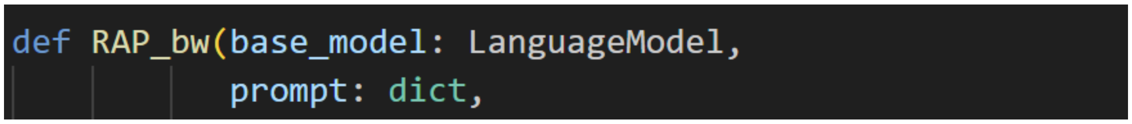
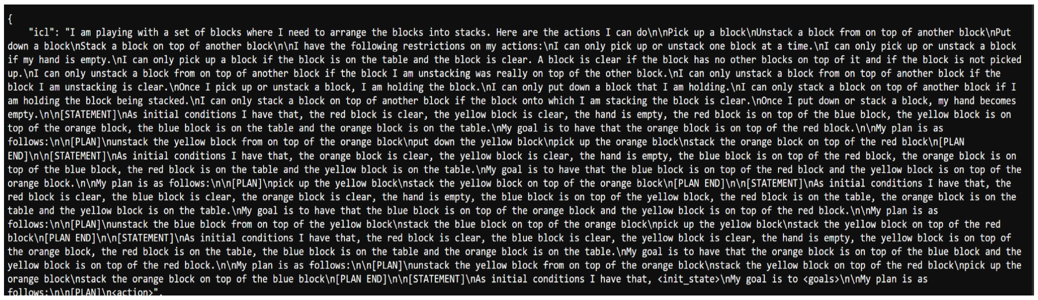
CSE 576: Topics in NLP
Class Project : Team LLM Parrots

1. **Week 1 (30th October, 2024 - 6th November, 2024):**
    a. Studied and Analyzed the paper to understand the code structure for RAP and LATS. (going through the github repository)
    b. Requested Access for Hugging Face Model: Llama-3.1-8B-instruct.
    c. Attempted to implement RAP algorithm Zero-shot using LLama through demo.ipynb available in the GitHub using the Preprocessed Dataset.
    Issues faced while doing the zero shot implementation:
        1. The RAP is implemented with a prompt dictionary as input which is accessed from the path
        `‘examples/CoT/blocksworld/prompts/prompt.json'`



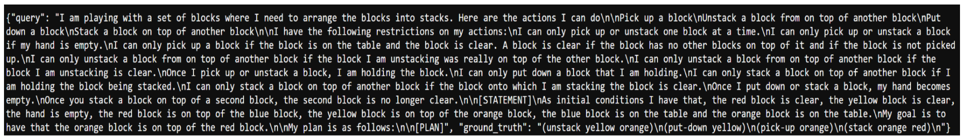
```
def RAP_bw(base_model: LanguageModel,
           prompt: dict,
```

        2. The structure of the prompt is as follows:



```
{
    "icl": "I am playing with a set of blocks where I need to arrange the blocks into stacks. Here are the actions I can do\n\nPick up a block\nUnstack a block from on top of another block\nPut
down a block\nStack a block on top of another block\n\nI have the following restrictions on my actions:\nI can only pick up or unstack one block at a time.\nI can only pick up or unstack a block
if my hand is empty.\nI can only pick up a block if the block is on the table and the block is clear. A block is clear if the block has no other blocks on top of it and if the block is not picked
up.\nI can only unstack a block from on top of another block if the block I am unstacking was really on top of the other block.\nI can only unstack a block from on top of another block if the
block I am unstacking is clear.\nOnce I pick up or unstack a block, I am holding the block.\nI can only put down a block that I am holding.\nI can only stack a block on top of another block if I
am holding the block being stacked.\nI can only stack a block on top of another block if the block onto which I am stacking the block is clear.\nOnce I put down or stack a block, my hand becomes
empty.\n\n[STATEMENT]\nAs initial conditions I have that, the red block is clear, the yellow block is clear, the hand is empty, the red block is on top of the blue block, the yellow block is on
top of the orange block, the blue block is on the table and the orange block is on the table.\nMy goal is to have that the orange block is on top of the red block.\n\nMy plan is as
follows:\n\n[PLAN]\nunstack the yellow block from on top of the orange block\nput down the yellow block\npick up the orange block\nstack the orange block on top of the red block\n[PLAN
END]\n\n[STATEMENT]\nAs initial conditions I have that, the orange block is clear, the yellow block is clear, the hand is empty, the blue block is on top of the red block, the orange block is on
top of the blue block, the red block is on the table and the yellow block is on the table.\nMy goal is to have that the blue block is on top of the red block and the yellow block is on top of the
orange block.\n\nMy plan is as follows:\n\n[PLAN]\npick up the yellow block\nstack the yellow block on top of the orange block\n[PLAN END]\n\n[STATEMENT]\nAs initial conditions I have that, the
red block is clear, the blue block is clear, the orange block is clear, the hand is empty, the blue block is on top of the yellow block, the red block is on the table, the orange block is on the
table and the yellow block is on the table.\nMy goal is to have that the blue block is on top of the orange block and the yellow block is on top of the red block.\n\nMy plan is as
follows:\n\n[PLAN]\nunstack the blue block from on top of the yellow block\nstack the blue block on top of the orange block\npick up the yellow block\nstack the yellow block on top of the red
block\n[PLAN END]\n\n[STATEMENT]\nAs initial conditions I have that, the red block is clear, the blue block is clear, the yellow block is clear, the hand is empty, the yellow block is on top of
the orange block, the red block is on the table, the blue block is on the table and the orange block is on the table.\nMy goal is to have that the orange block is on top of the blue block and the
yellow block is on top of the red block.\n\nMy plan is as follows:\n\n[PLAN]\nunstack the yellow block from on top of the orange block\nstack the yellow block on top of the red block\npick up the
orange block\nstack the orange block on top of the blue block\n[PLAN END]\n\n[STATEMENT]\nAs initial conditions I have that, <init_state>\nMy goal is to <goals>\n\nMy plan is as
follows:\n\n[PLAN]\n<action>",
```

        Highlights of the prompts.json is that we are
            a) It has a field like world_update_pickup which gives scenarios based on the states and what change needs to be done to do the pickup and the next state after the pickup is done.
            b) Similarly, world_update_stack, world_update_unstack, world_update_putdown
            c) The amount of information given in the prompt is very very vast and informative and the RAP is implemented using such a huge prompt. It was the main challenge we faced to replicate the RAP repository for a Zero-shot encoding. As in zero-shot encoding we won't be giving the LLM such information.(A sample query from the zero-shot prompt dataset we need to utilize in this project is given below)

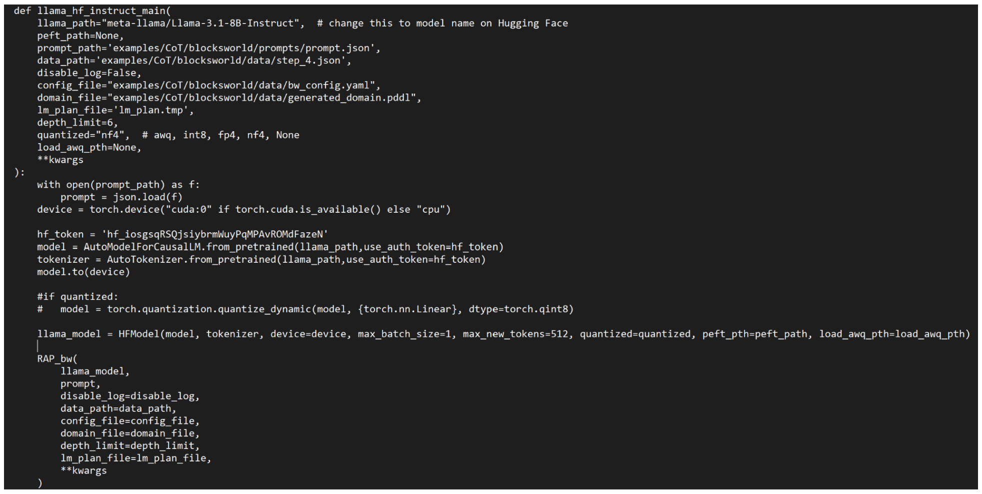


```
{"query": "I am playing with a set of blocks where I need to arrange the blocks into stacks. Here are the actions I can do\n\nPick up a block\nUnstack a block from on top of another block\nPut
down a block\nStack a block on top of another block\n\nI have the following restrictions on my actions:\nI can only pick up or unstack one block at a time.\nI can only pick up or unstack a block
if my hand is empty.\nI can only pick up a block if the block is on the table and the block is clear. A block is clear if the block has no other blocks on top of it and if the block is not picked
up.\nI can only unstack a block from on top of another block if the block I am unstacking was really on top of the other block.\nI can only unstack a block from on top of another block if the
block I am unstacking is clear.\nOnce I pick up or unstack a block, I am holding the block.\nI can only put down a block that I am holding.\nI can only stack a block on top of another block if I
am holding the block being stacked.\nI can only stack a block on top of another block if the block onto which I am stacking the block is clear.\nOnce I put down or stack a block, my hand becomes
empty.\nOnce you stack a block on top of a second block, the second block is no longer clear.\n\n[STATEMENT]\nAs initial conditions I have that, the red block is clear, the yellow block is clear,
the hand is empty, the red block is on top of the blue block, the yellow block is on top of the orange block, the blue block is on the table and the orange block is on the table.\nMy goal is to
have that the orange block is on top of the red block.\n\nMy plan is as follows:\n\n[PLAN]", "ground_truth": "(unstack yellow orange)\n(put-down yellow)\n(pick-up orange)\n(stack orange red)\n"}
```

d. Failed to reach baseline Zero-shot RAP as the algorithm required prompt dataset

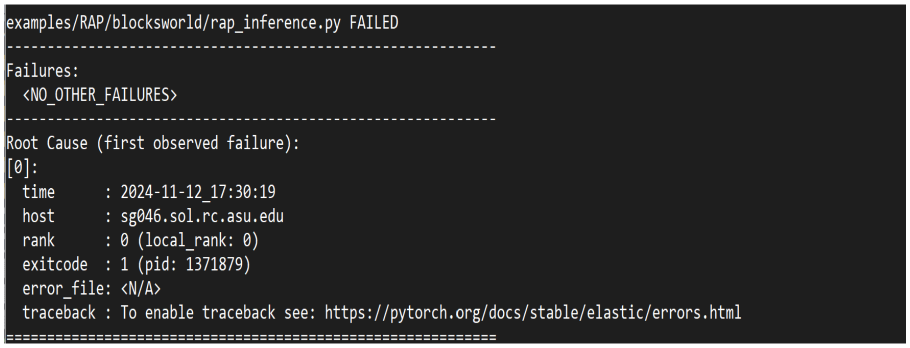
3. Repurposed to Implement RAP and LATS as is, based on how the original code is structured.

2. **Week 2 (7th November, 2024 - November 13th, 2024):**
   a. Code adjustment for RAP in Llama 3.1-8b-Instruct.
      We are in-progress with an implementation of the RAP algorithm for the blocksworld domain using Llama-3.1-8B-Instruct.
      Some roadblocks faced in the implementations:
      1. Added the new model in the rap_interface.py



```
def llama_hf_instruct_main(
    llama_path="meta-llama/Llama-3.1-8B-Instruct",  # change this to model name on Hugging Face
    peft_path=None,
    prompt_path='examples/CoT/blocksworld/prompts/prompt.json',
    data_path='examples/CoT/blocksworld/data/step_4.json',
    disable_log=False,
    config_file="examples/CoT/blocksworld/data/bw_config.yaml",
    domain_file="examples/CoT/blocksworld/data/generated_domain.pddl",
    lm_plan_file='lm_plan.tmp',
    depth_limit=6,
    quantized="nf4",  # awq, int8, fp4, nf4, None
    load_awq_pth=None,
    **kwargs
):
    with open(prompt_path) as f:
        prompt = json.load(f)
    device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")

    hf_token = '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'
    model = AutoModelForCausalLM.from_pretrained(llama_path,use_auth_token=hf_token)
    tokenizer = AutoTokenizer.from_pretrained(llama_path,use_auth_token=hf_token)
    model.to(device)

    #if quantized:
    #   model = torch.quantization.quantize_dynamic(model, {torch.nn.Linear}, dtype=torch.qint8)

    llama_model = HFModel(model, tokenizer, device=device, max_batch_size=1, max_new_tokens=512, quantized=quantized, peft_pth=peft_path, load_awq_pth=load_awq_pth)

    RAP_bw(
        llama_model,
        prompt,
        disable_log=disable_log,
        data_path=data_path,
        config_file=config_file,
        domain_file=domain_file,
        depth_limit=depth_limit,
        lm_plan_file=lm_plan_file,
        **kwargs
    )
```

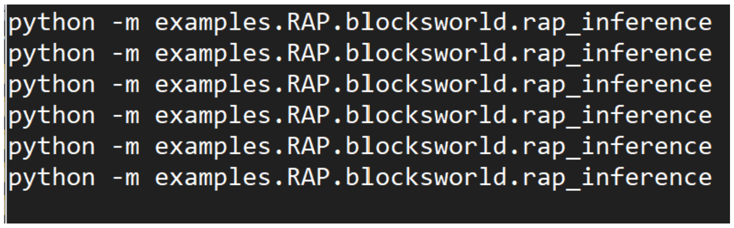
      Made changes in the rap_interface, test_llama2.sh for the model.
      2. We tried the test_llama2.sh with torch.distributed.run --nproc_per_node 1 but into the following errors



```
examples/RAP/blocksworld/rap_inference.py FAILED
------------------------------------------------------------
Failures:
  <NO_OTHER_FAILURES>
------------------------------------------------------------
Root Cause (first observed failure):
[0]:
  time      : 2024-11-12_17:30:19
  host      : sg046.sol.rc.asu.edu
  rank      : 0 (local_rank: 0)
  exitcode  : 1 (pid: 1371879)
  error_file: <N/A>
  traceback : To enable traceback see: https://pytorch.org/docs/stable/elastic/errors.html
============================================================
```

      To fix the following issues we ran it without the torch.distributed.run

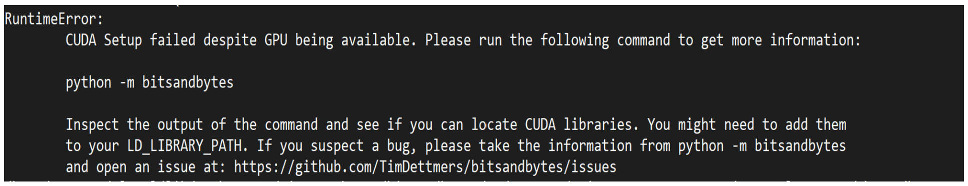
3. After some modifications and running the python file as script using -m the following changes were made in the test_llama2.sh file



```
python -m examples.RAP.blocksworld.rap_inference
python -m examples.RAP.blocksworld.rap_inference
python -m examples.RAP.blocksworld.rap_inference
python -m examples.RAP.blocksworld.rap_inference
python -m examples.RAP.blocksworld.rap_inference
python -m examples.RAP.blocksworld.rap_inference
```

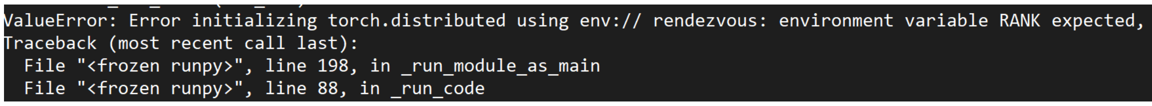
This got rid of the issue in point 1 , as well as the ModuleNotFound errors

4. The following error was coming after making the above modifications



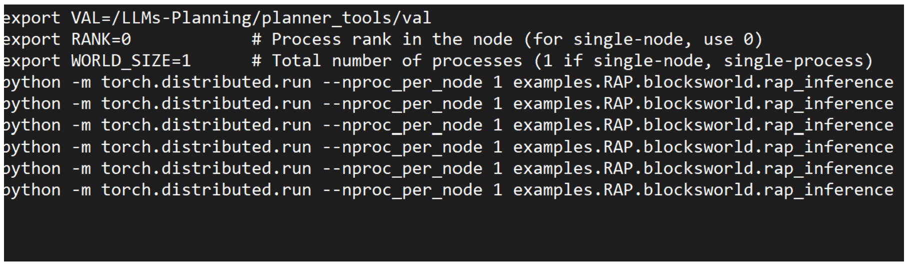
```
RuntimeError:
        CUDA Setup failed despite GPU being available. Please run the following command to get more information:

        python -m bitsandbytes

        Inspect the output of the command and see if you can locate CUDA libraries. You might need to add them
        to your LD_LIBRARY_PATH. If you suspect a bug, please take the information from python -m bitsandbytes
        and open an issue at: https://github.com/TimDettmers/bitsandbytes/issues
```

5. Removing the torch.distributed.run created the following issues.



```
ValueError: Error initializing torch.distributed using env:// rendezvous: environment variable RANK expected,
Traceback (most recent call last):
  File "<frozen runpy>", line 198, in _run_module_as_main
  File "<frozen runpy>", line 88, in _run_code
```

6. Made some changes in the sh file to fix this error



```
export VAL=/LLMs-Planning/planner_tools/val
export RANK=0           # Process rank in the node (for single-node, use 0)
export WORLD_SIZE=1     # Total number of processes (1 if single-node, single-process)
python -m torch.distributed.run --nproc_per_node 1 examples.RAP.blocksworld.rap_inference
python -m torch.distributed.run --nproc_per_node 1 examples.RAP.blocksworld.rap_inference
python -m torch.distributed.run --nproc_per_node 1 examples.RAP.blocksworld.rap_inference
python -m torch.distributed.run --nproc_per_node 1 examples.RAP.blocksworld.rap_inference
python -m torch.distributed.run --nproc_per_node 1 examples.RAP.blocksworld.rap_inference
python -m torch.distributed.run --nproc_per_node 1 examples.RAP.blocksworld.rap_inference
```

Still have been getting errors, We haven't received a session after this experimentation.

b. Replication of LATS implementation for the given dataset
   i. We are in-progress with an implementation of the LATS algorithm specifically for the blocksworld domain
   ii. The major challenges that we faced while implementing the algorithm included state extraction from prompts, an accurate reward metric for simulation, computing new states based on the recommended next action by the LLM
      1. State extraction from prompts

We implemented a python parser to convert the initial state from prompts into an object as follows:

```
Clear blocks: {'red', 'yellow'}, Hand: None, Block on top
positions: {'red': 'blue', 'yellow': 'orange', 'blue': 'table',
'orange': 'table'}
```

2. Reward metric for simulation
   The initial thought was to rely on the LLM itself to give us the reward. For this we have a “reward prompt” which takes the current state and asks the LLM to estimate the number of steps required to reach the goal state. We then use the number of states to derive a metric. Ex (100 - n)/100, where n = no. of steps)

   Another approach is to derive this metric programmatically by comparing the relative positions of blocks in the current and the goal state. We are still working on this.

3. Computing new states
   We created prompts to derive new states from the current state and the given action by the model. Here, we rely on the model to take us from current to next state. We observed the next states generated by the model are not that accurate. We could also derive the next steps programmatically which would be more accurate but would take away model contribution from the solution.