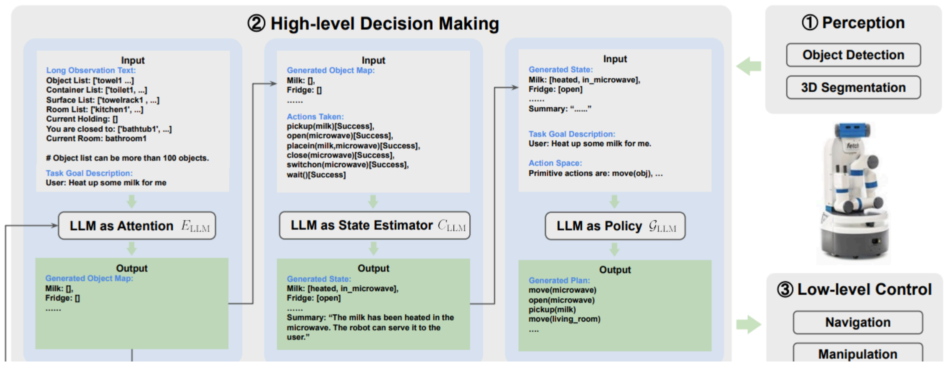
[NUS'23] LLM-State: Expandable State Representation for Long-horizon Task Planning in the Open World

1. Link: https://arxiv.org/pdf/2311.17406
2. Arthurs and institution: Siwei Chen, Anxing Xiao, David Hsu from NUS **TL;DR** we propose a novel state representation for task planning with LLM in the open-world, constructed and updated automatically by LLM for object state tracking and reasoning

comments and critisims

1. implicitly handling errors in a close loop (in LLM)
2. the system did not consider achieving optimal solution from the prompt setting

Key points


② High-level Decision Making
LLM as Attention E_{LLM}
LLM as State Estimator C_{LLM}
LLM as Policy $\mathcal{G}_{\text{LLM}}$
① Perception
Object Detection
3D Segmentation
③ Low-level Control
Navigation
Manipulation

```
Input
Long Observation Text:
Object List: ['towel1 ...]
Container List: ['toilet1, ...]
Surface List: ['towelrack1 , ...]
Room List: ['kitchen1', ...]
Current Holding: []
You are closed to: ['bathtub1', ...]
Current Room: bathroom1

# Object list can be more than 100 objects.
Task Goal Description:
User: Heat up some milk for me

Output
Generated Object Map:
Milk: [],
Fridge: []
......
```

```
Input
Generated Object Map:
Milk: [],
Fridge: []
......
Actions Taken:
pickup(milk)[Success],
open(microwave)[Success],
placein(milk,microwave)[Success],
close(microwave)[Success],
switchon(microwave)[Success],
wait()[Success]

Output
Generated State:
Milk: [heated, in_microwave],
Fridge: [open]
......
Summary: "The milk has been heated in the microwave. The robot can serve it to the user."
```

```
Input
Generated State:
Milk: [heated, in_microwave],
Fridge: [open]
......
Summary: "......"

Task Goal Description:
User: Heat up some milk for me.

Action Space:
Primitive actions are: move(obj), ...

Output
Generated Plan:
move(microwave)
open(microwave)
pickup(milk)
move(living_room)
....
```

Algorithm 1 Task Planning Execution Flow

```
Require: I: Goal Instruction, A: Primitive Actions
         O: Observed Objects, $E_{\text{LLM}}$: LLM as Attention
         $C_{\text{LLM}}$: LLM as State Estimator, $\mathcal{G}_{\text{LLM}}$: LLM as Policy
 1: Initialize history action list $A_h \leftarrow []$
 2: Initialize key objects $\hat{s}_0 \leftarrow []$
 3: while not done do
 4:     Update key objects $\hat{s}_t \leftarrow E_{\text{LLM}}(O_t, I) + \hat{s}_{t-1}$
 5:     Update state representation $s_t \leftarrow C_{\text{LLM}}(\hat{s}_t, A_h, O_t)$
 6:     Generate $n$ actions $a_{t:t+n} \leftarrow \mathcal{G}_{\text{LLM}}(s_t, O_t, I)$
 7:     for $a_t$ in $a_{t:t+n}$ do
 8:         $O_t, r_t \leftarrow$ execute($a_t$) # $r_t$: binary success indicator
 9:         Append actions, $A_h \leftarrow A_h + (a_t, r_t)$
10:         if not $r_t$ then
11:             break # action failed, re-plan
12:         end if
13:     end for
14: end while
```

Inputs

1. Goal Instruction in text
2. Step-wise Observation
    1. current location
    2. observed objects
    3. rooms
    4. pre-defined primitave actions

Outputs

1. Step-wise Primitive Action Commands

State representation

1. structured obj entry

        1. a dictionary of objects, where key is the object name and values are attributions of objects (status, location, condition)
    2. Unstructured Retrospective Summary Entry
        1. action histroy of robot
            1. helps to predict attributes of objects
        2. recent failures
            1. unknown preconditions
            2. unexpected transitions

Modules

LLM as Attention

1. extend the list of key objects relevant to the task based on the latest observed object list
2. functions
    1. add_related_object()

LLM as state estimator

1. uses the key objects list, historically executed action with binary success indicator, and observation to generate the state representation with state change for the object in the key objects list and the summary
2. functions
    1. generate_summary(text)
    2. add_attributes(obj, attr)

LLM as Policy

1. generates the action sequences based on the current state representation, observation and instruction.

Exploration

1. For objects not present in the current state but required to complete a task, the LLM policy actively generates hypotheses using common sense.

Details

1. perception

    1. 6 angles to fully cover 360 degree
    2. Recognize anything + GroundingDINO
    3. SAM
    4. back-projetion
2. low-level control
    1. navigation
        1. use grid map and Ros-build-in motion planning
    2. manipulation
        1. imitation learning with VR controller
        2. 30 trajectories per primitave skill
3. hardware
    1. running in a 4090 GPU
    2. stop by human monitor if the generated action is not callable
4. simulation: virtualhome