LLM-RSPF: Large Language Model-based Robotic System Planning Framework for Domain Specific Use-cases

Chandan Kumar Singh, Devesh Kumar, Vipul Sanap, Rajesh Sinha
Tata Consultancy Services (TCS) Research, New Delhi, India
{ck.singh1, k.devesh1, vipul.sanap1, rajesh.sinha}@tcs.com

1. Additional Implementation Details of the Retail Order Fulfilment Use-case

The detailed description of some of the modules of the LLM-RSPF ontology corresponding to the retail order fulfilment system discussed earlier are provided below:

- State: The <agent_1> state considered here is the robot EOAT, i.e., <current_eoat>. Since each picking skill defined above uses a different EOAT, therefore, the robot EOAT as a State is incorporated.

- Interfacing: It represents a commentarial communication within intra-Agent, inter-Agent or Agent-User. Here, the commentaries include, intra-Agent communication of object transfer messages between two robots and Agent-User task-relevant conversations.

- Experience: For this domain-specific use-case (DSU), only failure scenarios are considered to avoid Re-Failure scenarios.

- Pick location: Bin1 contains heterogeneous set of objects and is intended for object from high mix low volume category. Bin2 contains homogeneous set of objects. Bin3 contains homogeneous set of objects, however, different from the one available in Bin2.

- Drop location: Drop1 (Conveyor) is a flat belt conveyor to transfer objects from <robot_1>to <robot_2>. Drop2 (Box) is a carton or a retail package box to send objects for packaging or order fulfilment. Drop3 (User) is a convenient fixed location intended for the user retrieval.

- Arrangement: The <robot_1> and <robot_2> are at the front and rear-end of the Conveyor, respectively.

Concerning the robotic manipulation skills of the Embodiment, there are specific reasons behind the choice of adopting domain-independent picking skill (DIPS) and instance-retrieval picking skill (IRIS). These skills are inspired from a Good-to-Picker model in retail order fulfilment. DIPS is advantageous when the system has to adapt quickly to new objects without any deep learning-based training. DIPS is useful for homogeneous items as these can be changed entirely and the system can quickly adapt with it. On the other hand, IRIS targets heterogeneous items, which are intended for object classes from high mix low volume category, a quite popular concept in order fulfilment industry. It helps in achieving high space utilization.

- Mapping: Bin1 $\rightleftharpoons$ <iris>; Bin2 $\rightleftharpoons$ <dips>; Bin3 $\rightleftharpoons$ <dips>; <iris> $\rightleftharpoons$ <suction>; <dips> $\rightleftharpoons$ <2fgripper>.

- Classification:
  - Valid: It results in a valid plan as all plan-relevant information is available in the instruction.
  - Invalid: It requests unreasonable tasks provided the robotic system's capabilities.
  - General: It seeks irrelevant queries about the robotic system.
  - System information instruction (SII): SII seeks system related information such as Workspace details, Embodiment details, etc.
  - Additional data request (ADR): ADR is related to the robotic system and almost valid, however, any one piece of information is missing.

2. Detailed Description of Dataset

To start with estimating the dataset unit size, the total instruction classifications m is set as 5 according to the definitions from the Mapping module. The total robotic skills S defined in the Action module are 3. As a result, the total combinations becomes 3 as there is a single <agent_1>. In this work, considering 5 instructions as a sufficient spanning factor against each instruction classification, the domain rules infusion part comes out to be 40. Now, concerning the few-shot tuning part, i.e., diversity inclusion, the ϵ' is kept as 5. Thus, a unit size estimate comes out to be 40 instructions. Following the dataset division ratio of 1:0.5:2, the training, validation, and testing size are 40, 20, and 80, respectively. Theses 140 instructions are now created using 14 human oracles. Each oracle is briefed about the framework, its purpose, and different classifications in advance. The pool of 140 instructions are reviewed through an elementary quality check to ensure format consistency. Subsequently, this instruction set is categorized into five capability tests that helps us to detect and understand any possible bias in the LLM: (a) Unitary action (UAT) (b) Task complexity (TCT) (c) Contextual understanding (CUT) (d) Diversity and consistency (DCT) (e) Sanity and robustness (SRT). Such a way of human oracle-based annotation and five-bucket division helps in opening the scope for having sufficient diversity in the dataset, which also caters to the diversity inclusion part. Finally, this high-quality human-annotated instruction set becomes ready for Prompt-tuning.

3. Additional Information on Prompting

Initially, the CoHT fundamentally exercises few-shot prompting. Any DSU is interpreted and transformed into an elementary prompt. Next, the standard prompting style of using placeholders, contextual separations using symbolic cues, usage of tone or style, context priming, etc., is used to structure and bring clarity to the prompting. This also helps in setting optimal positioning of the textual domain rules. Moving forward, the next portion of the prompting is defining few-shot examples. It is observed that in DSUs, the LLM's complex reasoning abilities are significantly governed by the domain examples as compared to abstractly representing the domain rules in the form of contextually separated paragraphs. As a consequence, it becomes inevitable to carefully craft the few-shots with sufficient diversity for the LLM to gain contextual understanding in appropriate plan generation. One of the key advantage of the CoHT is the scope of attesting hard-bound rules with each hierarchical thought within a linear reasoning step, which significantly helps in improving LLM's reasoning for any specific event that otherwise is quite difficult to achieve.

Although increasing the few-shot examples improves the LLM's performance, however, it is observed that by keeping a single highly complex example in place of multiple simple examples improves the LLM's performance substantially. This serves the other purpose as well to reduce the token length by replacing multiple examples with a single one. The aforementioned observation can be confirmed through the Fig. 1. Here, simple, moderate, and complex denote the task plan complexity.

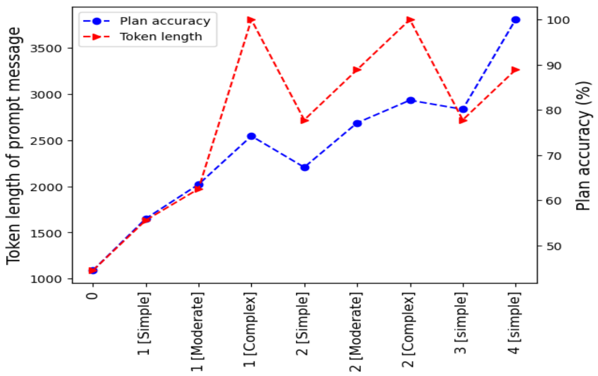

Figure 1. Performance of CoHT-Regimen on varying shots and token length

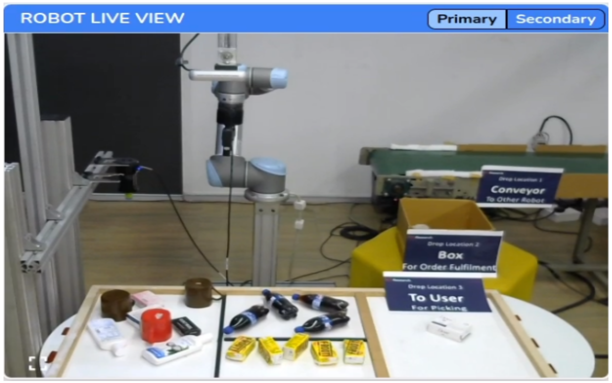

Figure 2. Primary robot for the scalability study

Figure 3. Secondary robot for the scalability study

4. Additional Information on Scalability of the LLM-RSPF

At first, an illustration of both the robots used in the scalability study are shown in Fig. 2 and Fig. 3, respectively. Next, concerning the extended dataset, the number of instruction classifications remains the same. The robotic skills are now increased to 5, i.e., initial 3 skills (refer Section 3) mapped to primary robot, whereas, the other 2 mapped to the secondary robot. These 2 skills include, an altered version of DIPS for secondary robot and a custom manipulation skill for opening and closing the drawer. Note, the pick location is conveyor for the secondary robot, whereas drop location is a drawer. The agent count here increases to 2. Considering the ϵ' as 5, and 5 instructions per class, the dataset unit size becomes 50 as per formula (5) given in Section 3. Subsequently, the training, validation, and testing dataset size becomes 50, 25, 100, respectively. Note, with the inclusion of tactile perception sensing capability in addition to visual scene information, there is an addition into the dataset input. The tactile quality of each object is also provided as an input along with the detected objects from the workspace. The usage of tactile perception helps in slip detection and slip prevention of objects, while grasping and transferring them to drawer.