Using LLM and LTL to Enhance Human-Robot Cooperation

Yichen Wei, Naicheng He
Brown University
{yichen_wei, naicheng_he}@brown.edu

Abstract

Recent development in Large Language Models have provided AI a general way to understand and encode natural language. We propose a framework that enables multiple robots to collaborate with human on tasks, by communicating with human on task specifications and task allocation, through translation of natural language into Formal Logic and Linear Temporal Logic. We present state-of-the-art translation and simulation performance of our approach in navigation tasks.

1 Introduction

With the development of modern AI algorithms and robotics, robots have been able to do more tasks in all ways than ever before. A significant scenario is for the robots to receive commands and collaborate with human in completing certain task.

As a motivating example, imagine a Human trying to collaborate with the robot clean up a space, and the human's goal is to vacuum the room before mopping the room. Additionally, the human wish to clean up room 1 and room 2 by themselves because there are delicate machines in the room where they do not wish the robot to go to. The robot need to understand the task and the orders, and understand the preferences of the human.

Due to the vague and fuzzy nature of natural languages [17, 6], it is not straightforward to translate natural languages directly into commands, plans, or logical expressions. Yet natural language is such a crucial part of human-robot interaction. To allow robots to be dependable and trustworthy, it is crucial to correctly interpret human's instructions and understand human's intentions. Many approaches of robotics and natural language processing require human to issue command in some specific format to ensure that the robot can properly interpret the commands. This puts a huge assumption on what human can do using natural languages.

It is thus crucial to find solutions where it is easy and reliable for human to collaborate with robots on complex tasks using natural language. To this end, we propose a general framework that is able to convert both task specifications and human's intentions into logical expressions to enable reliable human-robot collaboration.

2 Approach and Methods

To turn natural language into plans executable by a multiple robots, we take a learning-planning hybrid, two-step approach. The first step is to convert the natural language into logical formulas using Large Language Models (LLM). Then the logical formula is converted into a finite state machine on the main station for planning. Lastly, at every time step, the robots receive high-level discrete actions from the planner and uses their onboard navigation or motion planning algorithm to achieve the goal. Our approach also has flexibility when new commands are issued, as logical formulas can be updated on-the-fly and the plans can be updated accordingly.

Preprint. Under review.

2.1 Translation of Natural Language into logic using Large Langauge Models (LLM)

The first part of the process is to leverage Large Language Models to encode natural language into a form our planner can understand. Large Language Models are transformers [15, 3] which are pre-trained with rich information scraped from the internet. With the breadth of data that it was trained on, it is especially good at understanding different sentence structures and is a good fit for a general-purpose language modeling framework.

We classify human-robot collaboration tasks into two separate categories, human issuing commands and human giving task preferences when doing the research.

2.1.1 Translating Commands using Linear Temporal Logic (LTL)

In the case of human issuing temporally extended commands, a type of formal language is needed to capture the relationship between the objects in the space. Linear temporal Logic is an extension of formal logic which includes two new operators, **N** and **U**, meaning next and until, respectively [1]:

$$\phi ::= p \mid \neg\phi \mid \phi \vee \psi \mid \mathbf{N}\phi \mid \phi \mathbf{U} \psi \qquad \text{where } p \in \mathcal{P}$$

Since LTL formulas are evaluated over temporally extended states and observations, they are able to capture long horizon tasks requiring multiple different subtasks to be done, and capture the dependency in the tasks. Intuitively, $\mathbf{N}\phi$ holds true if ϕ is true in the next time step. $\phi \mathbf{U} \psi$ holds true if ϕ always holds true until ψ holds true.

In addition to those operators, more operators can be defined as a shorthand, which are a combination of existing operator. Those are the $\wedge$, **F**, and **G**operators. Intuitively, $\mathbf{F}\phi$ means that eventually, ϕ should be true at least once. Lastly, $G\phi$ means that ϕ should be always true.

Prior work has shown that large language models have been effective in translating natural language instructions into LTL formulas, from 50% to 90% depending on the Large Language Models used. [9]. We build on top of prior work and extend the approach and the dataset to multi-robot domains by adding necessary operators. By using the updated dataset from [9], we may utilize techniques of finetuning [11, 8] to efficiently tune the model to predict the correct LTL formula given the natural language input.

2.1.2 Translating Human Task Preferences during Task Allocation

In the case of understanding human preferences when allocating tasks, we wish to understand human's real intention. To achieve this, we need to capture both who is the sentence is referring to, and whether the sub-task mentioned should be acted upon or not. For example, "I want to do a" means the person needs to do a, while "You should do a" means that the robot should do task a. While the previous commands are clear commands that may be easy to interpret, sentences like "I think I really need help with a", or "I really hate doing a" expresses the person's frustration, which requires more understanding of the sentiments and context than simple rule-based translation.

The current iteration of our approach translates the human's preference into logical expressions specifying which sub-task the robot should or should not do. If the human expresses that they wish not to do "b", the logical expression will be b, meaning that b will be a task for the robot to complete, and vice versa. Future Iterations of the work may add urgency of the user's request to further optimize the human-robot coordination experience.

This type of task is best suited for an instruction fine-tuned Large Language Models, as it's tuned on human conversations and has superior ability in understanding human's needs. [16]

2.2 Multi-Agent Planning

With the aformentioned LTL task specification and the cases we proposed, We will make an abstraction of the procedure we use to make multi-agent plans with Linear Temporal Logic.

Firstly, we utilize spot [4] to convert LTL formula into DFA's [cite]. Impossible edges are marked as disabled. One such DFA can be seen in figure 1. The States Q are marked by numbers from 0 to n, with the final condition qn = 0 marking the end of execution. We assign a transition function θ so that it returns a 'cost' of that transition. Impossible transitions are marked and discarded, while optimizations were made on the other.

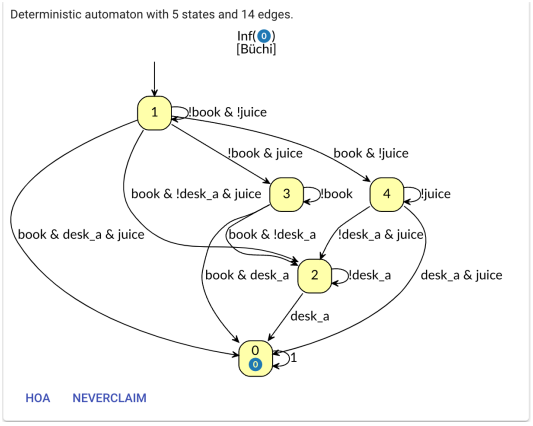

Figure 1: $F(\text{book} \wedge F\text{desk\_a}) \wedge F(\text{juice} \wedge F\text{desk\_a})$

When human-robot interaction happens, we maintain two linear temporal logic sentence, one for the entire execution the other for human preferences. At each time step we will mask the human-preferences if they are true, and a transition function with un-fulfilled human-preferences is considered impossible.

Algorithm 1 Planning Algorithm

procedure LTLPLAN(ltl, p) ▷ The LTL and human-preference translated from natural language
  $r \leftarrow q_0$ ▷ Current State is initial state
  $plan \leftarrow \langle\rangle$
  while $r \neq q_n$ **do** ▷ Until r is in goal/terminal state
    θ_r = min{cost (θ) $||\theta$ doesn't violate p}
    ▷ Find the lowest multi-agent cost transition.
    Update(robots_actions) ▷ Based on the robots' situations, update their future plans
    $r \leftarrow \theta_r(r)$ ▷ Update the State based on DFA
    Add θ_r to $plan$ ▷ Update plans according to the transition we made
  end while
  return Plan
end procedure

The main difference between multi-agent and single agent algorithm here is the nature of cost function. The cost function C for any transition function θ penalizes for (1) The time left for the robots that are needed but didn't finished their execution; (2) The average cost (in our simulated case, average distance) for robots to reach their assigned destinations; and (3) The 'remoteness' of destinations. After the execution we want the robots to be as near the center as possible as it will greatly boost the cost of next execution.

3 Implementation

We ran the experiments of the entire pipeline in simulation to demonstrate the ability of our system. The implementation of each component is detailed as follows:

3.1 Translating Natural Language into Logical Expressions

The translation part is run on Huggingface's transformer library and the finetuned ChatGLM3-6b-base model [18] with 6 billion parameters. We used a modified version of Liu's dataset for fine tuning [9]. Fine tuning is performed through Prompt Fine Tuning v2 [11]. The Input of the fine-tuned model is the Utterance, and the output of the model is the LTL, and there's no prompting involved in this approach. Training took roughly 4 hours on a workstation with a 16-Core AMD Ryzen 9 CPU and a single NVIDIA's RTX 4090 GPU.

The perference part is prompted by providing a general question and 5 examples of translation in the prompt. Llama2-7b [14] on on a computer with a 16-Core 5GHz AMD Ryzen 9 CPU and NVIDIA's RTX 4090 GPU. The prompt used is in the appendix.

3.2 Using DFA to plan actions & Game simulations

For the demonstration of idea, we utilized matplotlib library to create a 100x100 grid world for simulation. Obstacles and collision avoidance were included, real-life physics were not considered as the generalization of theory might not fit into this moving scheme. Gradient-based path finding along with real-time collision avoidance(by predicting robot teammates' future moves) were programmed.

The algorithm assumes a centralized planning under pre-determined graph, in the real world this graph could be collected using SLAM or converting from existing maps, while centralized planning is at this stage necessary.

The algorithm works as traversing the DFA with θ being go-to commands. A real-time available-robots list was maintained and the potential values for each single robot associated with each target was the cost for every θ.

4 Results and evaluations

4.1 Translating Natural Language into LTL

4.1.1 LTL

To evaluate the performance of the LTL translation model, we perform an evaluation on Language to LTL dataset by Jason Liu et al. [9]. The dataset contains 34335 training samples and 15320 validation samples, and the sentence structure in the training samples and the validation samples are different. Therefore, our evaluation is indicative of generalization performance to unseen sentence structures. The graph below shows the performance of the tasks.

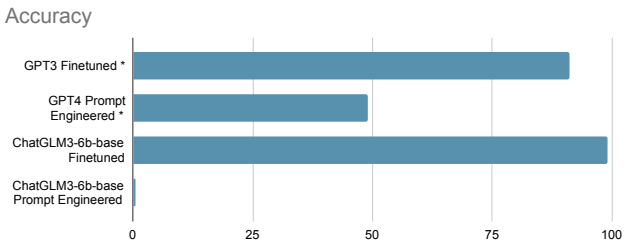

Figure 2: Comparison of Translation Accuracy across Models

The graph shows a consistent performance of finetuned model on Natural Language to LTL translation tasks, showing that modern LLMs can generalize and encode natural language with unseen structure reliably.

4.1.2 Preference

To evaluate this, we created a dataset of 116 different sentence structures, 5 manually created and 111 generated by prompting ChatGPT4. The sentence structures are used to generate different subtasks, a through e. The single-task evaluation consist of 928 sentences. Additional sentences are generated using different two-size combinations of the sub-tasks, which contains 8352 data points.

The result shows that the model is able to understand human's intentions and preference with just 5 examples. Especially in the case of human expressing preference one sub-task at a time, the accuracy can get to 82%. Note that this is only a one-way communication and translation, but with further communication and clarification between human and the robots, the accuracy of the translation can potentially be higher.

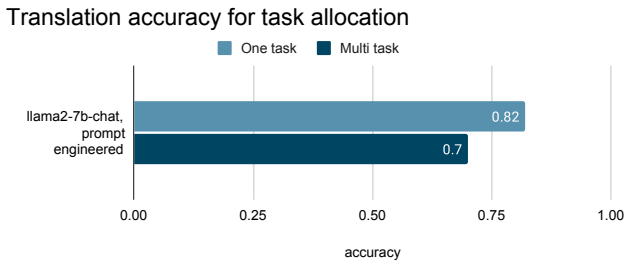

Figure 3: Accuracy of Human Preference Translation in Task Allocation

4.2 Game Simulation

Simulation's success rate is 100 percent given there is a valid path. With average optimization, we are able to saving 30 percent of the time if just randomly selecting the next action from the tasks left to be done. The video of the simulation is uploaded to YouTube. [1] [2]

5 Strength and Weaknesses

5.1 Strength

The strength of the work is the ability to utilize LTL and LLM to understand human commands and create plans in a general, extensible way. We also created a rich dataset consist of utterances in human-robot interaction.

5.2 Weaknesses

Our current approach definitely is just a proof of concept and requires further systematic study. While it is able to generalize across different structures, the accuracy is still 100 percent. Additionally, a more comprehensive dataset or a more systematically designed prompt can be used to further improve the performance.

We are also limited in the amount of information we can present in natural language. Some concept such as spacial concepts require more than one sentence to explain, sometimes even requiring computer vision.

Our work is also limited in that only task allocation is possible and we can not ask the robot and the human to work on the same task at the same time.

Lastly, the current approach only works in simulation with a naive approach of gradient-based navigation. More advanced path finding and robotics demo will be performed in the future to demonstrate the performance of our approach.

6 Related Work

There have been many solutions to try to represent natural languages in a structural way. [10] Dantam, Neil, and Mike Stilman presented a Linguistic Method for Robot Control, which employs context-free grammars (CFGs) provide a natural representation for hierarchies in the system. Ontology trees are also common technique used by mapping meaningful concepts together. However, it lacks standards for concept interpretation and interpretation evaluation. Lastly, statistical models have also been employed to better understand human commands[7, 13], however they do not have the ability to generalize as well due to the limited training set and model size.

Recently, due to the development of LLM, there have been many work studying the use of large or special statistical models on grounding natural languages to logic in robotics learning. [9, 2, 12, 5].

[1] https://www.youtube.com/watch?v=jpmWZjVp4vk

[2] https://www.youtube.com/watch?v=gH_uRuzYv88

However, most of the work only focus on single-agent, and often the dataset provided are mostly biased toward single-agent tasks as well.

Compared to existing work, our approach is able to combine the generalizability of large language models, and the focus on human-robot interaction in multi-robot settings.

7 Ethical Implications

Several Ethical Implications could result from the development of such robotics framework:

Job Displacement and Unemployment: While robots that are really good at understanding natural languages can provide valuable support for human, that also means that they will replace some of the human workers, leading to job displacement.

Bias: Although LLMs are usually trained on a variety of data in different languages and different situations, it is sometimes still prone to data biases, with the added complication that we have less control over the original LLM and training data as compared to if we are just training our own model.

Safety: Since robots will be collaborating with humans, ensuring the safety of humans will be a critical concern. And the inability of robots to properly interpret human commands can lead to questions of responsibility and liability.

Privacy and Surveillance: Since such intelligent robots may be deployed not just in factory settings but also home settings, the issue of privacy is import here, as the robot may be able to gain access to private information. It is important to investigate how we store or transfer collected data in our robots when it gets deployed, so that users can trust and depend on the robot.

8 Summary

In conclusion, we presented a framework to translate natural language into formal logic, and then plan using the resulting logical expressions, allowing the multiple robots to collaboratively complete tasks with human.

In the future, ...

more robust dataset

References

[1] C. Baier and J.-P. Katoen. *Principles of model checking*. MIT press, 2008.

[2] M. Berg, D. Bayazit, R. Mathew, A. Rotter-Aboyoun, E. Pavlick, and S. Tellex. Grounding language to landmarks in arbitrary outdoor environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 208–215. IEEE, 2020.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] A. Duret-Lutz, E. Renault, M. Colange, F. Renkin, A. G. Aisse, P. Schlehuber-Caissier, T. Medioni, A. Martin, J. Dubois, C. Gillard, and H. Lauko. From Spot 2.0 to Spot 2.10: What's new? 13372:174–187, Aug. 2022. doi: 10.1007/978-3-031-13188-2_9.

[5] F. Fuggitti and T. Chakraborti. Nl2ltl–a python package for converting natural language (nl) instructions to linear temporal logic (ltl) formulas. In *AAAI Conference on Artificial Intelligence*, 2023.

[6] C. Gupta, A. Jain, and N. Joshi. Fuzzy logic in natural language processing–a closer view. *Procedia computer science*, 132:1375–1384, 2018.

[7] G.-J. M. Kruijff, H. Zender, P. Jensfelt, and H. I. Christensen. Situated dialogue and spatial organization: What, where... and why? *International Journal of Advanced Robotic Systems*, 4 (1):16, 2007.

[8] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. A. Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

[9] J. X. Liu, I. Idrees, S. Liang, B. Schornstein, S. Tellex, and A. Shah. Lang2ltl: Translating natural language commands to temporal robot task specification. *CoRR*, abs/2302.11649, 2023. doi: 10.48550/ARXIV.2302.11649. URL `https://doi.org/10.48550/arXiv.2302.11649`.

[10] R. Liu and X. Zhang. A review of methodologies for natural-language-facilitated human–robot cooperation. *International Journal of Advanced Robotic Systems*, 16(3):1729881419851402, 2019.

[11] X. Liu, K. Ji, Y. Fu, W. L. Tam, Z. Du, Z. Yang, and J. Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

[12] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu. Unifying large language models and knowledge graphs: A roadmap. *arXiv preprint arXiv:2306.08302*, 2023.

[13] N. Shimizu and A. Haas. Learning to follow navigational route instructions. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[14] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[16] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[17] R. M. Weischedel. Knowledge representation and natural language processing. *Proceedings of the IEEE*, 74(7):905–920, 1986.

[18] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

A Contribution of each author

Yichen Wei contributed to the LLM translation part and led the development of the project.

Arnie He contributed to the LTL planning part and the lower-level simulation of the robots.

B LLM Prompts

B.1 Intentions

[INST] «SYS»
You are a robot trying to do task allocation with human.
«/SYS»
The AI has been trained to answer questions, provide recommendations, and help with decision making. The AI needs to to do task allocation with human. The AI follows user requests and understands user feelings. The AI should offer to help when needed.
Translate the human-robot coordination conversation above into logical expression regarding what the robot should do. If human requests the robot to do something, the AI should do it. If human is unwilling to or can't do something, the AI should do it. However, if human elects to do something, then the AI should not do it. For each human sentence, give a logic command for what the robot should do. Start your solution for each entry with "Logical Command for AI:" and put your expression in the quotes.
Do an analysis of what human needs. Give logical explanations. Additionally, give a natural language response as a confirmation or support to the human asking for help.

Human Sentence: I'll handle the other tasks, so could you please focus on e and g?
Logical Command for AI: "e & g"

Human Sentence: Leave a to me, and you can handle the other responsibilities.
Logical Command for AI: "¬a"

Human Sentence: Do not touch a
Logical Command for AI: "¬a"

Human Sentence: c will be done for you
Logical Command for AI: "¬c"

Human Sentence: I'm frustrated with a and could really use your help.
Logical Command for AI: "a"

Human Sentence: <Sentence>
Logical Command for AI: