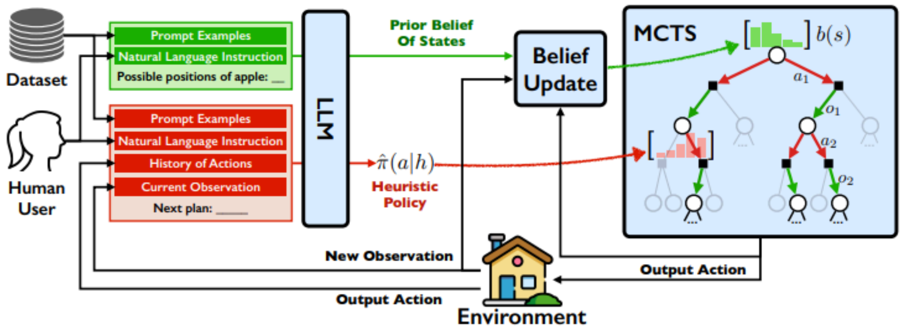
[NIPS'23] Large Language Models as Commonsense Knowledge for Large-Scale Task Planning

1. Link: https://llm-mcts.github.io/
2. Arthurs and institution: Zirui Zhao, Wee Sun Lee, David Hsu from NUS

TL;DR We use Large Language Models as both the commonsense world model and the heuristic policy within the Monte Carlo Tree Search framework, enabling better-reasoned decision-making for daily tasks.



Differences of Setups between ULbench and this

	ULBench	LLM-MCTS
Simulator	Coppeliasim	VirtualHOme
Scene	table top, 1 room	multiple rooms
object set	fixed	fixed
object position	random	random
task	fixed	random
action space	9 * action param	5 * action param

transition of action	stochastic/determined	determined

Existing problems

1. Even though LLMs are trained on internet-scale data, LLM-Policies shows limits in generalization.

Contributions

1. We use LLMs as the commonsense world model and the heuristic policy within MCTS to achieve better-reasoned decision-making for daily tasks.
2. We propose using the minimum description length (MDL) principle to decide the question “when would using LLM as a model outperform using LLM as a policy, and vice versa?”

Key concepts

Algorithm

Algorithm 1 LLM-MCTS

procedure SEARCH$(h, b, \mathcal{T}, N)$
    $n \leftarrow 0$
    while $n < N$ **do**
        $s \sim b(s)$
        SIMULATE$(s, h, \text{False}, 0, \mathcal{T})$
        $n \leftarrow n + 1$
    end while
    return $\text{argmax}_{a \in A}\, Q(h, a)$
end procedure
procedure ROLLOUT(s, h, done, d)
    if $\gamma^d < \epsilon$ or done $=$ True **then**
        return 0
    end if
    $a \sim \pi_{\text{rollout}}(h, \cdot)$
    $(s', o, r, \text{done}) \sim \mathcal{G}(s, a)$
    $h' \leftarrow$ PUSHBACK$(h, [a^*, o]), d' \leftarrow d+1$
    return $r + \gamma \cdot$ROLLOUT(s, h', done, d')
end procedure
procedure SIMULATE$(s, h, \text{done}, d, \mathcal{T})$
    if $\gamma^d < \epsilon$ or done $=$ True **then**
        return 0
    end if
    if h is not in $\mathcal{T}$ **then**
        $\mathcal{T} \leftarrow \mathcal{T} \cup h, N(h) \leftarrow 0$
        $\forall a \in A, N(h, a) \leftarrow 0, Q(h, a) \leftarrow 0$
        return ROLLOUT(s, h, done, d)
    end if
    $\hat{\pi}(a|h) \leftarrow$QUERYLLMPOLICY$(h)$
    $a^* \leftarrow \underset{a \in A}{\text{argmax}}\, Q(h, a) + c\hat{\pi}(a|h)\frac{\sqrt{N(h)}}{N(h,a)+1}$
    $(s', o, r, \text{done}) \sim \mathcal{G}(s, a^*)$
    $h' \leftarrow$ PUSHBACK$(h, [a^*, o]), d' \leftarrow d+1$
    $R \leftarrow r + \gamma \cdot$SIMULATE$(s', h', \text{done}, d', \mathcal{T})$
    $N(h, a^*)$ += 1, $N(h)$ += 1
    $Q(h, a^*) \leftarrow Q(h, a^*) + \frac{R - Q(h,a^*)}{N(h,a^*)}$
    return R
end procedure

1. The parts done by LLM are:
    1. L-Model
        1. generate the initial belief of state
        2. translate the natural language goal into a formal goal for MCTS
    2. L-policy

1. guide the action selection in the simulation procedure
2. sample the LLM for M times to approximate the policy probability distribution
3. empirical policy

cosine similarity $\text{CosineSim}(\alpha_i, a)$. The empirical policy distribution is formulated as follows: $\hat{\pi}(a|h) = \lambda \frac{1}{|A|} + (1-\lambda)\text{Softmax}\{\sum_{i=1}^{M} \text{CosineSim}(\alpha_i, a) - \eta\}$, where η is the average value of $\sum_i \text{CosineSim}(\alpha_i, a)$ and $|A|$ is the size of the admissible action space. λ is a hyper-parameter that adds randomness to the belief, as the sampled actions from LLM could be very deterministic.

MDL

Theorem 4.1 (Occam's Razor). *Let $\mathcal{H}$ be a hypothesis class and let $d : \mathcal{H} \to \{0,1\}^*$ be a prefix-free description language for $\mathcal{H}$. Then, for every sample size, m, every confidence parameter, $\delta > 0$, and every probability distribution, D, with probability greater than $1 - \delta$ over the choice of $S \sim D^m$ we have that, $\forall h \in \mathcal{H}, L_D(h) \leq L_S(h) + \sqrt{(|h| + \ln(2/\delta))/2m}$ where $L_S(h)$ is the empirical loss of h on the S, $L_D(h)$ is the expected loss of h, and $|h|$ is the length of $d(h)$.*

1. a method that suggests choosing the method that has a shorter description when both methods fit the training data. (the simpler the model ,the better the model, ceteris paribus)
2. selecting between a model or policy depending on which of them has the smaller description length given the description language.

Details

1. action space {pick(obj), place(obj, placement), open(container), close(container), move(room/obj/container)}
2. belief contruction
    1. predict the position of all moveable object with INSIDE/ON relationship. (sample of llm queried results)


```
{"alcohol": [["INSIDE", "kitchencabinet"], ["INSIDE", "kitchencabinet"],
["INSIDE", "kitchencabinet"], ["INSIDE", "kitchencabinet"], ["INSIDE",
"kitchencabinet"], ["INSIDE", "kitchencabinet"], ["INSIDE", "kitchencabinet"],
["INSIDE", "kitchencabinet"], ["INSIDE", "kitchencabinet"], ["INSIDE",
```


(frequency of an object)


```
"bathroom_cabinet": [["INSIDE", "bathroom", 50]],
```