OMNI

OPEN-ENDEDNESS VIA MODELS OF HUMAN NOTIONS OF INTERESTINGNESS

Training Reinforcement Learning (RL) Agents in Large Environments

- Large environment → Large search space
- → Infinitely many possible tasks
  - Even when we only count tasks that the agent is able to learn

How do we choose which tasks to learn first?

- Large Language Models (LLMs) contain human knowledge
  - Humans know which tasks are interesting
- → An LLM could tell an RL agent which tasks to learn first

FIGURE 1 Minecraft – an example of an extremely large environment, with an infinitely large action space. Mojang 2011, *Minecraft*. Screenshot from https://minecraft.fandom.com/wiki/Gameplay

Method

PROMPT

```
You are a player in a game. You want to learn as many skills as possible.
You can do these tasks well: <tasks done well>.
Suggest whether the given tasks are interesting: <tasks to be determined>.
```

ALGORITHM

Algorithm 1 Mechanism to partition the task set into interesting and boring sets.

1: Sort the tasks based on the evaluated task success rates.
2: Create two empty sets, one to track the interesting tasks and one to track the boring tasks.
3: Identify the task with highest success rate and not in any of the sets. Add it to the interesting set.
4: Prompt the LM to determine if any of the remaining tasks are boring, contexted on the current set of interesting tasks. Tasks in the interesting set are input as `<tasks done well>` and tasks yet to be categorized are input as `<tasks to be determined>` in the LM prompt (above).
5: Update the boring set with tasks that the LM has determined as boring.
6: Repeat steps 3 - 5 until all tasks are in either set.

FIGURE 2 Algorithm as presented in [1].

Usage in Practice

- Algorithm tested in Crafter
- RL agent trained using Proximal Policy Optimization (PPO)
  - State-of-the-art «standard» RL method
- OMNI's role: Suggest tasks for agent to perform
  - Interesting tasks will be chosen more often
  - Influences policy of RL agent (choosing an action)
- «Boring» tasks were added to show LLM's decision-making ability

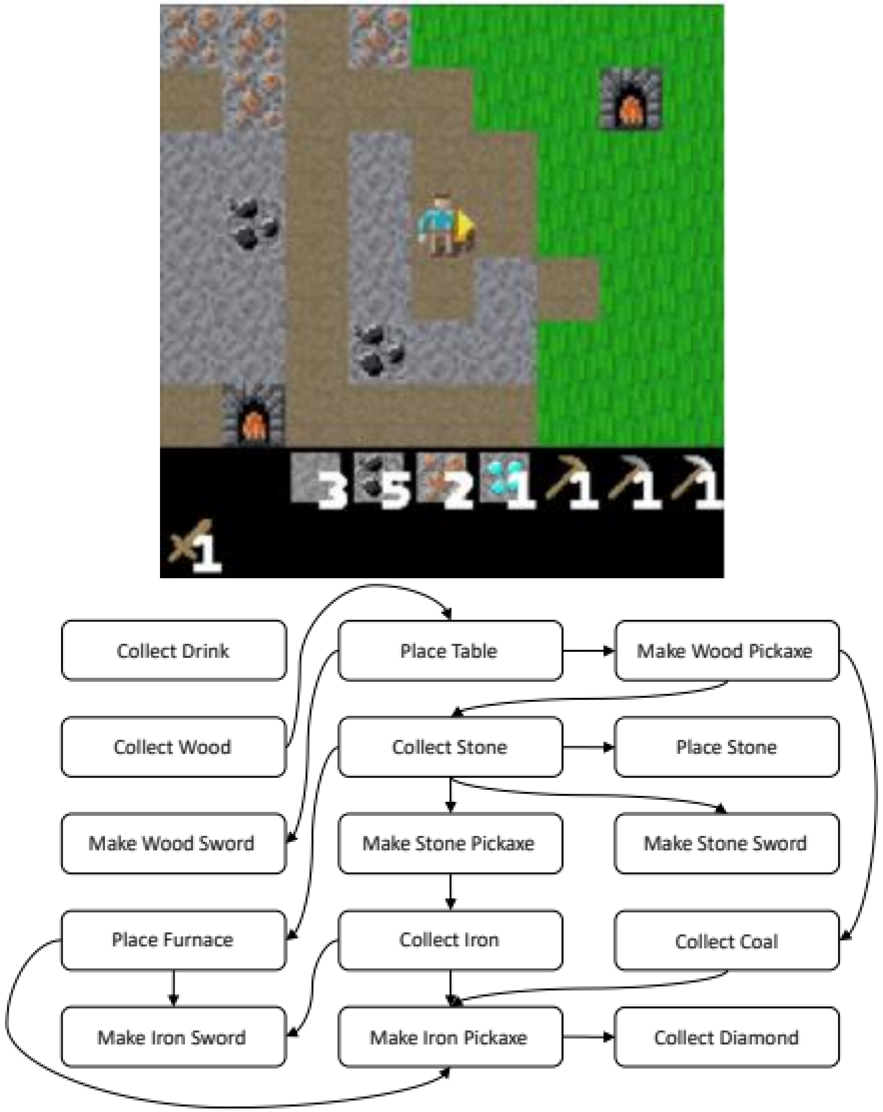

FIGURE 3 Above: Danijar Hafner 2021, *Crafter*. Screenshot from [1]. Below: Example of actions considered interesting, and the order in which they should be completed.

Relevance to Our Project

- We also want to choose relevant actions
- Generalized algorithm
  - It may be used even in different environments
- Other ways of using LLMs also possible
  - For reward shaping, instead of policy
- Interpretation of «interestingness»
  - Interesting = action with highest success rate?
  - Interesting = action most similar to other interesting actions?
    - OMNI algorithm assumes the two above
  - Interesting = (performed) action most similar to goal?

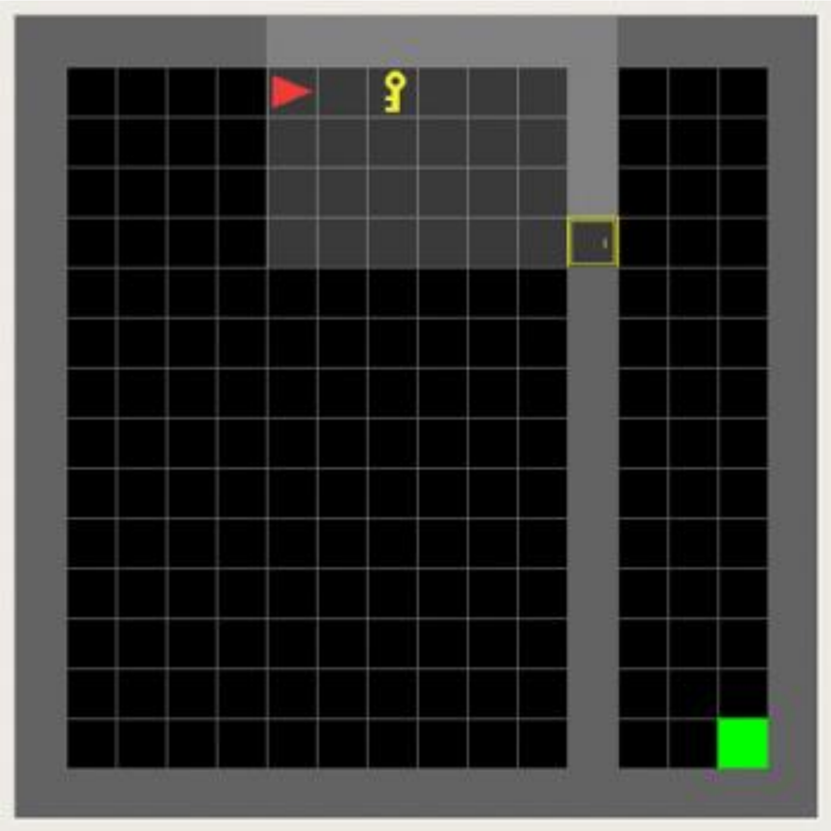

FIGURE 4 Minigrid, the testing environment we use in our project [2]. Screenshot from https://minigrid.farama.org/

References

[1] J. Zhang et al., «OMNI: Open-endedness via Models of human Notions of Interestingness». https://arxiv.org/abs/2306.01711

[2] M. Chevalier-Boisvert et al., «Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks». https://arxiv.org/abs/2306.13831