Accelerating PDDL Planning Time through Subgoal Generation Leveraging Large Language Models

Kaitlyn Kwan, GLAMOR Lab
Arcadia High School, Class of 2024
USC Viterbi Department of Computer Science, SHINE 2023

Introduction

Through a synthesis of the Large Language Model (LLM) and the Planning Domain Definition Language (PDDL), LLM+P [1] provides an accurate and efficient plan to solve common tasks. The LLM takes in a natural language problem and translates the natural language problem into a corresponding PDDL file based on a context example consisting of a different set of a natural language problem and the corresponding PDDL problem file. The downward planner then comes up with a PDDL solution and utilizes the LLM to translate the solution back into natural language. The LLM+P planning method has restrictions where the planner time increases exponentially based on the complexity of the environment.

Impact of Professor's Research

My professor, Jesse Thomason, leads the GLAMOR Lab. Professor Thomason's research ties natural language processing and robotics together. The LLM+P with subgoals allows for more effective creation of solution plans to tasks. The new method would significantly cut down on planner time as environments get more complex.

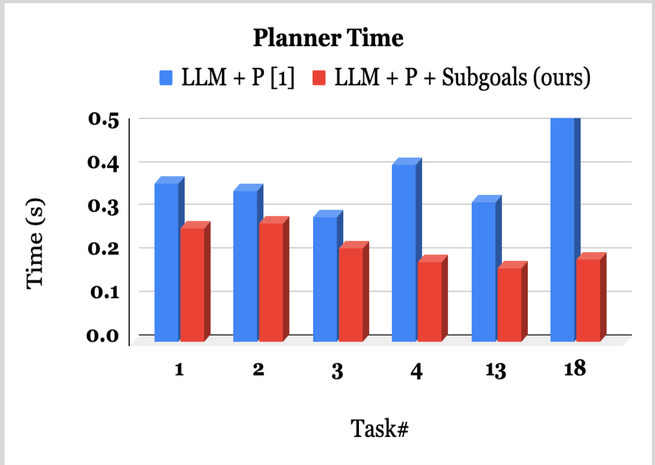
Research & Learning Process

To reduce the planner time, I had to update the context files to reflect the change I hope to see in the newly generated PDDL problem file from the LLM. Each PDDL file has an environment description, current initial conditions, and goal conditions. In order to prompt the LLM to generate several PDDL files with different sequential initial and goal conditions, I split the PDDL context problem file into 2 files. Utilizing the built-in validator and employing my knowledge in natural language processing, I figured out which initial and goal conditions should be altered at each subgoal and derived a few experimental prompts for the LLM. In order to conveniently test accuracy and planner time, I implemented my skills in python programming to automate prompting, planning, and validation for all tasks. Each prompting attempt was trialed on 20 different tasks, varying in completion difficulty and environment complexity.

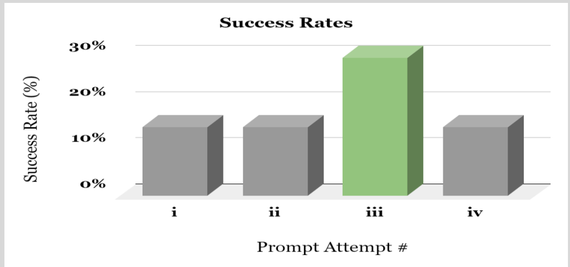
Examples of Attempted Prompting and Success Rate:

i) Environment description, example initial & goal, example initial & subgoals, task initial & goal, LLM produce task initials & subgoals (15%)
ii) Example problem description, example initials & subgoals, task description, LLM produce task initials & subgoals (15%)
iii) Example initial & goal, example initial & subgoals, task initial & goal, LLM produce task initials & subgoals (30%)
iv) Environment description, example problem description, example initials & subgoals, task description, LLM produce task initials & subgoals (15%)

Methods & Results



Failed prompting attempts for tasks either didn't have a proper solution or didn't solve the task correctly according to the validator. Every successful task underwent 5 trials using the LLM+P+Subgoals method and the average planner time was observed in comparison to the original LLM+P planner time. With a success rate of 30% LLM+P+Subgoals reduced the planner time by approximately 55.18% on average. As environments got more complex, the reduction on planner time increased more notably. Further steps should be taken to improve the accuracy while maintaining the speed.



iii)

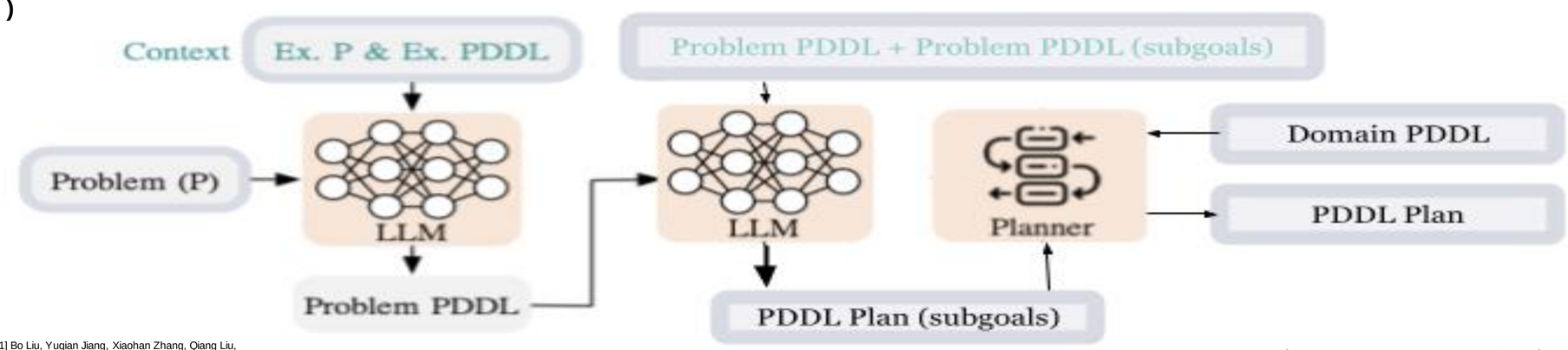


image source: LLM+P, Liu et al 2023

[1] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. 2023a. LLM+P: empowering large language models with optimal planning proficiency. CoRR, abs/2304.11477.

Next Steps for You

From this experience with SHINE, I hope to leverage my newfound knowledge and skills to improve practical applications and utilizations of computer science to optimize the diverse interactions, conversations, and experiences of people worldwide. From being proactive and maintaining a positive work habit, I've learned to never be afraid to ask questions, propose ideas, or simply get out of my comfort zone.

Acknowledgements

I would like to thank my professor, Jesse Thomason, and lab mentors, Ishika Singh and Abrar Anwar, for teaching me new skills, guiding me throughout my project, and allowing me to work in their lab.