Ask more, know better: Reinforce-Learned Prompt Questions for Decision Making with Large Language Models

Xue Yan[1], Yan Song[1], Xinyu Cui[1], Filippos Christianos[2], Haifeng Zhang[1], David Mguni[2] ,Jun Wang[3]

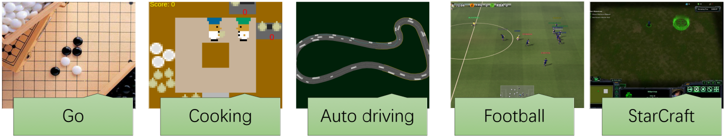
[1]Institute of Automation, CAS, [2]Noah's Ark Lab, Huawei, [3]University College London

December 6, 2023

Introduction

Background

- Our goal is to achieve **complex decision-making and reasoning** to tackle problems such as **auto-driving, autonomous cooking etc**
- Large language models (LLMs) offer a powerful tool to be able to do this since they **capture large amounts of human prior knowledge**



Bottleneck: LLMs though powerful suffer from critical drawbacks:

- Previous studies, such as Tree of Thought (ToT) [4] and Reasoning via Planning (RAP) [2], **require human-engineered prompts & action grounding functions** which are **labor-intensive and costly**.
- Decision-making approach with LLMs **do not generalise well** and are **susceptible to errors**.
- Needing vast human input detracts from goal of achieving **fully autonomous general artificial intelligence**

Our Algorithm I

- We propose an **end-to-end framework with automated prompt generation, CoT reasoning, and action policies**
- Our framework tackles **autonomous decision-making tasks**
- **Makes use of human knowledge** captured in LLMs
- Does not require extensive human input to **hand-craft prompts**

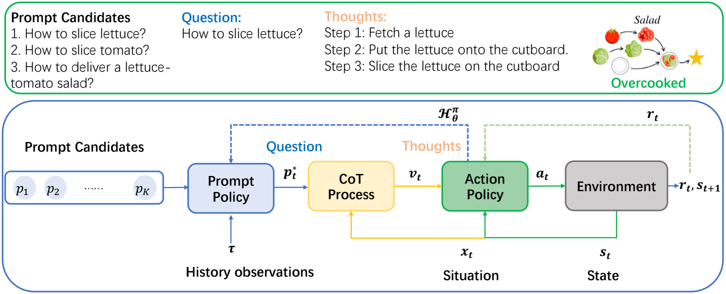


Figure: *Top:* Example of the workflow from Prompt candidates to CoT reasoning. *Bottom:* The illustration of our bilevel optimisation framework.

Our Algorithm II

An example on Overcooked

1. At time step t, the system is at an environment state s_t and receives the observation o_t.

o_t : *Item X at Position X. Agent* 1 *is at location (3,2) and currently holds nothing.*

2. There is a predefined prompt candidate set $\mathcal{P}$ containing possible questions for the environment.

$\mathcal{P}$: *How to slice lettuce? How to slice tomato? How to deliver a lettuce-tomato salad?*

3. A prompt p_t is selected from the prompt candidate set by the prompt generation policy i.e. $p_t \sim \pi_\phi(\cdot|o_t, \ldots, o_{t-j\wedge 0})$.

p_t : *How to slice lettuce?*

4. Get output of the CoT process $\upsilon_{t^+} \sim \pi^{\text{re}}(p_t)$.

CoT υ_{t^+}*: Step 1: Fetch a lettuce Step 2: Put the lettuce onto the cutboard. Step 3: Slice the lettuce on the cutboard*

Our Algorithm III

5. An action $a_{t^+} \sim \pi_\theta(\cdot | o_t, v_{t^+})$ is taken.

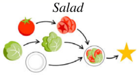
Action a_{t^+}: Go left

Bilevel-LLM Inference Process

There is a predefined **prompt candidate set** $\mathcal{P}$ containing possible questions

1. How to slice lettuce? 2. How to slice tomato?3. How to deliver a lettuce-tomato salad?

Overcooked



- At time step t, the environment provides:
  The observation o_t: *Items X at Positions X. Agent is at location (3,2). Agent currently holds nothing.*
  The situation x_t: *Agent currently holds nothing.*
- A prompt p_t is selected from the prompt candidate set via the prompt generation policy i.e. $p_t \sim \pi_\phi(\cdot | o_t, \cdots, o_{t-j\wedge 0})$.
  The selected prompt question p_t: *How to slice lettuce?*
- Get output of the CoT reasoning process $v_t \sim \pi^{\mathrm{re}}(p_t, x_t)$
  The thought v_t: *Step 1: Fetch a lettuce. Step 2: Put the lettuce onto the cutboard. Step 3: Slice the lettuce on the cutboard.*
- An action $a_t \sim \pi_\theta(\cdot | o_t, v_t)$ is taken.
  Action a_t: *Go left*

Bilevel Framework

We offer a new **leader-follower bilevel framework** capable of

- learning to ask relevant questions (prompts)
- learning to perform actions in an environment with the guidance of CoT reasoning for these questions

Bilevel Optimization: The prompt generation policy and action policy are alternately optimized with the other frozen.

- In the outer loop, the prompt generation policy is optimized by policy gradient to reduce the uncertainty of the output of the action policy.

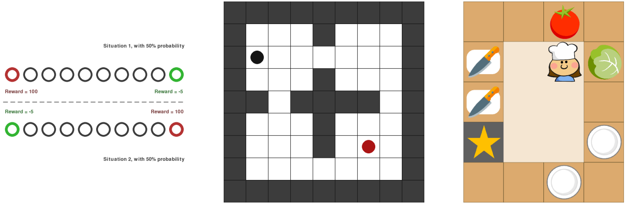
$$J(\pi_\phi) = \mathbb{E}_{\pi_\theta,\pi_\phi,\pi^{\mathrm{re}}}\left[-\sum_{t\geq 0}\gamma^t \mathcal{H}^{\pi_\theta}(y_{t^+}) | y_{t^+} = (o_t, \upsilon_{t^+}), \upsilon_{t^+} \sim \pi^{\mathrm{re}}(p_t)\right] \tag{1}$$

- In the inner loop, the action policy is expected to maximize the cumulative environment rewards.

$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta,\pi_\phi,\pi^{\mathrm{re}}}\left[\sum_{t\geq 0}\gamma_I^t r_{t^+} | p_t \sim \pi_\phi\right], \text{ with } \pi_\phi, \pi^{\mathrm{re}} \text{fixed} \tag{2}$$

Experiments I

Three classes of environments: 1. **ChainWorld** under partial and full observation. 2. **FourRoom** 3. **Overcooked** with 2 recipes.



(a) ChainWorld (b) FourRoom (c) Overcooked

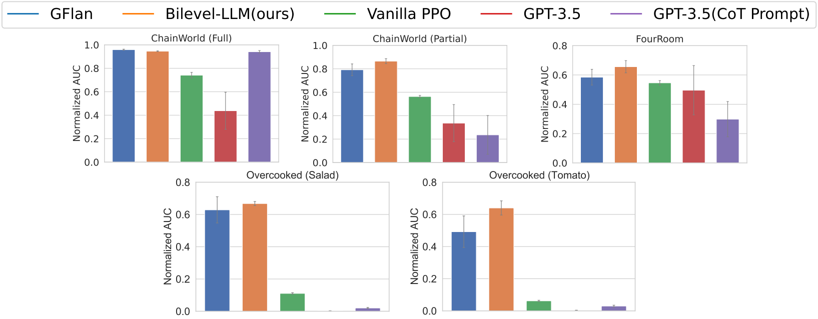
Five baselines

- **Vanilla PPO**: The symbolic embedding of observation is the input of action policy.
- **GFlan** [1]: A PPO algorithm uses the LLM Flan-T5 [3] as the action policy.

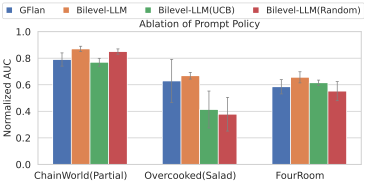
Experiments II

- **GPT3.5**: Here Task descriptions, textual context, and executable action candidates are used as input prompts.
- **GPT3.5 (CoT prompt)**: Besides those used in the GPT-3.5 setting, we further incorporate examples of human interactions with the environment or human-established task decompositions as a part of the input prompt.
- **Bilevel-LLM (ours)**: 1. The Bilevel LLM framework integrates prompt generation, CoT reasoning, and action policies. 2. The Flan-T5 LLM as the action policy 3. Train it by the PPO following the setting of GFlan.

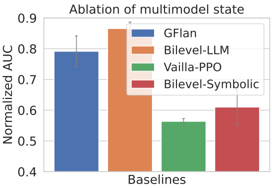
Results of comparison with baselines. Bilevel-LLM outperforms other baselines in almost all environments and also exhibits a smaller standard error than the suboptimal GFlan. **1.** This indicates the prompt question and subsequent CoT guidance are helpful for performing precise actions. **2.** GPT-3.5 and GPT-3.5 (CoT Prompt), methods without gradient update or prompt adjustment, struggle to solve long-term decision-making tasks.



Ablation Study I



(d) Ablation on prompt policy



(e) Ablation on multimodal

1. Does the prompt policy with policy gradient improve performance? Bilevel-LLM uses policy gradient to optimise the prompt generation policy. Figure (d) shows that Bilevel-LLM outperforms other versions including using UCB and random prompt policies.

2. Can the Bilevel-LLM framework accommodate multimodal state representation? As shown in Figure (e), our framework incorporating the prompt questions and CoT reasoning into the action decision is useful in situations where the action policy uses textual observation and symbolic observation as input.

Ablation Study II

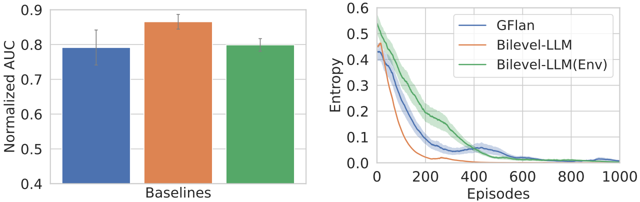


Figure: Ablation of the entropy objective on Chainworld (Partial). *Left:* Normalized AUC reward. *Right:* Entropy of the action policy.

3. Does the entropy objective improve performance? Bilevel-LLM with minus entropy of the action policy as prompt policy's objective outperforms Bilevel-LLM (Env) with environment rewards as the objective and exhibits lower entropy of the action policy.

References I

[1] Thomas Carta et al. "Grounding large language models in interactive environments with online reinforcement learning". In: *arXiv preprint arXiv:2302.02662* (2023).

[2] Shibo Hao et al. "Reasoning with Language Model is Planning with World Model". In: *arXiv preprint arXiv:2305.14992* (2023).

[3] Jack W Rae et al. "Scaling language models: Methods, analysis & insights from training gopher". In: *arXiv preprint arXiv:2112.11446* (2021).

[4] Shunyu Yao et al. "Tree of thoughts: Deliberate problem solving with large language models". In: *arXiv preprint arXiv:2305.10601* (2023).