly task in behavior tree"
PS-wheel prompt: Use a car-wheel-assembly behavior tree as source task in prompt
PS-desktop prompt: Use a desktop-assembly behavior tree as source task in prompt
R: a metric for tree modularity, If sequential task, $R = 0$
N_{total}: total number of generated *Action* nodes

AAAI-MAKE

PURDUE UNIVERSITY

Evaluation 2: Task Quality Assessment

- Does the Phase-Step prompt affect the quality of generated tasks?
- Yes. Prompts containing more details tend to generate more informative tasks.

	GPT-3 text-davinci-003		ChatGPT	
	Avg. N_{mate}	Avg. N_{total}	Avg. N_{mate}	Avg. N_{total}
PS-wheel prompt	1.87	7.80	3.20	10.80
PS-desktop prompt	5.33	8.87	5.73	10.00

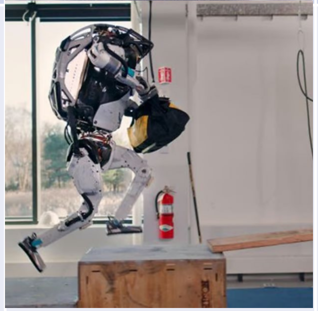
Each test was conduct using 15 different target task descriptions in robotic assembly domain.
N_{mate}: total number of part-mating operations in robotic assembly
PS-wheel prompt: Use a car-wheel-assembly behavior tree as source task in prompt, $N_{mate} = 1$
PS-desktop prompt: Use a desktop-assembly behavior tree as source task in prompt, $N_{mate} = 5$

AAAI-MAKE

PURDUE UNIVERSITY

Evaluation 3: Limitation in Uncommon Tasks

- Can LLMs generate uncommon tasks?
- Sometimes cannot.

Target task description	Picture	Feature	davinci-003	ChatGPT 3.5	ChatGPT 4.0
Vince Lombardi Trophy crafting		Superbowl trophy, American football shape	Know football shape	Fail to generate	Know football shape
Atlas robot assembly		Boston Dynamics, hydraulic actuators	Don’t know hydraulic actuators	Know hydraulic actuators	Don’t know hydraulic actuators

MAKE

PURDUE UNIVERSITY

Conclusion

Utilize large language models to generate robot behavior trees.

- Phase-Step prompt for modular task generation

- Decomposition strategy to match robot capabilities

- Automatic source-task selection from knowledge base

AAAI-MAKE

PURDUE UNIVERSITY

Take-Aways on Large Language Model Applications (outside of robotics)

- Input side:
  Integrate the information from knowledge base to augment prompt. It can provide better guide/regulation for generation.
- Output side:
  Large language models (broadly, Generative AI) have surprising power in generating new things. But in many applications, there lack metrics to evaluate the generated results.

Thank you!

AAAI-MAKE

PURDUE UNIVERSITY